

Dokumentacja bramki płatniczej

Data wygenerowania: 2024-07-25

Płatności Online Autopay	9
Definicje	9
Adresy środowisk	13
Obsługa transakcji i rozliczeń	13
Schemat działania usługi obsługi transakcji i rozliczeń	13
Kroki integracji obsługi transakcji i rozliczeń	14
Dane wymagane podczas integracji obsługi transakcji i rozliczeń	14
Dane wymieniane w środowisku testowym	14
Dane przekazywane przez Partnera do AP:	14
Dane przekazywane przez AP do Partnera:	14
Dane przekazywane w środowisku produkcyjnym	15
Przez Partnera do AP:	15
Przez AP do Partnera:	15
Implementacja interfejsów i procesów po stronie Partnera	15
Testy integracyjne i migracja na środowisko produkcyjne	15
Rozpoczęcie transakcji	15
Opis rozpoczęcia transakcji	15
Lista parametrów rozpoczęcia transakcji	16
Sposób rozpoczęcia transakcji	19
Przekierowanie do Serwisu Partnera	20
Opis przekierowania do Serwisu Partnera	20
Lista parametrów przekierowania do Serwisu Partnera	20
Powiadomienia natychmiastowe (ITN)	21
Opis powiadomień natychmiastowych	21
Lista zwracanych parametrów dla powiadomień natychmiastowych	22
Odpowiedź na powiadomienie natychmiastowe	23
Opis pól potwierdzenia dla powiadomień natychmiastowych	24
Szczegółowy opis zachowania i zmiany statusów płatności (paymentStatus)	25
Obsługa statusów transakcji z ITN – model uproszczony	25
Obsługa statusów transakcji z ITN – model pełny	26
Bezpieczeństwo transakcji	27
Opis bezpieczeństwa transakcji	27
Obliczanie wartości funkcji skrótu	27
Sposób obliczania wartości funkcji skrótu – pole Hash	27
Przykładowe obliczenia wartości funkcji skrótu podczas rozpoczęcia transakcji	28
Przykładowe obliczenia wartości funkcji skrótu podczas powrotu Klienta do Serwisu Partnera	29
Przykładowe obliczenia wartości funkcji skrótu w komunikacie ITN	30
Przykładowe obliczenia wartości funkcji skrótu w odpytaniu o listę Kanałów Płatności	31
Rejestracja i obsługa Serwisów w oparciu o Formularz Integratorski	32
Schemat działania usługi dodawania i edycji Serwisów	32
Dane wymieniane podczas integracji Formularza Integratorskiego	34
Dane wymieniane w środowisku testowym	34
Dane przekazywane przez AP do Partnera:	34
Dane przekazywane przez Partnera do AP:	34
Dane przekazywane w środowisku produkcyjnym	34
Przez AP do Partnera:	34
Przez Partnera do AP:	34
Pobranie przez Integratora linku do Formularza Integratorskiego	35
Opis pól wysyłanych w komunikacie żądania do AP	35

Opis pól komunikatu odpowiedzi do Integratora	38
Kody błędów	39
Widget onboardingowy	40
Przykładowy kod strony HTML z wykorzystaniem widgetu onboardingowego	41
Powrót Klienta do Platformy Integratora	43
Wysyłka danych konfiguracyjnych Serwisu oraz Linków Weryfikacyjnych (ICN)	43
Wysyłka danych konfiguracyjnych poprzez REST	43
Opis pól wysyłanych w komunikacie żądania do Integratora	43
Opis pól komunikatu odpowiedzi do AP	45
Wysyłka danych konfiguracyjnych poprzez Web-Services	45
Opis pól wysyłanych w komunikacie żądania do Integratora -	
InstantConfigurationNotificationRequest	45
Opis pól komunikatu	46
Komunikat odpowiedzi do AP	48
Opis pól komunikatu	48
Wysyłka danych konfiguracyjnych poprzez Email	48
Natychmiastowe powiadomienia o zmianie danych AML po przeprowadzonej weryfikacji w	
Systemie Autopay- ICN Verification	48
Struktura komunikatu	49
Opis głównych pól	51
Wartości poszczególnych pól w zależności od scenariusza	52
Natychmiastowe powiadomienie o zmianie statusu płatności kartowych (ICN Card)	54
Opis pól komunikatu wysyłanego do Integratora	55
Opis pól komunikatu zwrotnego do AP	56
Natychmiastowe powiadomienie o dokonanej zmianie polegającej na włączeniu oraz	
wyłączeniu danego kanału płatności na sklepie (ICN Gateway)	56
Opis pól komunikatu wysyłanego do Integratora.	56
Opis pól komunikatu zwrotnego do Autopay.	58
Pobranie przez integratora danych Serwisu - GetAMLCompanyData	59
GetAMLCompanyDataReq	59
Opis pól komunikatu	59
Opis pól komunikatu	59
"GetAMLCompanyDataResp"	60
Opis pól komunikatu	61
Pobranie informacji o aktualnej liście dostępnych regulaminów - GetLegalData	61
Opis pól komunikatu żądania	61
Opis pól komunikatu odpowiedzi	63
Zmiana konfiguracji Serwisu - "UpdateConfiguration"	65
"UpdateConfigurationReq"	65
Opis pól komunikatu	66
Opis pól komunikatu	66
Opis pól komunikatu	67
"UpdateConfigurationResp"	68
Opis pól komunikatu	68
Anulowanie aktualnej edycji serwisu, która jest w trakcie weryfikacji - "CancelUpdate"	
.....	69
CancelUpdateReq	69
Opis pól	69
CancelUpdateResp	69
Opis pól	70
Typy złożone	70

Header	70
Opis pól komunikatu	70
Address	71
Opis pól komunikatu	71
NIPTYPE	71
Currency	72
ForeignTransferModeType	72
Beneficials	72
Beneficial	72
Opis pól komunikatu	73
BeneficialOwners	73
BeneficialOwner	73
Opis pól komunikatu	74
Plenipotentiary	74
Opis pól komunikatu	74
Partners	75
Partner	75
Opis pól komunikatu	75
Service	76
Opis pól komunikatu	77
Company	79
Opis pól komunikatu	80
PhysicalPerson	82
Opis pól komunikatu	83
Zasady weryfikacji na formularzu danych dotyczących beneficjentów	83
Lista parametrów dla Przelewu weryfikacyjnego	89
Lista parametrów, dla których rejestracja lub edycja danych serwisu spowoduje wygenerowanie linku Przelewu weryfikacyjnego	89
Link do przelewu weryfikacyjnego	90
Rejestracja i obsługa Punktów Rozliczeń pomiędzy AP a Platformą Marketplace	
Partnera	91
Opis działania usługi automatycznej rejestracji i obsługi Punktów Rozliczeń	91
Dane wymieniane podczas integracji obsługi Punktów Rozliczeń	91
Dane wymieniane w środowisku testowym	91
Dane przekazywane w środowisku produkcyjnym	92
Proces dodawania Punktów Rozliczeń	93
Opis procesu	93
Pobranie aktywnego linku do formularza rejestracyjnego	93
Opis pól komunikatu	93
Odpowiedź na żądanie	94
Przykłady wymienianych komunikatów podczas pobierania linka do formularza ...	95
Dodatkowe rozszerzenia	97
Alternatywne modele rozpoczęcia transakcji	97
Preautoryzacja kartowa	97
Opis ogólny działania usługi preautoryzacji kartowej	97
Etapy transakcji preautoryzacji kartowej	98
Założenie blokady na wniosek Partnera	98
Obciążenie karty na wniosek Partnera	99
Opis obciążenia karty na wniosek Partnera	99
Opis dostępnych parametrów dla obciążenia karty na wniosek Partnera	99
Potwierdzenie wykonania operacji dla obciążenia karty na wniosek Partnera ...	

100	
Opis zwracanych parametrów dla obciążenia karty na wniosek Partnera	101
Zwolnienie blokady na wniosek Partnera	101
Zwolnienie blokady po przeterminowaniu transakcji	101
Schematy dla Preautoryzacji	102
Schemat A dla Preautoryzacji: Zakładanie blokady podczas autoryzacji płatności jednorazowej	102
Schemat B dla Preautoryzacji: Zakładanie blokady podczas inicjowania płatności automatycznej (zapisywania karty)	102
Schemat C dla Preautoryzacji: Zakładanie blokady z wykorzystaniem wcześniej zapisanej karty	103
Schemat D dla Preautoryzacji: Zlecenie przez Partnera obciążenia wcześniej zautoryzowanej karty	104
Schemat E dla Preautoryzacji: Zlecenie przez Partnera zwolnienia blokady (bez potrącania środków)	105
Schemat F dla Preautoryzacji: Zwolnienie blokady przez System (bez potrącania środków)	106
Przedtransakcja	106
Opis Przedtransakcji	106
Wywołanie Przedtransakcji	108
Odpowiedź na Przedtransakcję – link do kontynuacji transakcji	109
Obiekt transaction dla Przedtransakcji	109
Atrybuty obiektu transaction dla Przedtransakcji	109
Odpowiedź na Przedtransakcję – brak kontynuacji transakcji	110
Wynik Przedtransakcji	110
Poprawna walidacja parametrów	111
Obsługa odpowiedzi dla Przetransakcji	112
Zamówienie danych do przelewu w transakcji typu Szybki Przelew	113
Opis zamawiania danych do przelewu w transakcji typu Szybki Przelew	113
Wywołanie	113
Odpowiedź – dane do przelewu	114
Lista zwracanych parametrów dla odpowiedzi	115
Płatność BLIK 0 OneClick	116
Opis płatności BLIK 0 OneClick	116
Wywołanie płatności BLIK 0 OneClick	117
Przelewy do Urzędu Skarbowego	117
Opis	117
Walidacja tytułów przelewów do Urzędu Skarbowego	117
Google Pay	121
Opis	121
Schemat komunikacji	121
Rejestracja Transakcji Google Pay	122
Informację dodatkowe	123
Apple Pay	123
Widget Autopay (model WhiteLabel)	124
Autopay WidgetJS SDK	125
Osadzenie SDK	125
Przykłady konfiguracji	126
Szczegółowe omówienie konfiguracji obiektu WidgetConnection	126
Widget Kartowy - Przykład implementacji na stronie partnera	127
Widget Kartowy - Szczegółowy schemat komunikacji i wymiany danych	135

Widget Visa Mobile - Przykład implementacji na stronie partnera	136
Widget Visa Mobile - Szczegółowy schemat komunikacji i wymiany danych	141
Omówienie przykładowego kodu HTML JS (Widget Kartowy i VisaMobile)	142
Płatność kartą osadzoną w Serwisie (iFrame)	145
Płatność automatyczna	146
Opis płatności automatycznej	146
Aktywacja płatności automatycznej	146
Proces aktywacji usługi płatności automatycznych	147
Komunikat startu transakcji automatycznej	147
Powiadomienie o uruchomieniu płatności automatycznej (RPAN)	148
Opis zwracanych parametrów dla uruchomienia płatności automatycznej	149
Element confirmation płatności automatycznej	153
Schemat ponawiania komunikatów ITN/ISTN/IPN/RPAN/RPDN	153
Obciążenie dla płatności automatycznej	154
Komunikat startu transakcji płatności automatycznej	154
Dezaktywacja usługi	155
Komunikat dezaktywacji płatności automatycznej	155
Lista parametrów dezaktywacji płatności automatycznej	155
Odpowiedź	156
Element confirmation	156
Powiadomienie o dezaktywacji płatności automatycznej (RPDN)	157
Opis zwracanych parametrów	157
Potwierdzenie otrzymania komunikatu	158
Element Confirmation	159
Weryfikacja tożsamości płatnika	159
Opis weryfikacji tożsamości płatnika	159
Rozpoczęcie transakcji z dodatkowymi parametrami	161
Koszyk produktów	168
Opis koszyka produktów	168
Element Params	169
Wyświetlanie koszyka produktów na ekranie wyboru Kanału Płatności	171
Dodatkowe opcje komunikacji online do Partnera	171
Dodatkowe pola w komunikacie ITN/IPN transakcji wejściowej	171
Opis	171
Pełna lista dodatkowych pól w komunikacie ITN/IPN	171
Szczegółowy opis zmiany statusu weryfikacji – dla transakcji zakończonej poprawnie (wynik pozytywny lub negatywny)	176
Szczegółowy opis zmiany statusu weryfikacji – dla transakcji nie zakończonej poprawnie	176
Szczegółowe statusy transakcji	176
Wartości statusów transakcji – Statusy ogólne (niezależne od kanału płatności)	177
Wartości statusów transakcji – Statusy kartowe	177
Wartości statusów transakcji – Statusy specyficzne dla transakcji BLIK	178
Natychmiastowe powiadomienia o zmianie statusu produktu (IPN)	178
Natychmiastowe powiadomienia o zmianie statusu transakcji rozliczeniowej (ISTN) ...	178
Zwracane parametry	179
Odpowiedź na powiadomienie	181
Szczegółowy opis zachowania i zmiany statusów rozliczenia (transferStatus)	181
Usługi dodatkowe	182

Odpytywanie o listę aktualnie dostępnych Kanałów Płatności	182
Opis	182
Lista dostępnych parametrów	182
Odpowiedź na żądanie	183
Odpytywanie o listę aktualnie dostępnych zgód formalnych	187
Opis	187
Lista dostępnych parametrów	187
Lista zwracanych parametrów	188
Opis obsługi odpowiedzi	192
Odpytanie o stan salda	193
Opis	193
Lista dostępnych parametrów dla aktualnego stanu salda	193
Odpowiedź na żądanie aktualnego salda	194
Lista pól odpowiedzi	195
Zasilanie salda	195
Opis	195
Wypłaty z salda	196
Opis	196
Lista dostępnych parametrów dla wypłat z salda	196
Odpowiedź na żądanie	200
Opis pól	200
Zwroty transakcji	201
Opis	201
Lista dostępnych parametrów	202
Opis pól	203
Zwroty produktu	204
Opis	204
Lista parametrów	204
Odpowiedź na żądanie	205
Opis pól	205
Odpytanie o status zwrotu lub wypłaty z salda	206
Opis	206
Lista dostępnych parametrów dla wypłat z salda	206
Odpowiedź na żądanie	207
Opis pól	207
Strona podsumowania transakcji	208
Opis	208
Lista parametrów dla metody podsumowania transakcji	208
Odpytanie o status transakcji	209
Opis	209
Lista parametrów dostępnych dla statusu transakcji	209
Nagłówek HTTP dla zapytania o status transakcji	210
Lista pól dla zapytania o status transakcji	210
Obsługa odpowiedzi zapytania o status transakcji - propozycja obsługi wielu transakcji w odpowiedzi	212
Anulowanie nieopłaconej transakcji	212
Opis anulowania nieopłaconej transakcji	212
Lista parametrów dostępnych dla anulowania nieopłaconej transakcji	212
Nagłówek dla anulowania nieopłaconej transakcji	213
Lista parametrów dla anulowania nieopłaconej transakcji	214
Odpowiedzi na żądania anulowania transakcji	214

Komunikaty błędów	215
Opis komunikatów błędów	215
Schematy obsługi transakcji i rozliczeń	215
Rozszerzona struktura serwisów i punktów rozliczeń	215
Modele rozliczeń	217
Model rozliczeń zbiorczych transakcji (model domyślny)	217
Model rozliczeń transakcji po każdej wpłacie	218
Model rozliczeń zbiorczych produktów	219
Model rozliczeń produktów po każdej wpłacie	220
Model rozliczeń transakcji na żądanie	221

Płatności Online Autopay

Płatności Online Autopay to kompleksowe rozwiązanie umożliwiające przyjmowanie płatności od klientów sklepów internetowych, obsługujące liczne metody płatności dostępne na rynku, takie jak przelewy, Pay by link, BLIK, Google Pay, Apple Pay. W niniejszej dokumentacji znajdziesz wszystko, czego potrzebujesz do szybkiego wdrożenia bramki płatności w sklepie internetowym. Dokumentacja Płatności Online Autopay zawiera części takie jak [Obsługa transakcji i rozliczeń](#), [Rejestracja i obsługa Punktów Rozliczeń pomiędzy AP a Platformą Marketplace Partnera](#), [Rejestracja i obsługa Serwisów w oparciu o Formularz Integratorski](#), [Dodatkowe rozszerzenia](#).

Definicje

Aplikacja - Aplikacja Mobilna Partnera, komunikująca się z SDK Systemu Płatności Online Autopay w celu rejestrowania Transakcji.

AP - Autopay Spółka Akcyjna z siedzibą w Sopocie przy ulicy Powstańców Warszawy 6, wpisana do rejestru przedsiębiorców prowadzonego przez Sąd Rejonowy Gdańsk-Północ w Gdańsku, VIII Wydział Gospodarczy Krajowego Rejestru Sądowego pod numerem KRS 0000320590, NIP 585-13-51-185, Regon 191781561, o kapitale zakładowym w wysokości 2 205 500 PLN (w całości opłaconym), nadzorowana przez Komisję Nadzoru Finansowego i wpisana do rejestru krajowych instytucji płatniczych pod numerem IP17/2013, właściciel Systemu.

BalancePoint (Punkt Rozliczeń) - obiekt w Systemie Płatności reprezentujący Sklep zintegrowany z Platformą Marketplace oraz zarejestrowany w Systemie Płatności przy użyciu formularza udostępnionego Partnerowi przez AP.

ClientHash - parametr w komunikatach; pozwala w sposób zanonimizowany przypisać Instrument płatniczy (np. Kartę) do Klienta. Na jego podstawie Partner może wywoływać kolejne obciążenia w modelu płatności automatycznych.

CommissionModel - model prowizyjny ustalony z Integratorem. Opisuje wartości prowizji za transakcje przekazywane AP oraz Integratorowi.

Dzień Roboczy - dzień tygodnia od poniedziałku do piątku, z wyłączeniem dni ustawowo wolnych od pracy.

Formularz Integratorski - strona internetowa, na której umieszczony jest formularz umożliwiający Klientowi rejestrację, edycje lub dodanie kolejnego Serwisu.

Instrument Płatniczy (Kanał Płatności) - uzgodniony przez Klienta i jego dostawcę zbiór procedur lub zindywidualizowane urządzenie, wykorzystywane przez Klienta do złożenia zlecenia płatniczego np. Karta, PBL.

Narzędzie e-przelew - uzgodniony przez Partnera i AP zbiór procedur lub zindywidualizowane urządzenie, wykorzystywane przez Partnera do złożenia zlecenia płatniczego umożliwiającego realizację wypłatę środków z salda na rachunek bankowy Partnera lub Klienta oraz inny instrument płatniczy należący do Partnera lub Klienta. Udostępnienie funkcjonalności jest zależne od indywidualnych ustaleń między Partnerem i AP.

Integrator – Integratorami nazywamy partnerów, którzy wdrożyli w swoich systemach Płatności Online Autopay i umożliwiają ich aktywację z poziomu własnych rozwiązań. Do integratorów zaliczamy takie podmioty jak: Shoper, Sky-Shop, Gymsteer, Selly weryfikacje, FaniMani, AtomStore, Ebexo, Selly Azymut, PayNow, Comarch.

IPN (Instant Product Notification) – natychmiastowe powiadomienie wysyłane z Systemu płatności online do Serwisu Partnera przekazujące zmianę statusu produktu. Struktura IPN jest podobna do ITN (rozszerzona jedynie o węzeł *product*).

ITN (Instant Transaction Notification) – natychmiastowe powiadomienie wysyłane z Systemu płatności online do Serwisu Partnera przekazujące zmianę statusu transakcji.

ISTN (Instant Settlement Transaction Notification) – natychmiastowe powiadomienia o zmianie statusu transakcji rozliczeniowej. System niezwłocznie przekazuje powiadomienia o fakcie zlecenia transakcji rozliczeniowej (ew. wypłatach/zwrotach) oraz zmianie jej statusu.

ICN (Instant Configuration Notification) – natychmiastowe powiadomienie o konfiguracji nowozarejestrowanego sklepu, przekazujące informacje o zmianie statusu kartowego w sklepie (np. w przypadku uruchamiania kart). Komunikaty ICN mogą być również wysyłane w przypadku zmiany konfiguracji sklepu, zmiany jego danych AML, włączenia/wyłączenia kanału płatności. Udostępnienie funkcjonalności jest zależne od indywidualnych ustaleń między Partnerem i AP.

Karta – karta płatnicza wydana w ramach systemów VISA i Mastercard, dopuszczona regulacjami tychże systemów do realizacji Transakcji bez fizycznej jej obecności.

Klient (Płatnik) – osoba uiszczająca w Serwisie płatność za usługi lub produkty Partnera przy wykorzystaniu Systemu.

Koszyk produktów – jest to informacja o składowych płatności, przekazywana (w Linku płatności) do Systemu w celu późniejszego jej przetwarzania. Każdy produkt koszyka opisują dwa obowiązkowe pola: kwota składowa oraz pole pozwalające przekazać parametry charakterystyczne dla produktu.

Link płatności – żądanie umożliwiające start Transakcji wejściowej (opisanej w części [Rozpoczęcie transakcji](#)). Można go stosować bezpośrednio na stronach www (metoda POST), natomiast w mailach do Klientów należy posłużyć się *Przedtransakcją* w celu uzyskania krótkiego linka do płatności (metoda GET).

Link weryfikacyjny – adres URL kierujący do Przelewu weryfikacyjnego.

Marketplace – rozwiązanie płatnicze działające w ramach Systemu Płatności Online Autopay. Umożliwia Partnerowi obsługę platformy sprzedażowej, na której produkty lub usługi oferowane są Klientom przez Kontrahentów Partnera. Zaawansowane modele rozliczenia Transakcji oraz Transakcji Rozliczeniowych pozwalają na realizację płatności bezpośrednio od Klienta do Kontrahenta Partnera, z uwzględnieniem Koszyka produktów. Udostępnienie funkcjonalności jest zależne od indywidualnych ustaleń między Partnerem i AP.

Model Płatnika – model, w którym prowizję za przeprowadzoną transakcję opłaca klient na rzecz AP (koszt doliczany do kwoty transakcji). W tym przypadku klient podczas płatności akceptuje również regulamin AP.

Model Merchanta – model, w którym prowizja jest rozliczana między Autopay a partnerem i nie jest

doliczana do kwoty transakcji opłacanej przez klienta.

Partner - podmiot będący odbiorcą środków z tytułu sprzedaży produktów lub usług dystrybuowanych przez Partnera w Serwisie.

Partner, w przypadku modelu Marketplace, to podmiot, niebędący konsumentem, zainteresowany obsługą przyjmowania przez AP należnych Partnerowi płatności za produkty lub usługi dystrybuowane przez Partnera.

Pay By Link (PBL) - narzędzie umożliwiające realizację płatności za pośrednictwem przelewu wewnątrzbankowego z rachunku Klienta na rachunek AP. Po zalogowaniu do bankowości internetowej - dane potrzebne do realizacji przelewu (dane informacyjne odbiorcy, numeru jego rachunku bankowego, kwota i data realizacji przelewu) są wypełnione automatycznie dzięki systemowi wymiany danych pomiędzy bankiem a AP.

Pełnomocnik techniczny - podmiot posiadający prawo dostępu do Rachunku Płatniczego Partnera, który autoryzuje ten dostęp (zgoda lub umowa). W systemie pełnomocnictwo jest reprezentowane przez konfigurację **PlenipotentiaryID**: jeden podmiot może mieć wiele pełnomocnictw dla różnych Partnerów.

Platforma Marketplace - platforma, na której udostępniona jest opcja rejestracji Punktów rozliczeń.

Płatność automatyczna - płatność dokonywana bez potrzeby każdorazowego wprowadzania danych Karty lub danych do autoryzacji przelewu.

Płatność jednym kliknięciem - jest to Płatność automatyczna zlecana przez Klienta.

Płatność cykliczna - jest to Płatność automatyczna zlecana bez udziału Klienta (przez Serwis Partnera).

Przedtransakcja - specyficzny (wykonywany w tle) sposób zamawiania linku do płatności.

Przelew weryfikacyjny - część procesu związana z rejestracją oraz edycją Serwisu/Serwisów Partnera w Systemie. Polega na wykonaniu przez Partnera przelewu środków w celu weryfikacji danych i rachunku bankowego do wypłat środków do Partnera. Weryfikacja danych jest obowiązkiem AP wynikającym m.in. z Ustawy z dnia 16.11.2000 r. o przeciwdziałaniu praniu pieniędzy oraz finansowaniu terroryzmu. Każdy przelew weryfikacyjny musi otrzymać końcowy status weryfikacji (pozytywny lub negatywny) w ciągu 30 dni od opłacenia transakcji. Jeżeli status końcowy weryfikacji nie zostanie nadany w wyżej określonym czasie, system automatycznie nada jej status negatywny. Proces ten dotyczy weryfikacji, w której Autopay kieruje prośbę o uzupełnienie danych do klienta i nie otrzyma zwrotnej informacji niezbędnej do przeprowadzenia tejże weryfikacji.

Rachunek Płatniczy (Saldo) - rachunek płatniczy prowadzony przez AP dla Partnera, na którym gromadzone są środki wpłacone od Klientów. Udostępnienie funkcjonalności jest zależne od indywidualnych ustaleń między Partnerem i AP.

RPAN (Recurring Payment Activation Notification) - komunikat o aktywowaniu usługi płatności automatycznych.

RPDN (Recurring Payment Deactivation Notification) - komunikat o dezaktywacji usługi płatności automatycznych.

Serwis – strona lub strony internetowe Partnera zintegrowane z Systemem, na których Klient może nabyć od Partnera (lub od Kontrahenta Partnera w przypadku Marketplace) produkty lub usługi. W przypadku Marketplace, obiekt w Systemie Płatności reprezentujący Marketplace Partnera. Dowiązane są do niego wszystkie transakcje startowane przez wspomniany Marketplace.

Specyfikacja – dokumentacja opisująca komunikację pomiędzy Serwisem a Systemem.

System Płatności Online AP (System) – rozwiązanie informatyczno-funkcjonalne, w ramach którego AP udostępnia Partnerowi aplikację umożliwiającą procesowanie płatności Klientów dokonanych przy użyciu Instrumentów Płatniczych, a także weryfikację statusu płatności oraz odbiór płatności.

Szybki Przelew – realizacja płatności za pośrednictwem przelewu wewnątrzbankowego z rachunku Klienta na rachunek AP. Od płatności dokonywanych za pośrednictwem PBL płatność różni się koniecznością samodzielnego wypełnienia wszystkich danych potrzebnych do dokonania przelewu przez Klienta.

Transakcja - oznacza transakcję płatniczą w rozumieniu Ustawy z dnia 19 sierpnia 2011 r. o usługach płatniczych.

Transakcja wejściowa – część procesu obsługi płatności, dotycząca wpłaty dokonywanej przez Klienta do AP.

Transakcja rozliczeniowa – część procesu obsługi płatności, dotycząca przelewu wykonywanego przez AP na rachunek Partnera. Aby powstała Transakcja rozliczeniowa, Transakcja wejściowa musi zostać przez Klienta opłacona. Transakcja rozliczeniowa może dotyczyć pojedynczej transakcji wejściowej (wpłaty), bądź agregować ich wiele.

Ustawa – ustawa z dn. 19 sierpnia 2011 r. o usługach płatniczych.

Ważność linku – parametr określający moment, po przekroczeniu którego Link płatności przestaje być aktywny. Powinna być ustawiana przez parametr LinkValidityTime w Linku płatności.

Ważność transakcji – parametr określający moment, po przekroczeniu którego System płatności online blokuje i automatycznie zwraca wpłaty Klienta. Wartość domyślna obliczana jest poprzez dodanie 6 dni do dnia wybrania przez Klienta Kanału Płatności. Może ona być również ustawiana przez parametr ValidityTime w Linku płatności. W takim przypadku, po upływie wskazanego czasu link przestaje być aktywny, a wpłaty są zwracane do Klienta. Maksymalna ważność transakcji to 31 dni.

Widget Autopay – mechanizm umożliwiający realizację płatności Kartą za produkty/usługi oferowane przez Partnera, w której dane Karty wprowadzane są przez Klienta do mechanizmu osadzonego bezpośrednio w Serwisie Partnera. Wywołanie formatki kartowej widgetu wymaga implementacji kodu JavaScript wykorzystującego dedykowaną bibliotekę AP.

Widget Onboardingowy - rozwiązanie pozwalające Integratorowi na osadzenie Formularza Integratorskiego (przygotowanego przez Autopay) bezpośrednio na stronie internetowej Integratora, dzięki czemu Partner podczas rejestracji swojego sklepu nie jest przekierowywany na domenę Autopay - cały proces jest realizowany w Serwisie Partnera.

WhiteLabel – model integracji, w którym Klient już w Serwisie dokonuje wyboru kanału płatności oraz akceptuje regulaminy (o ile konieczność ich akceptacji wynika z indywidualnych ustaleń między

Partnerem i AP), a start transakcji zawiera wypełnione pole GatewayID (oraz w określonych przypadkach DefaultRegulationAcceptanceID lub RecurringAcceptanceID).

Adresy środowisk

TEST

- host_bramki: <https://testpay.autopay.eu>
- host_kartowy_bramki: <https://testcards.autopay.eu>

PRODUKCJA

- host_bramki: <https://pay.autopay.eu>
- host_kartowy_bramki: <https://cards.autopay.eu>

Obsługa transakcji i rozliczeń

Schemat działania usługi obsługi transakcji i rozliczeń

W Serwisie Partnera, po skompletowaniu zamówienia, Klientowi prezentowana jest opcja możliwości wykonania płatności z wykorzystaniem Systemu. Kliknięcie w odpowiedni link powoduje rozpoczęcie transakcji i otwarcie w nowym oknie:

- a) dedykowanej strony Systemu przygotowanej przez AP, na której Klientowi prezentowana jest lista dostępnych Kanałów Płatności oraz podsumowanie zarejestrowanej transakcji lub
- b) bezpośrednio strony Kanału Płatności (Banku, BLIK lub do płatności Kartą).

Po stronie Systemu następuje walidacja przekazanych parametrów i zapisanie transakcji z ustalonym okresem ważności. Jeśli w momencie walidacji, czas ważności linku będzie już przekroczony, Klientowi zostanie wyświetlony odpowiedni komunikat (weryfikacja ważności transakcji następuje także przy zmianie statusu płatności). Po pozytywnej weryfikacji parametrów transakcji (oraz po wybraniu Kanału Płatności), Klient dokonuje autoryzacji transakcji. W jej tytule, oprócz nadawanych przez System identyfikatorów, może być także umieszczany stały opis, ustalony wcześniej pomiędzy AP a Partnerem lub dynamiczna wartość przekazywana przez Partnera przy starcie transakcji.

Zalecany model integracji polega na nadaniu komunikatu rozpoczęcia transakcji w tle, tzn. bez przekierowania użytkownika do Systemu (patrz [Przedtransakcja](#)). W tym modelu, możliwe jest zastosowanie zaawansowanych form autoryzacji transakcji (WhiteLabel, płatności automatyczne, SDK mobilne), diagnozowania poprawności przekazywanych parametrów oraz wielu innych rozszerzeń.

Po zakończeniu autoryzacji transakcji (na stronie Kanału Płatności) klient powraca z niego do Systemu, gdzie następuje automatyczne przekierowanie Klienta do Serwisu Partnera.

WSKAZÓWKA: Szczegółowy opis struktury linku powrotu znajduje się w części [Przekierowanie do serwisu Partnera](#).

Otrzymany z Kanału Płatności status autoryzacji (płatności) przekazywany jest z Systemu do Serwisu Partnera za pomocą komunikatu ITN. System będzie ponawiać wysyłanie komunikatów, aż do potwierdzenia odbioru przez Serwis Partnera lub upłynięcia czasu ważności powiadomienia. Transakcje, które zostaną zapłacone po opływie okresu ważności transakcji – zostaną zwrócone do Klienta (nadawcy przelewu).

Opcjonalnie System może powiadamiać o fakcie wystawienia Transakcji rozliczeniowej. Służy do tego odpowiednio zmodyfikowany komunikat ISTN.

Kroki integracji obsługi transakcji i rozliczeń

Dane wymagane podczas integracji obsługi transakcji i rozliczeń

Wymagane dane wymieniane podczas integracji różnią się dla środowiska testowego i produkcyjnego. Poniżej lista parametrów przekazywanych przez AP do Partnera oraz w kierunku odwrotnym.

Przekazywane są też informacje ogólne tj. aktywne kanały płatności wraz z grafikami (w wyniku odpytywania o listę dostępnych kanałów płatności).

Opcjonalnie, mogą pojawić się dodatkowe dane przekazywane przez Partnera do AP, na przykład: informacje o wymaganej zawartości koszyka i sposobie przetwarzania go (w raportach, rozliczeniach, panelu administracyjnym), dodatkowe wymagania (dla zasilenia salda przedpłaconego). Dla płatności automatycznych BLIK również domyślna długość życia aktywowanych płatności automatycznych oraz domyślna etykieta aktywowanych płatności automatycznych.

Dane wymieniane w środowisku testowym

Dane przekazywane przez Partnera do AP:

- Adres dla komunikatów ITN
- Adres dla komunikatów RPAN (może być ten sam, co dla komunikatów ITN) [dla płatności automatycznych]
- Adres dla komunikatów RPDN (może być ten sam, co dla komunikatów ITN) [dla płatności automatycznych]
- Adres powrotu z płatności (bez parametrów)

Dane przekazywane przez AP do Partnera:

- Adres Systemu płatności online
- ServiceID
- AcceptorID [dla portfeli w modelu WhiteLabel]
- Klucz współdzielony
- Mechanizm funkcji skrótu
- Adres testowego formularza
- Adres IP, z którego wysyłane są ITNy
- Adres do panelu administracyjnego
- Login

- Hasło

Dane przekazywane w środowisku produkcyjnym

Przez Partnera do AP:

- Adres dla komunikatów ITN
- Adres dla komunikatów RPAN (może być ten sam, co dla komunikatów ITN) [dla płatności automatycznych]
- Adres dla komunikatów RPDN (może być ten sam, co dla komunikatów ITN) [dla płatności automatycznych]
- Adres powrotu z płatności (bez parametrów)
- Adresy email dla raportów transakcyjnych
- Adresy email dla faktur i raportów rozliczeniowych
- Adresy email dla reklamacji (wysyłany w wiadomościach do płatników)

Przez AP do Partnera:

- Adres Systemu płatności online
- ServiceID
- AcceptorID [dla portfeli w modelu WhiteLabel]
- Klucz współdzielony
- Mechanizm funkcji skrótu
- Adres IP, z którego wysyłane są ITNy
- Adres do panelu administracyjnego
- Login
- Hasło

Implementacja interfejsów i procesów po stronie Partnera

W minimalnej wersji integracji należy zaimplementować obsługę rozpoczęcia transakcji, powrotu z niej oraz obsługę komunikacji ITN.

WSKAZÓWKA: Zaleca się zapoznanie ze schematem działania. W razie potrzeby warto zapoznać się także z dodatkowymi parametrami lub usługami.

Testy integracyjne i migracja na środowisko produkcyjne

Podczas wykonywania testów należy uzupełniać białe pola arkusza oraz podesłać go zwrótnie do AP, w celu potwierdzenia poprawności integracji przed migracją na środowisko produkcyjne.

WSKAZÓWKA: Przed wdrożeniem produkcyjnym zaleca się wykonać testy zgodnie ze [scenariuszami testowymi](#) w wersji podstawowej, a dla bardziej zaawansowanych integracji również zgodnie ze scenariuszami dodatkowymi.

Rozpoczęcie transakcji

Opis rozpoczęcia transakcji

Serwis Partnera inicjując transakcję przekazuje do Systemu płatności online parametry niezbędne do jej zrealizowania oraz późniejszego przekazania statusu płatności.

Wszystkie parametry przekazywane są metodą POST (*Content-Type: application/x-www-form-urlencoded*).

Protokół rozróżnia wielkość liter zarówno w nazwach jak i wartościach parametrów. Wartości przekazywanych parametrów powinny być kodowane w UTF-8 (oraz protokołem transportowym – zakodować przed wysłaniem, o ile narzędzie wykorzystane do wysyłki komunikatu nie robi tego samodzielnie, przykład kodowania: URLEncode).

Lista parametrów rozpoczęcia transakcji

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	ServiceID	TAK	string{1,10}	Identyfikator Serwisu Partnera, nadawany w trakcie rejestracji usługi jednoznacznie identyfikuje Serwis Partnera w Systemie płatności online. Dopuszczalne są cyfry.
2	OrderID	TAK	string{1,32}	Identyfikator transakcji o długości do 32 znaków alfanumerycznych alfabetu łańciskiego. Wartość pola musi być unikalna dla Serwisu Partnera. Dopuszczalne alfanumeryczne znaki alfabetu łańciskiego oraz znaki z zakresu: - _

kolejność Hash	nazwa	wymagany	typ	opis
3	Amount	TAK	amount	<p>Kwota transakcji. Jako separator dziesiętny używana jest kropka - '.' Format: 0.00; maksymalna długość: 14 cyfr przed kropką i 2 po kropce.</p> <p>UWAGA: Dopuszczalna wartość pojedynczej Transakcji w Systemie produkcyjnym to odpowiednio:</p> <ul style="list-style-type: none"> • dla PBL - min. 0.01 PLN, max. 100000.00 PLN (lub do wysokości ustalonej przez Bank wydający instrument płatniczy); • dla Kart płatniczych - min. 0.10 PLN, max. 100000.00 PLN (lub do wysokości indywidualnego limitu pojedynczej transakcji w Banku wydawcy Karty); • dla Szybkich przelewów - min. 0.01 PLN, max. 100000.00 PLN (lub do wysokości indywidualnego limitu pojedynczej transakcji w Banku dla przelewu wewnątrzbankowego); • dla BLIK - min. 0.01 PLN, max. 75000.00 PLN (lub do wysokości indywidualnego limitu pojedynczej transakcji w Banku dla przelewu wewnątrzbankowego); • dla OTP - min. 100.00 PLN, max. 2000.00 PLN; • dla Alior Rat - min. 50.00 PLN, max. 7750.00 PLN
4	Description	NIE	string{1,79}	<p>Tytuł transakcji (wpłaty); na początku tytułu przelewu umieszczane są identyfikatory transakcji nadawane przez System płatności online, do tego doklejana jest wartość tego parametru. W niektórych przypadkach, niezależnych od AP tytuł przelewu może zostać dodatkowo zmodyfikowany przez Bank, w którym nastąpiła wpłata dokonana przez klienta. Wartość parametru dopuszcza alfanumeryczne znaki alfabetu łacińskiego oraz znaki z zakresu: . : - , spacja.</p>

kolejność Hash	nazwa	wymagany	typ	opis
5	GatewayID	NIE	integer{1,5}	<p>Identyfikator Kanału Płatności, za pomocą, którego Klient zamierza uregulować płatność. Ten parametr odpowiada w szczególności za model prezentowania Kanałów Płatności:</p> <ul style="list-style-type: none"> - na stronie AP - wartość parametru „0”; - w Serwisie Partnera - wartość parametru odpowiada wybranemu przez Klienta Kanałowi Płatności np. GatewayID=3. <p>Wszystkie Kanały Płatności do samodzielnego osadzenia w Serwisie udostępniane są Partnerowi w ramach usługi gatewayList.</p>
6	Currency	NIE	string{1,3}	<p>Waluta transakcji; domyślną walutą jest PLN (użycie innej waluty musi być uzgodnione w trakcie integracji). W ramach ServicelD obsługiwana jest jedna waluta. Dopuszczalne jedynie wartości: PLN, EUR, GBP oraz USD.</p>
7	CustomerEmail	NIE	string{3,255}	<p>Adres email Klienta.</p>
19	ValidityTime	NIE	string{1,19}	<p>Moment upłynięcia ważności transakcji; po jego przekroczeniu link przestaje być aktywny, a wszelkie wpłaty są zwracane do nadawcy przelewu; przykładowa wartość: 2021-10-31 07:54:50; w przypadku braku parametru ustawiana jest wartość domyślna 6 dni.</p> <p>Maksymalna ważność transakcji to 31 dni (w przypadku ustawienia wartości parametru dalej, niż 31 dni do przodu, czas ważności zostanie odpowiednio skrócony).</p> <p>Np. transakcja wystartowana w chwili 2020-05-01 08:00:00, z parametrem ValidityTime = 2021-05-01 08:00:00, otrzyma ważność do 2020-06-01 08:00:00.(Czas w CET)</p>

kolejność Hash	nazwa	wymagany	typ	opis
34	LinkValidityTime	NIE	string{1,19}	Moment upłynięcia ważności linku; po jego przekroczeniu link przestaje być aktywny, nie wpływa to jednak na czas oczekiwania na wpłatę; przykładowa wartość: 2014-10-30 07:54:50; proszę zwrócić uwagę na to, aby do czasu ważności linku, dostosowany był czas ważności transakcji (być może zajdzie potrzeba podania również parametru ValidityTime , aby wydłużyć jej standardową ważność).
nd.	Hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji .

Sposób rozpoczęcia transakcji

Rozpoczęcie transakcji następuje przez przesłanie wywołaniem HTTPS kombinacji powyższych parametrów na ustalony w trakcie rejestracji usługi adres Systemu płatności online.

UWAGA: Liczba transakcji wystartowanych przez Partnera w ciągu jednej minuty może wynieść maksymalnie 100, chyba że Partner i AP ustalą wyższy limit w ramach zawartej umowy.

Przykład rozpoczęcia transakcji:

Adres:

- https://host_bramki/sciezka

Parametry:

- ServiceID=2
- OrderID=100
- Amount=1.50

```
Hash=2ab52e6918c6ad3b69a8228a2ab815f11ad58533eed963dd990df8d8c3709d1
```

Przesłanie komunikatu bez wszystkich **wymaganych** parametrów (**ServiceID**, **OrderID**, **Amount** i **Hash**) lub zawierającego błędne ich wartości, spowoduje zatrzymanie procesu płatności wraz z podaniem kodu błędu transakcji i krótką informacją o błędzie (brak powrotu na stronę Serwisu

Partnera).

WAŻNE! "Para parametrów **ServiceID** i **OrderID** jednoznacznie identyfikuje transakcję. Niedopuszczalne jest powtórzenie się wartości parametru **OrderID** przez cały okres świadczenia usług przez System na rzecz jednego Serwisu Partnera (**ServiceID**)."

Opcjonalny parametr **GatewayID** służy do określenia Kanału Płatności, za pomocą którego ma zostać zrealizowana płatność. Pozwala to przenieść ekran wyboru Kanałów Płatności do Serwisu. Aktualna lista identyfikatorów Kanałów Płatności, wraz z logotypami, dostępna jest poprzez metodę **gatewayList**.

Komunikat rozpoczęcia transakcji może być nadany w tle, tzn. bez przekierowania użytkownika do Systemu płatności online. W tym modelu, samego wyboru Kanału Płatności, Klient także dokonuje w Serwisie Partnera.

Przekierowanie do Serwisu Partnera

Opis przekierowania do Serwisu Partnera

Niezwłocznie po zakończeniu autoryzacji transakcji przez Klienta jest on przekierowywany z witryny Kanału Płatności na witrynę Systemu płatności online, gdzie następuje automatyczne przekierowanie Klienta do Serwisu Partnera.

Przekierowanie realizowane jest poprzez wysłanie żądania HTTPS (metodą GET) pod ustalony wcześniej adres powrotu w Serwisie Partnera. Protokół rozróżnia wielkość liter zarówno w nazwach jak i wartościach parametrów.

Lista parametrów przekierowania do Serwisu Partnera

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	ServiceID	TAK	string{1,10}	Identyfikator Serwisu Partnera.
2	OrderID	TAK	string{1,32}	Identyfikator transakcji nadany w Serwisie Partnera i przekazany w starcie transakcji.
nd.	Hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis.

Przykład komunikatu przekierowującego Klienta z Systemu płatności online do Serwisu Partnera

```
https://sklep_nazwa/strona_powrotu?ServiceID=123458&OrderID=123402816&Hash=5432d69a66d721b2f5f785432bf5a1fc1b913bdb3bba465856a5c228fe95c1f8
```

Powiadomienia natychmiastowe (ITN)

Opis powiadomień natychmiastowych

System przekazuje powiadomienia o zmianie statusu transakcji niezwłocznie po otrzymaniu takiej informacji z Kanału Płatności, a komunikat zawsze dotyczy pojedynczej transakcji. Potwierdzenia przesyłane są przez System płatności online na adres na serwerze Serwisu Partnera, ustalony w trakcie dodawania konfiguracji Serwisu Partnera.

UWAGA: Domena musi być publiczna i dostępna przez System. Domena musi być zabezpieczona ważnym certyfikatem, wystawionym przez publiczny urząd certyfikacji (Certificate authority) Serwer musi się przedstawiać pełnym łańcucha certyfikatów (Chain of Trust) Komunikacja musi się odbywać w oparciu o protokół TLS w wersji 1.2 lub 1.3 *Inne formy zabezpieczenia połączenia np. VPN, mTLS muszą być uzgadniane indywidualnie z osobą odpowiedzialną za wdrożenie.

Przykład:

```
https://sklep_nazwa/odbior_statusu
```

Powiadomienie o zmianie statusu transakcji wejściowej polega na wysłaniu przez System dokumentu XML zawierającego nowe statusy transakcji.

Dokument wysyłany jest protokołem HTTPS (domyślnie port 443) – metodą POST, jako parametr HTTP o nazwie transactions. Parametr ten zapisany jest mechanizmem kodowania transportowego Base64.

Format dokumentu (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
  <transactionList>
    <serviceID>ServiceID</serviceID>
    <transactions>
      <transaction>
        <orderID>OrderID</orderID>
        <remoteID>RemoteID</remoteID>
        <amount>999999.99</amount>
        <currency>PLN</currency>
        <gatewayID>GatewayID</gatewayID>
        <paymentDate>YYYYMMDDhhmmss</paymentDate>
        <paymentStatus>PaymentStatus</paymentStatus>
      <paymentStatusDetails>PaymentStatusDetails</paymentStatusDetails>
    </transaction>
  </transactions>
  <hash>Hash</hash>
</transactionList>
```

UWAGA: Węzeł **transactions** może zawierać jedynie jeden węzeł **transaction** (a więc powiadomienie dotyczy zawsze jednej transakcji). Wartości elementów **orderID** i **amount** dotyczących każdej z transakcji są identyczne z wartościami odpowiadających im pól, podanymi przez Serwis Partnera przy rozpoczęciu danej transakcji. Wyjątkiem są tutaj modele, w których prowizja doliczana jest do kwoty transakcji. Wówczas wartość amount w ITN jest powiększona o tą prowizję. Walidację kwot można wtedy przeprowadzić na podstawie opcjonalnego pola ITN startAmount. Należy jednak zgłosić zapotrzebowanie na to pole podczas integracji.

Lista zwracanych parametrów dla powiadomień natychmiastowych

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceID	TAK	string{1,10}	Identyfikator Serwisu Partnera, nadawany w trakcie rejestracji usługi, jednoznacznie identyfikuje Serwis Partnera w Systemie płatności online.
2	orderID	TAK	string{1,32}	Identyfikator transakcji nadany w Serwisie Partnera i przekazany w starcie transakcji.
3	remotelID	TAK	string{1,20}	Alfanumeryczny identyfikator transakcji nadany przez System płatności online. Warto go zapisać przy zamówieniu na potrzeby dalszej obsługi (dla wielu transakcji z tym samym OrderID , dla zwrotów itp.). Sytuacja taka może mieć miejsce np. w przypadku, gdy Klient zmieni Kanał Płatności, wywoła ponownie ten sam start transakcji z historii przeglądarki itp. System umożliwia blokowanie takich przypadków, jednak opcja nie jest zalecana (nie byłoby możliwe opłacenie porzuconej transakcji).
5	amount	TAK	amount	Kwota transakcji. Jako separator dziesiętny używana jest kropka - '.' Format: 0.00; maksymalna długość: 14 cyfr przed kropką i 2 po kropce.
6	currency	TAK	string{1,3}	Waluta transakcji.
7	gatewayID	NIE	string{1,5}	Identyfikator Kanału Płatności, za pomocą, którego Klient uregulował płatność.

kolejność Hash	nazwa	wymagany	typ	opis
8	paymentDate	TAK	string{14}	Moment zautoryzowania transakcji, przekazywany w formacie YYYYMMDDhhmmss. (Czas CET)
9	paymentStatus	TAK	enum	Status autoryzacji transakcji, przyjmuje wartości (opis zmian statusów dalej): <ul style="list-style-type: none"> • PENDING - transakcja rozpoczęta. • SUCCESS - poprawna autoryzacja transakcji, Serwis Partnera otrzyma środki za transakcje - można wydać towar/usługę. • FAILURE - transakcja nie została zakończona poprawnie.
10	paymentStatusDetails	NIE	string{1,64}	Szczegółowy status transakcji, wartość może być ignorowana przez Serwis Partnera.
nd.	hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis.

WSKAZÓWKA: Element **hash** (komunikatu) służy do autentykacji dokumentu. Opis sposobu obliczania skrótu znajduje się w części [Bezpieczeństwo transakcji](#).

Odpowiedź na powiadomienie natychmiastowe

W odpowiedzi na powiadomienie oczekiwany jest status HTTP 200 (OK) oraz tekst w formacie XML (niekodowany Base64), zwracany przez Serwis Partnera w tej samej sesji HTTP, zawierający potwierdzenie otrzymania statusu transakcji.

Struktura potwierdzenia (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
  <confirmationList>
    <serviceID>ServiceID</serviceID>
    <transactionsConfirmations>
      <transactionConfirmed>
        <orderID>OrderID</orderID>
        <confirmation>Confirmation</confirmation>
      </transactionConfirmed>
    </transactionsConfirmations>
    <hash>Hash</hash>
  </confirmationList>
```

Opis pól potwierdzenia dla powiadomień natychmiastowych

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceID	TAK	string{1,10}	Identyfikator Serwisu Partnera pochodzący z komunikatu.
2	orderID	TAK	string{32}	Identyfikator transakcji, nadany w Serwisie Partnera i przekazany w starcie transakcji, pochodzący z komunikatu.
3	confirmation	TAK	string{1,25}	<p>Element służy do przekazania stanu weryfikacji autentyczności transakcji przez Serwis Partnera. Wartość elementu wyznaczana jest przez sprawdzenie poprawności wartości parametru serviceID oraz currency, porównanie wartości pól orderID i amount w komunikacie powiadomienia oraz w komunikacie rozpoczynającym transakcję, a także weryfikację zgodności wyliczonego skrótu z parametrami komunikatu z wartością przekazaną w polu hash komunikatu. Wyjątkiem są modele, w których prowizja doliczana jest do kwoty transakcji. Wówczas wartość amount w ITN jest powiększona o tą prowizję. Walidację kwot można wtedy przeprowadzić na podstawie opcjonalnego pola ITN startAmount. Należy jednak zgłosić zapotrzebowanie na to pole podczas integracji.</p> <p>Przewidziano dwie wartości elementu confirmation:</p> <ul style="list-style-type: none"> • CONFIRMED - wartości parametrów w obu komunikatach oraz parametr hash są zgodne - transakcja autentyczna; • NOTCONFIRMED - wartości w obu komunikatach są różne lub niezgodność hash - transakcja nieautentyczna;
nd.	hash	TAK	string{1,128}	Element hash (w odpowiedzi na komunikat) służy do autentykacji odpowiedzi i liczony jest z wartości parametrów odpowiedzi. Opis sposobu obliczania skrótu znajduje się w części Bezpieczeństwo transakcji .

W wypadku braku poprawnej odpowiedzi na wysłane powiadomienia, System podejmie kolejne próby przekazania najnowszego statusu transakcji po upływie określonego czasu. Serwis Partnera powinien wykonywać własną logikę biznesową (np. mail z potwierdzeniem), jedynie po pierwszym komunikacie o danym statusie płatności.

WSKAZÓWKA: Warto zapoznać się ze *Schematem ponawiania komunikatów ITN/ISTN/IPN/RPAN/RPDN*.

Szczegółowy opis zachowania i zmiany statusów płatności (**paymentStatus**)

Wybór przez Klienta metody płatności każdorazowo spowoduje wysłanie statusu **PENDING**. W kolejnym komunikacie ITN system dostarczy status **SUCCESS** lub **FAILURE**.

UWAGA Status **PENDING** nie zostanie wysłany, jeśli:

- klient zrezygnuje lub powróci z ekranu listy metod płatności bez wybrania konkretnej metody. W tym wypadku od razu zostanie wysłany status **FAILURE**. Nie pojawi się status **PENDING**, ponieważ klient nie rozpoczął procesu płatności.
- status finalny (**SUCCESS** lub **FAILURE**) zostanie dostarczony przed wysłaniem ITN ze statusem **PENDING**.

Dla pojedynczej transakcji (o unikatowych parametrach **OrderID** oraz **RemoteID**) nie może nastąpić zmiana statusu **SUCCESS** na **PENDING** lub **SUCCESS** na **FAILURE**.

W każdym przypadku może nastąpić zmiana statusu szczegółowego – **paymentStatusDetails** (kolejne komunikaty o zmianie statusu szczegółowego są jedynie informacyjne i nie powinny prowadzić do ponownego wykonania opłacanej usługi/wysyłki produktu itp.).

W szczególnych przypadkach użycia może nastąpić zmiana statusu:

a) **FAILURE** na **SUCCESS** (np. po zatwierdzeniu przez konsultanta AP transakcji wpłaconej z błędną kwotą. Takie zachowania wymaga specjalnych uzgodnień biznesowych i nie jest włączone domyślnie),

b) **SUCCESS** na **FAILURE** (np. po wywołaniu wielu transakcji z tym samym **OrderID**, ale różnym **RemoteID**). Taki przypadek występuje w sytuacji rozpoczęcia przez Klienta wielu płatności z tym samym **OrderID** (np. Klient zmienia decyzję, jakim Kanałem płatności chce opłacić transakcję). Każda z rozpoczętych przez niego płatności generuje ITNy i poszczególne transakcje Partner powinien rozróżnić na podstawie parametru **RemoteID**. Ponieważ czas otrzymania statusu **FAILURE** może być bardzo różny, może się zdarzyć otrzymanie takiego statusu po odebraniu **SUCCESS** (oczywiście z innym **RemoteID**). W takim wypadku, komunikat ITN powinien być potwierdzany, ale nie powinien pociągać za sobą anulowania statusu transakcji w systemie Partnera.

Obsługa statusów transakcji z ITN - model uproszczony

W modelu, w którym nie jest potrzebne powiadamianie mailem/smssem Klienta o statusach innych niż **SUCCESS**, można ograniczyć ilość informacji zapisywanych do bazy Serwisu oraz śledzenie zmian **RemoteID**.

Wystarczy:

- dla statusów innych niż **SUCCESS**, za każdym razem potwierdzać ITN poprawną strukturą odpowiedzi, statusem **CONFIRMED** oraz poprawnie policzoną wartością pola Hash,
- w przypadku otrzymania **pierwszego** statusu **SUCCESS**, dodać również aktualizację statusu, jego czasu i RemoteID w bazie Serwisu oraz realizację procesów biznesowych (powiadomienia do Klienta o zmianie statusu, wykonania opłacanej usługi/wysyłki produktu itp.),
- w przypadku otrzymania kolejnego statusu **SUCCESS**, za każdym razem potwierdzać ITN poprawną strukturą odpowiedzi, statusem **CONFIRMED** oraz poprawnie policzoną wartością pola Hash, bez aktualizacji bazy Serwisu oraz bez procesów biznesowych.

Obsługa statusów transakcji z ITN - model pełny

W modelu, w którym potrzebna jest cała historia zmian statusów transakcji i/lub powiadomianie Klienta o ważniejszych zmianach statusów transakcji należy zastosować logikę przybliżoną do poniższego opisu.

Dotychczasowy ogólny Status Płatności	PaymentStatus w ITN	Różne RemoteID	Proces biznesowy (powiadomienia do Klienta o zmianie statusu)	Proces biznesowy (wykonanie opłacanej usługi/wyylki produktu itp.)	Zawartość pola confirmation odpowiedzi	Aktualizacja ogólnego statusu transakcji, jej czasu i wartości RemoteID	Uwagi
Brak	Pending	Nd	Tak	Nie	CONFIRMED	Tak	
Brak	Failure	Nd	Tak	Nie	CONFIRMED	Tak	
Brak	Sucess	Nd	Tak	Tak	CONFIRMED	Tak	
Pending	Pending	Nie	Nie	Nie	CONFIRMED	Nie	
Pending	Failure	Nie	Tak	Nie	CONFIRMED	Tak	
Pending	Sucess	Nie	Tak	Tak	CONFIRMED	Tak	
Failure	Pending	Nie	Nie	Nie	CONFIRMED	Nie	
Failure	Failure	Nie	Nie	Nie	CONFIRMED	Nie	
Failure	Sucess	Nie	Tak	Tak	CONFIRMED	Tak	
Sucess	Pending	Nie	Nie	Nie	CONFIRMED	Nie	Niemozliwe
Sucess	Failure	Nie	Nie	Nie	CONFIRMED	Nie	Niemozliwe
Sucess	Sucess	Nie	Nie	Nie	CONFIRMED	Nie	
Pending	Pending	Tak	Nie	Nie	CONFIRMED	Nie	
Pending	Failure	Tak	Tak	Nie	CONFIRMED	Tak	
Pending	Sucess	Tak	Nie	Tak	CONFIRMED	Tak	
Failure	Pending	Tak	Nie	Nie	CONFIRMED	Tak	
Failure	Failure	Tak	Nie	Nie	CONFIRMED	Nie	
Failure	Sucess	Tak	Tak	Tak	CONFIRMED	Tak	
Sucess	Pending	Tak	Nie	Nie	CONFIRMED	Nie	Niemozliwe
Sucess	Failure	Tak	Nie	Nie	CONFIRMED	Nie	Niemozliwe
Sucess	Sucess	Tak	Nie	Nie	NOTCONFIRMED	Nie	Niemozliwe

Bezpieczeństwo transakcji

Opis bezpieczeństwa transakcji

W Systemie płatności online zastosowano kilka mechanizmów zwiększających bezpieczeństwo realizowanych przy jego użyciu transakcji. Transmisja między wszystkimi stronami transakcji realizowana jest z użyciem bezpiecznego połączenia opartego na protokole TLS z 2048-bitowym kluczem.

Dodatkowo, komunikacja zabezpieczana jest funkcją skrótu obliczoną z wartości pól komunikatu i współdzielonego klucza (sam klucz współdzielony przechowywany jest w Systemie w postaci zaszyfrowanej algorytmem AES-ECB).

Jako funkcja skrótu wykorzystywany jest algorytm SHA256 lub SHA512 (metoda ustalana na etapie konfigurowania danego Serwisu Partnera w Systemie płatności online). Domyślna funkcja to SHA256.

Obliczanie wartości funkcji skrótu

Opis sposobu obliczania wartości funkcji skrótu oraz przykłady obliczeń dla podstawowych komunikatów.

UWAGA: Przykłady nie uwzględniają wszystkich możliwych pól opcjonalnych, dlatego w razie występowania takich pól w konkretnym komunikacie, należy uwzględnić je w funkcji skrótu w kolejności zgodnej z numerem obok pola.

Sposób obliczania wartości funkcji skrótu - pole Hash

Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wartości pól, bez nazw parametrów, a pomiędzy kolejnymi (niepustymi) wartościami wstawiany jest separator (w postaci znaku |). Kolejność sklejania pól jest zgodna z kolejnością ich występowania na liście parametrów w dokumentacji.

WAŻNE! W przypadku braku opcjonalnego parametru w komunikacie lub w przypadku pustej wartości parametru, nie należy używać separatora!

Do powstałego w powyższy sposób łańcucha doklejany jest na jego końcu klucz, współdzielony między Serwis Partnera i System płatności online. Z tak powstałego łańcucha obliczana jest wartość funkcji skrótu i stanowi ona wartość pola Hash komunikatu.

Hash = funkcja(wartości_pola_1_komunikatu + "|" + wartości_pola_2_komunikatu + "|" + ... + "|" + wartości_pola_n_komunikatu + "|" + klucz_współdzielony);

Przykładowe obliczenia wartości funkcji skrótu podczas rozpoczęcia transakcji

Dane Serwisu Partnera:

ServiceID = 2

klucz_współdzielony = 2test2

Adres bramki https://host_bramki/sciezka

a. Rozpoczęcie transakcji.

Wywołanie POST bez koszyka, z parametrami:

ServiceID=2

OrderID=100

Amount=1.50

```
Hash=2ab52e6918c6ad3b69a8228a2ab815f11ad58533eed963dd990df8d8c3709d1
```

gdzie

```
Hash=SHA256("2|100|1.50|2test2")
```

b. Rozpoczęcie transakcji. Wywołanie POST z koszykiem.

WSKAZÓWKA: Opcja szczegółowo omówiona w części [Koszyk produktów](#).

ServiceID = 2

OrderID = 100

Amount = 1.50

Product (opisany niżej)

klucz_współdzielony = 2test2

Koszyk produktów (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
  <productList>
    <product>
      <subAmount>1.00</subAmount>
      <params>
        <param name="productName" value="Nazwa produktu 1" />
      </params>
    </product>
    <product>
      <subAmount>0.50</subAmount>
      <params>
        <param name="productType" value="ABCD" />
        <param name="ID" value="EFGH" />
      </params>
    </product>
  </productList>
```

Po zakodowaniu funkcją base64 otrzymujemy wartość parametru Product:

```
PD94bWwgdMVyc2lvcj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz48cHJvZHVjdExp3Q+PHByb2R1Y3Q+PHN1YkFt
b3VudD4xLjAwPC9zdWJBbW91bnQ+PHBhcmFtcz48cGFyYW0gYmFtZT0icHJvZHVjdE5hbWUiIHZhbHVlPSJ0YXp3
YSBwcm9kdWt0dSaxIiAvPjwvcGFyYW1zPjwvcHJvZHVjdD48cHJvZHVjdD48c3ViQW1vdW50PjAuNTA8L3N1YkFt
b3VudD48cGFyYW1zPjxwYXJhbSBuYW1lPSJwcm9kdWN0VHlwZSIgdmFsdWU9IkFCQ0QiIC8+PHBhcmFtIG5hbWU9
Ik1EiB2Ywx1ZT0iRUZHSCiGZ48L3BhcmFtcz48L3Byb2R1Y3Q+PC9wcm9kdWN0TGZldD4=
```

Wartość Hash liczona jest w następujący sposób:

```
Hash=SHA256("2|100|1.50|PD94bWwgdMVyc2lvcj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz48cHJvZHVjdExp
c3Q+PHByb2R1Y3Q+PHN1YkFtb3VudD4xLjAwPC9zdWJBbW91bnQ+PHBhcmFtcz48cGFyYW0gYmFtZT0icHJvZHVj
dE5hbWUiIHZhbHVlPSJ0YXp3YSBwcm9kdWt0dSaxIiAvPjwvcGFyYW1zPjwvcHJvZHVjdD48cHJvZHVjdD48c3Vi
QW1vdW50PjAuNTA8L3N1YkFtb3VudD48cGFyYW1zPjxwYXJhbSBuYW1lPSJwcm9kdWN0VHlwZSIgdmFsdWU9IkFC
Q0QiIC8+PHBhcmFtIG5hbWU9Ik1EiB2Ywx1ZT0iRUZHSCiGZ48L3BhcmFtcz48L3Byb2R1Y3Q+PC9wcm9kdWN0
TGZldD4=|2test2")
```

Przykładowe obliczenia wartości funkcji skrótu podczas powrotu Klienta do Serwisu Partnera

Dane Serwisu Partnera:

ServiceID = 2

klucz_współdzielony = 2test2

```
<https://sklep_nazwa/strona_powrotu?ServiceID=2>&OrderID=100&Hash=254eac9980db56f425acf8a9df715cbd6f56de3c410b05f05016630f7d30a4ed
```

gdzie

Hash=SHA256("2|100|2test2")

Przykładowe obliczenia wartości funkcji skrótu w komunikacie ITN

Dane Serwisu Partnera:

serviceID = 1

klucz_współdzielony = 1test1

ITN (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
  <transactionList>
    <serviceID>1</serviceID>
    <transactions>
      <transaction>
        <orderID>11</orderID>
        <remoteID>91</remoteID>
        <amount>11.11</amount>
        <currency>PLN</currency>
        <gatewayID>1</gatewayID>
        <paymentDate>20010101111111</paymentDate>
        <paymentStatus>SUCCESS</paymentStatus>
        <paymentStatusDetails>AUTHORIZED</paymentStatusDetails>
      </transaction>
    </transactions>
    <hash>a103bfe581a938e9ad78238cfc674ffafdd6ec70cb6825e7ed5c41787671efe4</hash>
  </transactionList>
```

gdzie

Hash=SHA256("1|11|91|11.11|PLN|1|20010101111111|SUCCESS|AUTHORIZED|1test1")

Przykładowa odpowiedź (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<confirmationList>
  <serviceID>1</serviceID>
  <transactionsConfirmations>
    <transactionConfirmed>
      <orderID>11</orderID>
      <confirmation>CONFIRMED</confirmation>
    </transactionConfirmed>
  </transactionsConfirmations>
  <hash>c1e9888b7d9fb988a4aae0dfbff6d8092fc9581e22e02f335367dd01058f9618</hash>
</confirmationList>

```

gdzie wartość

```
Hash=SHA256("1|11|CONFIRMED|1test1");
```

Przykładowe obliczenia wartości funkcji skrótu w odpytaniu o listę Kanałów Płatności

Dane Serwisu Partnera:

serviceID = 1

messageID = cfb91538ad854d74813ea76893cc020c

klucz_współdzielony = 1test1

gdzie wartość

```
Hash=SHA256("1|cfb91538ad854d74813ea76893cc020c|1test1");
```

Odpowiedź na powyższe wywołanie może być następująca:

```

<?xml version="1.0" encoding="UTF-8"?>
  <list>
    <serviceID>1</serviceID>
    <messageID>cfb91538ad854d74813ea76893cc020c</messageID>
    <gateway>
      <gatewayID>19</gatewayID>
      <gatewayName>Przelew PKOBP</gatewayName>
      <gatewayType>Szybki Przelew</gatewayType>
      <bankName>INTELIGO</bankName>
      <iconURL>https://host/sciezka/19.png</iconURL>
      <statusDate>2015-10-14 12:12:31</statusDate>
    </gateway>
    <gateway>
      <gatewayID>106</gatewayID>
      <gatewayName>platnosc testowa PG</gatewayName>
      <gatewayType>PBL</gatewayType>
      <bankName>NONE</bankName>
      <statusDate>2015-10-14 12:12:31</statusDate>
    </gateway>
    <hash>8b69fcfac99ab7113b6b6c60e42fd6ec316a161027bcf2eb63bf6a5ff595fcd3</hash>
  </list>

```

gdzie wartość

```
Hash=SHA256("1|cfb91538ad854d74813ea76893cc020c|19|Przelew PK0BP|Szybki  
Przelew|INTELIG0|https://host/sciezka/19.png|2015-10-14 12:12:31|106|platnosc testowa  
PG|PBL|NONE|2015-10-14 12:12:31|1test1");
```

Rejestracja i obsługa Serwisów w oparciu o Formularz Integratorski

Schemat działania usługi dodawania i edycji Serwisów

Niniejszy rozdział opisuje zasady związane z wymianą komunikatów pomiędzy AP, a Platformą Integratora w ramach funkcjonalności dodawania i edycji Serwisów, która domyślnie odbywa się za pomocą REST API oraz opcjonalnie przy użyciu Web-Services protokołu SOAP (Usługa udostępnia swoją definicję w postaci dokumentu WSDL).

Partner udostępnia na swojej Platformie odnośnik, którego wybór przez Klienta powoduje wysłanie komunikatu do AP celem otrzymania linku kierującego do Formularza Integratorskiego przygotowanego przez AP (efekty wizualne takie jak kolorystyka, czy logo Integratora na formularzu ustalone są podczas integracji).

UWAGA Istnieje również możliwość osadzenia Formularza Integratorskiego przygotowanego przez Autopay bezpośrednio na stronie internetowej Integratora (w Iframe). Do tego celu wykorzystywany jest element języka HTML o nazwie Iframe. Wykorzystanie tego typu rozwiązania wiąże się z dodatkowym nakładem prac po stronie Integratora, jednak umożliwia pozostanie Partnera na stronie www Integratora w całym procesie rejestracji/edycji sklepu. Rozwiązanie zostało szczegółowo opisane w rozdziale pt. "**Widget Onboardingowy**".

Dane zebrane na formularzu (po jego wypełnieniu przez Klienta) są przetwarzane przez AP, gdzie w zależności od typu formularza wykonywana jest rejestracja/edycja/dodanie kolejnego serwisu. Po pozytywnym przejściu tego kroku dane konfiguracyjne serwisu oraz linki weryfikacyjne (przy edycji danych niewrażliwych linki się nie pojawiają) wysyłane są asynchronicznie do Integratora kanałami ustalonymi w trakcie integracji. Równolegle odbywa się również wysyłka do Klienta wiadomości email zawierającej linki do płatności weryfikacyjnej (istnieje możliwość wyłączenia tej wysyłki celem zastąpienia jej komunikacją odbywającą się bezpośrednio przez Partnera).

Klient w tym czasie zostaje automatycznie przekierowany na stronę powrotu Integratora (adres zostaje ustalony w trakcie integracji) lub zostaje mu zaprezentowana strona z podziękowaniem za skorzystanie z usługi, na której opcjonalnie mogą znaleźć się linki do płatności weryfikacyjnych i/lub link do Platformy Integratora.

Po opłaceniu przez Klienta transakcji weryfikacyjnej, AP sprawdza poprawność danych zadeklarowanych przez tegoż Klienta podczas rejestracji serwisu. W przypadku nadania przez AP pozytywnego statusu weryfikacji, usługa płatności w serwisie zostaje aktywowana, a do Klienta

zostaje wysłana wiadomość z regulaminem zaakceptowanym przez niego na formularzu rejestracyjnym.

WAŻNE! Produkcyjna wersja usługi znajduje się za firewallem. Dostęp jest przydzielany dla skończonej i zdefiniowanej w trakcie integracji puli IP. Nie dotyczy to środowiska testowego.

WAŻNE! Dla jednego Integratora na danym środowisku (test/produkcja) przewidziany jest jeden identyfikator Platformy (PlatformID) oraz klucz współdzielony wykorzystywany do budowania hasha dla wszystkich komunikatów wymienianych pomiędzy Integratorem a AP w obrębie procesu rejestracji i edycji serwisów.

WAŻNE! Wygenerowany przez AP link prowadzący do formularza rejestracji/edycji danych serwisu ma domyślny czas ważności 24 godzin. Wartość ta może być zmieniona na prośbę Integratora.

WAŻNE! Niedopuszczalne jest udostępnianie w jakiegokolwiek formie (również w kodzie uruchamianej na serwerze osób trzecich) danych autoryzacyjnych do usługi udostępnianej przez AP (PlatformID/klucz współdzielony).

WAŻNE! Jeśli podczas rejestracji lub edycji sklepu klient wybierze kilka walut, przy pomocy których mogą być wykonywane płatności w sklepie, każda z tych walut wraz z przypisanym do niej rachunkiem rozliczeniowym musi zostać osobno zweryfikowana, poprzez przelew weryfikacyjny.

WAŻNE! Na formularzu służącym do edycji danych, przed wyświetlaniem na nim aktualnych danych Serwisu musi nastąpić weryfikacja tożsamości. W tym celu Klientowi zaprezentowane są dwa kanały weryfikacji: wiadomość sms lub email. Po wyborze jednego z nich, do Klienta jest wysyłany kod weryfikacyjny (do tego celu wykorzystany jest numer telefonu lub adres email podany przez Klienta w procesie rejestracji), który musi zostać przez niego wpisany na dodatkowym formularzu. Po poprawnym uzupełnieniu tego pola i jego weryfikacji przez AP Klient otrzymuje dostęp do formularza edycji danych wraz z całą jego zawartością.

UWAGA: W części [Obsługa transakcji i rozliczeń](#) opisane zostały funkcjonalności i sposób ich integracji w zakresie związanym z obsługą płatności dla Serwisu oraz usługami powiązanymi z obsługą płatności np. schematem rozliczeń.

W modelu Integratora nie są dostępne poniższe funkcje części transakcyjnej:

- a) Dane wymieniane podczas integracji
- b) Koszyk produktów

Dane wymieniane podczas integracji Formularza Integratorskiego

Dane wymieniane w środowisku testowym

Dane przekazywane przez AP do Partnera:

- Adres Systemu płatności online
- Adres usługi (adresy, na których wystawione są poszczególne metody, z których może skorzystać Integrator)
- PlatformID
- ServiceID (dla usługi Przelewu weryfikacyjnego)
- Klucz współdzielony dla usługi rejestracji i edycji
- Klucz współdzielony dla usługi Przelewu weryfikacyjnego
- Mechanizm funkcji skrótu
- Adres IP, z którego wysyłane są ITNy
- Adres do panelu administracyjnego dla integratora (opcja)
- Login
- Hasło

Dane przekazywane przez Partnera do AP:

- Adres ITN po przelewie weryfikacyjnym
- Adres powrotu po przelewie weryfikacyjnym
- Informacja o walutach jakie mają być dostępne dla sklepów integratora
- Informacja o kanałach wysyłki konfiguracji Serwisu oraz Linków Weryfikacyjnych
- Adres email Integratora w przypadku wysyłki konfiguracji serwisów poprzez adres email
- Adres, pod którym Integrator wystawi usługi do odbierania komunikatów ICN
- Informacja w przypadku zmiany domyślnego czasu ważności linka prowadzącego do formularza rejestracji/edycji danych serwisu (domyślnie 24 godziny)

Dane przekazywane w środowisku produkcyjnym

Przez AP do Partnera:

- Adres Systemu płatności online
- Adres usługi (adresy, na których wystawione są poszczególne metody, z których może skorzystać Integrator)
- PlatformID
- ServiceID (dla usługi Przelewu weryfikacyjnego)
- Klucz współdzielony dla usługi rejestracji i edycji
- Klucz współdzielony dla usługi Przelewu weryfikacyjnego
- Mechanizm funkcji skrótu
- Adres IP, z którego wysyłane są ITNy
- Adres do panelu administracyjnego dla integratora (opcja)
- Login
- Hasło

Przez Partnera do AP:

- Adres IP, z którego następuje połączenie do usług wystawionych przez Autopay
- Adres ITN po przelewie weryfikacyjnym

- Adres powrotu po przelewie weryfikacyjnym
- Adresy email dla raportów rozliczeniowych dla Integratora
- Informacja o walutach jakie mają być dostępne dla sklepów integratora
- Informacja o kanałach wysyłki konfiguracji Serwisu oraz Linków Weryfikacyjnych
- Adres email Integratora w przypadku wysyłki konfiguracji serwisów poprzez adres email
- Adres, pod którym Integrator wystawi usługę do odbierania komunikatów ICN
- Adres, pod którym integrator wystawi usługę do odbierania powiadomień o statusie kartowym (konieczne w przypadku, kiedy Integrator chce taką informację otrzymywać)
- Informacja w przypadku zmiany domyślnego czasu ważności linka prowadzącego do formularza rejestracji/edycji danych serwisu (domyślnie 24 godziny)

Pobranie przez Integratora linku do Formularza Integratorskiego

Wymiana komunikatów (w formacie JSON) pomiędzy AP, a Platformą Integratora, realizujących funkcjonalność pobierania linków do formularza rejestracji/edycji serwisów odbywa się za pomocą REST API. Dostęp do usługi zabezpieczony jest poprzez filtrowanie adresów IP.

Opis pól wysyłanych w komunikacie żądania do AP

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	platformId	TAK	integer	Stały unikalny identyfikator Platformy nadany przez System płatności online.
2	messageId	TAK	string{32}	Unikalny identyfikator żądania w ramach Platformy nadawany przez stronę inicjującą dany komunikat.
3	messageTime	TAK	dateTime	Czas wygenerowania komunikatu, komunikaty z czasem ustawionym późniejszym niż 5 minut od czasu serwera Systemu płatności online będą odrzucane. Warto ustawić czas komunikatu now()-1min, na wypadek rozsynchronizowania czasu na serwerach. Przykład: 2016-07-20T09:35:00.000 (komunikat wygenerowany w chwili 2016-07-20 09:36:00).
nd.	hash	TAK	string	Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wyłącznie wartości pól, bez nazw parametrów. Kolejność sklejania pól jest zgodna z kolejnością ich występowania na liście parametrów. Do powstałego w powyższy sposób łańcucha doklejany jest na jego końcu klucz, współdzielony między System Platformy i System płatności online. Z tak powstałego łańcucha obliczana jest wartość funkcji skrótu SHA256 i stanowi ona wartość pola Hash komunikatu. <i>Hash = SHA256(wartości_pól_komunikatu + klucz_współdzielony)</i>
nd.	formAction	TAK	string	Parametr wskazujący jaki link do formularza ma zostać zwrócony w odpowiedzi na przesłane żądanie. Dozwolone wartości: <ul style="list-style-type: none"> • REGISTRATION - zostanie zwrócony link do formularza rejestracyjnego • UPDATE - zostanie zwrócony link do formularza pozwalającego na edycję danych serwisu • ADD_SERVICE - zostanie zwrócony link do formularza pozwalającego na dodanie kolejnego serwisu do istniejącego akceptanta

kolejność Hash	nazwa	wymagany	typ	opis
nd.	formParams	TAK/NIE	string{32}	<p>Wymagalność zależy od konfiguracji Integratora. Jest to obiekt zawierający dodatkowe pola, będące dla AP informacją o tym, jakiej konfiguracji rejestrowanego Serwisu oczekuje Integrator. W przypadkach kiedy Integrator posiada pewne informacje o Partnerze może je przekazać w odpytaniu o link, po to żeby określić konfigurację serwisu Partnera, zmniejszyć ilość pól na formularzu rejestracji/edycji i nie wymagać od Partnera wpisywania kolejny raz tych samych danych. Dla wskazanych poniżej pól istnieje możliwość skonfigurowania ich widoczności w zależności od formAction (REGISTER/UPDATE). Oznacza to, że np. pole SERVICE_URL może być podane przez integratora w odpytaniu o link do formularza i wyświetlone na formularzu rejestracji jako edytowalne, ale na formularzu edycji może zostać ukryte. Dla formAction = ADD_SERVICE zachowanie pól jest zdefiniowane zawsze tak samo jak dla formAction = REGISTER.</p> <p>Aktualnie dostępne pola:</p> <ul style="list-style-type: none"> • SERVICE_URL - adres url sklepu. • COMMISSION_MODEL - lista modeli prowizyjnych z jakimi ma zostać zarejestrowany sklep (po jednym modelu prowizyjnym dla każdej z walut). Jeśli Integrator nie poda przy odpytaniu o link wartości modelu prowizyjnego danej waluty, a zostanie ona wybrana przez Klienta na formularzu, wówczas system przypisze domyślny model prowizyjny odpowiadający walucie, ustalony z Partnerem podczas integracji z systemem BM. • IS_CARDS_ENABLED - informacja o tym, czy Klientowi mają zostać uruchomione płatności kartowe na Serwisie. • RETURN_URL - adres URL, na który ma zostać przekierowany Klient po wykonaniu płatności; akceptowany jest jedynie protokół HTTPS. • ITN_URL - adres URL, na który mają być wysyłane natychmiastowe powiadomienia o statusach płatności. • IS_REFUNDS_ENABLED - informacja czy Partner chce udostępnić dla Serwisu opcję wypłat z Rachunku Płatniczego oraz zwrotów transakcji. • SERVICE_NAME - nazwa sklepu. • ALLOWED_CURRENCIES - lista walut, do których ma być ograniczony formularz (waluty sklepu dostępne dla klienta na formularzu edycji lub rejestracji). <p>Przykład Ograniczenie formularza do walut PLN, EUR: "formParams": {"ALLOWED_CURRENCIES" : ["PLN","EUR"]}</p> <p>UWAGA W przypadku odpytania o link do formularza bez podania parametru ALLOWED_CURRENCIES, na formularzu zostaną wyświetlone wszystkie domyślne waluty skonfigurowane dla Integratora.</p> <ul style="list-style-type: none"> • ACTIVITY_KIND - forma prawna; dopuszczalne wartości zostały opisane w dalszej części tego dokumentu. • PLATFORM - ID platformy, pod którą został wystawiony serwis. • NUMERIC_TRADE - branża, w której specjalizuje się Serwis Partnera • ECONOMIC_PURPOSE - oświadczenie Partnera, iż zawiera Umowę w następującym celu gospodarczym; dopuszczalne wartości zostały opisane w dalszej części tego dokumentu. • CONTACT_EMAIL - adres email do kontaktu. • COMPLAINT_EMAIL - adres email do reklamacji. • REPORT_EMAIL - adres email do raportów. • INVOICE_EMAIL - adres email do faktur. • PHONE - numer telefonu Partnera. • COMPANY_NAME - nazwa firmy. • KRS - numer krs. • REGISTRATION_DATE - data rejestracji firmy. • ADDRESS - ulica. • POSTAL_CODE - kod pocztowy. • CITY - miasto. • COUNTRY - państwo. • AVERAGE_SERVICE_TURNOVER - spodziewany obrót miesięczny. • AVERAGE_TRANSACTION_AMOUNT - średnia wartość transakcji. • NIP - numer identyfikacji podatkowej.

Przykład 1:

```
{
  "platformId":1,
  "messageId": "22111111112411111111111111111130",
  "messageTime": "2016-07-20T09:35:00.000",
```



```

        "SERVICE_URL":"https:// servivce-integrator-test.pl ",
        "COMMISSION_MODEL":{
            "PLN":"5"
        },
        "IS_CARDS_ENABLED":"TRUE"
    }
}

```

Opis pól komunikatu odpowiedzi do Integratora

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

WSKAZÓWKA: Poprawna odpowiedź – status **http 200**.

kolejność Hash	nazwa	wymagany	typ	opis
1	link	TAK	integer	Zawiera wygenerowany link do formularza rejestracyjnego.
2	messageld	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łańciskiego (np. na bazie UID), wartość pola musi być unikalna w obrębie Integratora.
3	formHash	TAK	string	Identyfikator linku do formularza, który służy Integratorowi do powiązania linku do formularza konkretnego Klienta z komunikatem wysyłanym po rejestracji, informującym o powstałej konfiguracji serwisu.
nd.	hash	TAK	string	Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wyłącznie wartości pól, bez nazw parametrów Kolejność sklejania pól jest zgodna z kolejnością.

WSKAZÓWKA: Odpowiedź z komunikatem błędu – status **http 400**.

kolejność Hash	nazwa	wymagany	typ	opis
nd.	errors	TAK/NIE		Zawiera wygenerowany link do formularza rejestracyjnego.
nd.	field	TAK	string	Wskazuje na pole, którego dotyczy błąd.
nd.	error	TAK	string	Słowny opis błędu.
nd.	errorCode	TAK	integer	Kod błędu – pełen wykaz błędów poniżej.

Przykład poprawnej odpowiedzi:

```
{
  "link": "https://integrator-form-domain/
e2287514541105a2eda2b85751e88be5998aec8c99cd83ba3073365ce1a243a1",
  "hash":
"f9e02e83fe50920ee2efb0d2322b6200a71f3afcc366893487ced7ad2330a610",
  "messageId": "111111111111111111111111111111111111229",
  "formHash":
"e2287514541105a2eda2b85751e88be5998aec8c99cd83ba3073365ce1a243a1"
}
```

Przykłady błędnej odpowiedzi:

```
{
  "errors": [
    {
      "field": "acceptorId",
      "error": "Value for field acceptorId required!",
      "errorCode": 6003
    }
  ]
}
>
{
  "errors": [
    {
      "field": "messageTime",
      "error": "Message time is outdated",
      "errorCode": 6016
    }
  ]
}
>
{
  "errors": [
    {
      "field": "hash",
      "error": "Invalid hash",
      "errorCode": 6000
    }
  ]
}
```

Kody błędów

Parametr	Kod błędu	Opis
hash	6000	Błędny hash.
messageld	6001	Wartość parametru messageld nie jest unikatowa.

Parametr	Kod błędu	Opis
acceptorId	6002	Został podany parametr acceptorId , który jest niedozwolony.
acceptorId	6003	Parametr acceptorId nie został podany a jest wymagany.
serviceld	6004	Został podany parametr serviceld , który jest niedozwolony.
acceptorId	6005	Akceptor z podanym Id nie istnieje.
serviceld	6006	Serwis z podanym Id nie istnieje.
servicelds	6007	Parametr servicelds zawiera więcej lub mniej niż jeden element.
servicelds	6008	Sklep jest aktualnie weryfikowany, nie jest obecnie możliwe wykonanie kolejnej edycji.
formParams	6009	Dla formularza edycji podano dodatkowe parametry które są niedozwolone.
formParams	6010	W żądaniu o link do rejestracji serwisu podano nieznanne parametry.
formParams	6011	W żądaniu o link do rejestracji nie podano wymaganego parametru.
formParams	6012	W żądaniu o link do rejestracji podano błędny format parametru.
formParams	6013	W żądaniu o link do rejestracji podano błędny CommissionModel .
formParams	6014	W żądaniu o link do rejestracji podano nieobsługiwaną walutę.
messageTime	6015	Błędny format daty wysłanej w odpytaniu o link do formularza.
messageTime	6016	Data wysłana w odpytaniu o link do formularza jest nie poza dopuszczalnym zakresem.
platformId	6017	Parametr platformId nie został podany a jest wymagany.
messageld	6018	Parametr messageld nie został podany a jest wymagany.
messageld	6019	Wartość parametru messageld ma niepoprawną długość.
formAction	6020	Niedozwolona wartość parametru formAction .
messageTime	6021	Parametr messageTime nie został podany a jest wymagany.
hash	6022	Parametr hash nie został podany a jest wymagany.
hash	6023	Podano błędną wartość parametru hash.
formAction	6024	Parametr formAction nie został podany a jest wymagany.
hash	6025	Podano niepoprawny protokół w którymś z adresów URL. Oczekiwany HTTPS.
formparams	6026	Podano błędny składniowo adres URL w którymś z parametrów.
-	6099	Wystąpił nieoczekiwany/nieokreślony błąd.

Widget onboardingowy

Jest to dodatkowa, oferowana przez Autopay opcja w procesie onboardingu sklepu, pozwalająca

Integratorowi na osadzenie formularza bezpośrednio na stronie internetowej Integratora. W tej wersji rozwiązania Formularz rejestracji/edycji sklepu nadal przygotowany i utrzymywany jest po stronie Autopay, z tą różnicą, że Partner cały proces przechodzi znajdując się w serwisie Integratora, z pominięciem przekierowania na domenę Autopay. Umieszczenie formularza na stronie Integratora odbywa się poprzez użycie elementu języka HTML o nazwie IFRAME. Integrator decydując się na tego typu rozwiązanie będzie dodatkowo zobowiązany do zaimplementowania funkcjonalności pozwalającej na odbieranie eventów wysyłanych przez iframe Autopay, niezbędnych do prawidłowego wyświetlania Formularza Onboardingowego w serwisie Integratora.

UWAGA Autopay zaleca osadzenie Widgetu onboardingowego w serwisie Integratora, którego strona www zabezpieczona jest certyfikatem SSL.

UWAGA Adres www do Formularza onboardingowego umieszczony w IFRAME na serwisie Integratora jest wartością zmienną (zmienia się parametr formHash w linku), dlatego należy o niego odpytać przed każdorazowym wyświetleniem strony z formularzem na serwisie Integratora.

Przykładowy kod strony HTML z wykorzystaniem widgetu onboardingowego

```
<!doctype html>
<html lang="pl">
<head>
  <meta charset="utf-8">
  <title>Example usage of Autopay onboarding widget</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style>
    body {
      padding: 0;
      margin: 0;
    }
    .container {
      width: 100%;
      padding-left: 15px;
      padding-right: 15px;
      max-width: 1200px;
      margin: 0 auto;
    }
    header {
      padding: 30px 0;
      border-bottom: 1px solid #ccc;
    }
    footer {
      padding: 30px 0;
      border-top: 1px solid #ccc;
    }
    iframe {
      width: 700px;
      border: 0;
    }
  </style>
</head>
<body>
  <header>
    <div class="container">
```

```

    INTEGRATOR PAGE HEADER
  </div>
</header>
<main>
  <section>
    <div class="container">
      <h1>Example usage of Autopay onboarding widget</h1>
      <p>integrator text before</p>
    </div>
  </section>
  <section>
    <div class="container">
      <iframe id="iframe"
src="?"r=quot;https://adres-formularza-onboardingowego/form/<formHash>"></iframe>
    </div>
  </section>
  <section>
    <div class="container">
      <p> integrator text after</p>
    </div>
  </section>
</main>
<footer>
  <div class="container">
    INTEGRATOR PAGE FOOTER
  </div>
</footer>
<script type="text/javascript">
  // wait for page to load everything
  document.addEventListener("DOMContentLoaded", function() {
    // create listener for widget events
    window.addEventListener('message', (e) => {
      // if event origin not matches autopay onboarding origin, then event not belongs
to widget
      if (!/onboarding\.autopay\.eu$/.test(e.origin) {
        return;
      }
      // if there is no data in event or data is not string, then event is not valid
      if (!e.data || typeof e.data !== 'string') {
        return;
      }
      let messageData;
      // parse event data string to JSON, if it fails, messageData will remain empty
      try {
        messageData = JSON.parse(e.data)
      } catch (err) {}
      if (!messageData) {
        return;
      }
      // listener for HEIGHT_CHANGE event, thanks to which the iframe window is at
full height and the scroll bar is not displayed
      if (messageData.event === 'HEIGHT_CHANGE') {
        document.getElementById('iframe').style.height = messageData.data + 'px'
      }
      // listener for SCROLL_TOP event, which scrolls page to iframe top, because
scroll can't happen in full height iframe window
      if (messageData.event === 'SCROLL_TOP') {
        const scrollTopPosition = window.scrollY +
document.getElementById('iframe').getBoundingClientRect()['y'];
        window.scrollTo({left: 0, top: scrollTopPosition, behavior: 'smooth'});
      }
      // listener for FORM_SUCCESS event, which provides necessary data to continue

```

```

onboarding process
  if (messageData.event === 'FORM_SUCCESS') {
    console.log('Verification links:', messageData.data.verificationLinks)
  }
})
})
</script>
</body>
</html>

```

Powrót Klienta do Platformy Integratora

Przekierowanie Klienta może nastąpić bezpośrednio po poprawnym wykonaniu rejestracji/edycji sklepu lub może być udostępnione na stronie z podziękowaniem w formie odnośnika.

Wysyłka danych konfiguracyjnych Serwisu oraz Linków Weryfikacyjnych (ICN)

Po wysłaniu formularza przez Klienta i finalizacji procesu rejestracji/edycji, AP musi wysłać dane konfiguracyjne Serwisu oraz Linków Weryfikacyjnych do Integratora. Może się to odbyć na kilka sposobów. Kanały wysyłki ustalane są z Integratorem w trakcie integracji.

Umożliwiamy dostarczenie powyższych informacji poprzez wymianę komunikatów w technologii REST, Web-Services lub poprzez wysyłkę wiadomości email pocztą elektroniczną (w formie pliku zabezpieczonego hasłem).

Każdy z kanałów wysyłek ma swój system ponawiania w przypadku nieudanej próby przesłania informacji do Integratora.

Ze względów bezpieczeństwa sugerujemy, aby wymiana informacji o danych konfiguracyjnych była wykonywana przy użyciu REST API (domyślna) lub Web-Services na zestawionym tunelu IPsec lub poprzez filtrowanie adresów IP.

Do powiązania komunikatu ICN (otrzymanego przez Integratora) z konkretną rejestracją Klienta z formularza służy pole **formHash**, które jest wysyłane zarówno w komunikacie ICN jak i w odpowiedzi na odpytanie o link do formularza. Dzięki temu Integrator posiada informację, dla której rejestracji otrzymał dane konfiguracyjne w komunikacie ICN.

Wysyłka danych konfiguracyjnych poprzez REST

Wymiana komunikatów pomiędzy AP, a Serwisem Partnera realizujących funkcjonalności dodawania i edycji Serwisów Partnerów odbywa się za pomocą REST API. Komunikaty wysyłane są w formacie JSON.

Opis pól wysyłanych w komunikacie żądania do Integratora

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	acceptorId	TAK	integer	Id akceptanta.
2	serviceId	TAK	integer	Id serwisu.

kolejność Hash	nazwa	wymagany	typ	opis
3	serviceKey	TAK	string	Sól do hasha przypisana do serwisu. Przy jej pomocy Integrator będzie generował hash wykorzystywany w komunikatach dotyczących procesu płatności i rozliczeń za transakcje.
4	link	TAK	string	Link do przelewu weryfikacyjnego. W komunikacie przekazywany w formie listy linków obiektu verificationLinks . W przypadku przekazania kilku linków weryfikacyjnych kolejność do wyliczenia hash powinna być taka sama jak kolejność występowania linków w komunikacie.
nd.	hash	TAK	integer	Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wyłącznie wartości pól, bez nazw parametrów. Kolejność sklejania pól jest zgodna z kolejnością ich występowania na liście parametrów. Przykład wyliczenia: <i>SHA256(wartość_acceptoridwartość_serviceldwartość_serviceKeywartość_link1wartość_link2sól_do_hash)</i>
nd.	verificationLinks	NIE		Obiekt przechowujący listę parametrów dotyczących linków weryfikacyjnych.
nd.	currency	NIE	string	Waluta, której dotyczy link weryfikacyjny.
nd.	panelLogin	NIE	string	Login Klienta do panelu administracyjnego.
nd.	panelAddress	NIE	string	Adres URL do panelu administracyjnego.
nd.	formHash	TAK	string	Identyfikator linku do formularza, który służy Integratorowi do powiązania linku do formularza konkretnego Klienta z komunikatem wysyланym po rejestracji, informującym o powstałej konfiguracji serwisu.
nd.	method	NIE	string	Informacja o tym, jakiego typu formularza dotyczy wysłana konfiguracja. Dopuszczalne wartości: REGISTER UPDATE ADD_SERVICE

Przykład 1.

```

{
  "acceptorID":11111,
  "serviceID":22222,
  "serviceKey":"sa22a2729462f643cf4c989dddcf226b99b3a6bda11db43a433ab31a7ec2abe925",
  "hash":"580a285b9bf95e60d4eaeb470d32858a5f0191a2a51b40a18ee7612fa9ced187",
  "verificationLinks ":[
    {
      "link":"sampleLink",
      "currency":"PLN"
    }
  ],
  "panelLogin":"sample panel login",
  "panelPassword":"sample panel password",
  "panelAddress":"sample panel address",
  "formHash":"sample form hash",
  "method":"REGISTER"
}

```

Opis pól komunikatu odpowiedzi do AP

nazwa	wymagany	typ	opis
resultStatus	TAK	string	Status odpowiedzi. Dopuszczalne wartości: OK ERROR
description	TAK	string	Dodatkowy opis dla odpowiedzi.

```
{
    "resultStatus": "OK",
    "description": "sample description"
}
```

Wysyłka danych konfiguracyjnych poprzez Web-Services

Wymiana komunikatów pomiędzy AP, a Serwisem Partnera realizujących funkcjonalności dodawania i edycji Serwisów Partnerów odbywa się za pomocą technologii Web-Services protokołu SOAP.

Usługa ta udostępnia swoją definicję w postaci dokumentu WSDL (Web Service Definitions Language), dostarczanego przez AP podczas integracji.

Opis pól wysyłanych w komunikacie żądania do Integratora - InstantConfigurationNotificationRequest

```
<xsd:element name="InstantConfigurationNotificationRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:elementname="InstantConfigurationNotification"
type="tns: InstantConfigurationNotification "/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:complexType name=" InstantConfigurationNotification ">
  <xsd:sequence>
    <xsd:sequence>
      <xsd:element name="AcceptorID" type="xsd:int"/>
      <xsd:element name="ServiceID" type="xsd:int"/>
      <xsd:element name="ServiceKey" type="xsd:string"/>
      <xsd:element name="Hash" type="xsd:string"/>
      <xsd:element minOccurs="0" name="VerificationLinks"
type="tns:VerificationLinks"/>
      <xsd:element minOccurs="0" name="PanelLogin"
type="xsd:string"/>
      <xsd:element minOccurs="0" name="PanelPassword"
type="xsd:string"/>
      <xsd:element minOccurs="0" name="PanelAddress"
type="xsd:string"/>
      <xsd:element name="FormHash" type="xsd:string"/>
      <xsd:element name="Method" type="tns:Method"/>
    </xsd:sequence>
  </xsd:sequence>
</xsd:complexType>
```

```

<xsd:complexType name="VerificationLink">
  <xsd:sequence>
    <xsd:element name="Currency" type="xsd:string"/>
    <xsd:element name="Link" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

```

```

<xsd:complexType name="VerificationLinks">
  <xsd:sequence>
    <xsd:element maxOccurs="unbounded" name="VerificationLink"
type="tns:VerificationLink"/>
  </xsd:sequence>
</xsd:complexType>

```

```

<xsd:simpleType name="Method">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="REGISTER"/>
    <xsd:enumeration value="UPDATE"/>
    <xsd:enumeration value="ADD_SERVICE"/>
  </xsd:restriction>
</xsd:simpleType>

```

Opis pól komunikatu

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	Acceptorld	TAK	integer	Id akceptanta.
2	ServiceId	TAK	integer	Id serwisu.
3	ServiceKey	TAK	string	Sól do hasha przypisana do serwisu. Przy jej pomocy Integrator będzie generował hash wykorzystywany w komunikatach dotyczących procesu płatności i rozliczeń za transakcje.
4	Link	NIE	string	Link do przelewu weryfikacyjnego.
nd.	Hash	TAK	integer	Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wyłącznie wartości pól, bez nazw parametrów. Kolejność sklejania pól jest zgodna z kolejnością ich występowania na liście parametrów. Przykład wyliczenia: SHA256(wartość_acceptoridwartość_serviceworldwartość_serviceKeywartość_link1wartość_link2sól_do_hash)

kolejność Hash	nazwa	wymagany	typ	opis
nd.	VerificationLinks	NIE	string	Obiekt przechowujący listę obiektów VerificationLink dotyczących Linków Weryfikacyjnych.
nd.	VerificationLink	NIE	string	Obiekt przechowujący parametry dotyczące Linków Weryfikacyjnych.
nd.	Link	NIE	string	Link do przelewu weryfikacyjnego.
nd.	Currency	NIE	string	Waluta, której dotyczy Link Weryfikacyjny.
nd.	PanelLogin	NIE	string	Login Klienta do panelu administracyjnego.
nd.	PanelPassword	NIE	string	Tymczasowe hasło Klienta do panelu administracyjnego.
nd.	PanelAddress	NIE	string	Tymczasowe hasłoAdres URL do panelu administracyjnego.
nd.	FormHash	TAK	string	Identyfikator linku do formularza, który służy Integratorowi do powiązania linku do formularza konkretnego Klienta z komunikatem wysyłanym po rejestracji, informującym o powstałej konfiguracji serwisu.
nd.	Method	NIE	string	Informacja o tym, jakiego typu formularza dotyczy wysłana konfiguracja. Dopuszczalne wartości: REGISTER UPDATE ADD_SERVICE

Przykład 1.

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2: InstantConfigurationNotificationRequest
      xmlns:ns2="http://integrator/ws/"
      < InstantConfigurationNotification>
        <AcceptorID>11111</AcceptorID>
        <ServiceID>22222</ServiceID>
<ServiceKey>22a2729462f643cf4c989dddcf226b99b3a6bda11db43a433ab31a7ec2abe925</ServiceKey
>
<Hash>580a285b9bf95e60d4eae470d32858a5f0191a2a51b40a18ee7612fa9ced187</Hash>
        <VerificationLinks>
          <VerificationLink>
            <Currency>PLN</Currency>
            <Link>sampleLink</Link>
          </VerificationLink>
        </VerificationLinks>
        <FormHash>generated_hash</FormHash>
      <Method>REGISTER</Method>
    </ InstantConfigurationNotification>
  </ns2: InstantConfigurationNotificationRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Komunikat odpowiedzi do AP

```
<xsd:element name=" InstantConfigurationNotificationResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Result" type="tns:ResultStatus"/>
      <xsd:element name="Description" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

>

<xsd:simpleType name="ResultStatus">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="OK"/>
    <xsd:enumeration value="ERROR"/>
  </xsd:restriction>
</xsd:simpleType>
```

Opis pól komunikatu

nazwa	wymagany	typ	opis
ResultStatus	TAK	string	Status odpowiedzi. Dopuszczalne wartości: OK ERROR
Description	TAK	string	Dodatkowy opis dla odpowiedzi.

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2: InstantConfigurationNotificationResponse
      xmlns:ns2="http://integrator/ws/">
      <Result>OK</Result>
      <Description>sample description</Description>
    </ns2: InstantConfigurationNotificationResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Wysyłka danych konfiguracyjnych poprzez Email

Informacje o serwisie są wysyłane na adres email wskazany przez integratora w formie pliku zabezpieczonego hasłem. Numer telefonu, na który mają być wysyłane hasła jest ustalany z Integratorem w trakcie integracji.

Natychmiastowe powiadomienia o zmianie danych AML po przeprowadzonej weryfikacji w Systemie Autopay- ICN Verification

System Autopay umożliwia informowanie Integratora o zmianie danych AML Sklepu. Dzięki temu Integrator może po swojej stronie na bieżąco aktualizować dane, które zostały zmienione w Autopay i prezentować je Partnerowi.

Zostało to zrealizowane poprzez wysyłkę komunikatów w formacie JSON metodą HTTP POST. Poprawne działanie usługi wymaga zaimplementowania po stronie Integratora metody pozwalającej na przyjęcie wspomnianego komunikatu.

Struktura komunikatu

Struktura komunikatu wysyłanego do Integratora dzieli się na trzy główne węzły przedstawione w tabeli poniżej.

Nazwa węzła	opis
revisionHeader	Obiekt przechowujący dane dotyczące rewizji czyli zmiany w systemie, która została wprowadzona przez klienta w systemie Integratora
dataHeader	Obiekt przechowujący informacje służące do autentykacji komunikatu
currentData	Obiekt przechowujący informację aktualnych danych AML dotyczących Sklepu po stronie Autopay. Są to ostatnie pozytywnie zweryfikowane dane w systemie Autopay. Jeśli komunikat związany jest z rejestracją, która nie dostała pozytywnego statusu weryfikacji w polu revisionHeader.autopayVerificationStatus, to currentData = null (w systemie nie ma zweryfikowanych danych dla tego service/acceptanta)

Przykład

```
{
  "revisionHeader":{
    "orderId":"123_2d87806818cf319d7127ffb",
    "revisionId":"03d5627b-b9fe-40c0-ae32-5c9620d804ee",
    "autopayVerificationStatus":"POSITIVE"
  },
  "dataHeader":{
    "acceptorId":345,
    "dataTime":"2024-01-16T 09:16:13",
    "serviceId":123,
    "hash":"35a56e960b4a4650b727b098bafd1912cf4e3012c8927ec29b3af6408bc99199"
  },
  "currentData":{
    "company":{
      "name":"Testowa firma",
      "address":{
        "address":"ul Testowa 77",
        "postalCode":"80-462",
        "city":"Gdańsk",
        "country":"PL",
        "street":null,
        "houseNumber":null,
        "flatNumber":null
      },
      "nip":"8219663675",
      "regon":"955459555",
      "edg":"123",
      "krs":"1234567890",
    }
  }
}
```

```

"phone":{
  "currentValue":"+48123456789",
  "waitingValue":null
},
"representingPersons":[
  {
    "personName":{
      "firstName":"Jan",
      "lastName":"Kowalski"
    },
    "pesel":"PL",
    "documentData":{
      "number":"BKI332498",

      "type":"ID",
      "expirationDate":"2030-01-01",
      "country":"PL"
    },
    "citizenship":"32061970611",
    "birthData":{
      "date":"1932-06-19",
      "country":"PL"
    }
  }
],
"activityKind":"SP_Z00",
"beneficials":[
  {
    "personName":{
      "firstName":"Piotr",
      "lastName":"Nowak"
    },
    "citizenship":"PL",
    "pesel":"32061970611",
    "birthDate":"1932-06-19"
  }
],
"registrationDate":"2000-01-01",
"plenipotentiary":{
  "personName":{
    "firstName":"Jan",
    "lastName":"Kowalski"
  },
  "citizenship":"PL",
  "birthData":{
    "date":"1932-06-19",
    "country":"PL"
  },
  "pesel":"32061970611",
  "documentData":{
    "number":"BKI332498",
    "type":"ID",
    "expirationDate":"2030-01-01",
    "country":"PL"
  }
},
"physicalPerson":null,
"partners":[
  {
    "personName":{
      "firstName":"Jan",
      "lastName":"Kowalski"
    }
  }
]

```


Nazwa pola	Opis
dataHeader.serviceId	id service'u, którego dotyczą dane
dataHeader.acceptorId	id acceptanta, którego dotyczą dane
dataHeader.hash	wartość kontrolna weryfikująca integralność nagłówka liczona według wzoru SHA256(acceptorId dataTime serviceId secret_key) np: SHA256(345 2024-01-16T08:08:51.271 123 secret_key)
currentData.company	dane firmy rejestrującej płatności, głównie dane AML
currentData.service	dane konfiguracyjne serwisu

Wartości poszczególnych pól w zależności od scenariusza

- **Dane kontaktowe i ich weryfikacja**

Istnieje możliwość, że niektóre z danych kontaktowych będą wymagać weryfikacji wykonanej przez Klienta. Pola których może dotyczyć weryfikacja to:

```
currentData.company.phone
currentData.service.contactEmail
```

Każde z tych pól reprezentowane jest w postaci obiektu:

```
{
  "currentValue": "wartość",
  "waitingValue": null
}
```

currentValue - wartość aktualnie obowiązująca w systemie

waitingValue - wartość oczekująca na weryfikację po stronie klienta. W sytuacji gdy klient wprowadził kilka wartości nie weryfikując żadnej z nich pole waitingValue zawierać będzie ostatnią z nich. Wartość null oznacza, że żadna dana nie oczekuje na weryfikację po stronie klienta.

- **Nagłówek revisionHeader**

Może wystąpić kilka postaci tego nagłówka:

W pełni uzupełniony:

```
"revisionHeader": {
  "revisionId": "046b8dcf-3581-4079-813c-34d0a2871fae",
  "autopayVerificationStatus": "PENDING",
  "orderId": "151007_2d87806818cf319d7127ffb"
}
```

Nagłówek taki wystąpi, gdy podczas akcji zainicjalizowanej przez Integratora wystąpi konieczność opłacenia transakcji weryfikacyjnej przez klienta.

Puste OrderId:

```
"revisionHeader":{
  "revisionId":"046b8dcf-3581-4079-813c-34d0a2871fae",
  "autopayVerificationStatus":"PENDING",
  "orderId":null
}
```

Taka forma nagłówka wystąpi gdy akcja Integratora wygeneruje weryfikację niewrażliwą tzn taką, która nie wymaga opłacenia transakcji weryfikacyjnej.

Brak nagłówka revisionHeader:

```
"revisionHeader":null
```

Nagłówek revisionHeader nie wystąpi gdy zmiana w obrębie service/acceptanta zainicjalizowana zostanie przez Autopay np. w wyniku okresowej weryfikacji danych.

- **Pole revisionHeader.autopayVerificationStatus**

Pole określa status weryfikacji dla wskazanej rewizji. Możliwe wartości tego pola:

PENDING - rewizja oczekuje na weryfikację

POSITIVE - rewizja została zaakceptowana

NEGATIVE - rewizja została odrzucona

NEED_FEEDBACK - wymagana jest informacja/akcja po stronie klienta

Wartość pola **revisionHeader.autopayVerificationStatus** oznacza status weryfikacji po stronie Autopay i nie ma związku z weryfikacją danych kontaktowych wykonywanych przez klienta.

Wartość pola **revisionHeader.autopayVerificationStatus** nie dotyczy danych z węzła **currentData**, jest to status przesłanej rewizji.

- **Pole currentData.company.physicalPerson** Pole to wypełnione jest obiektem jak niżej, gdy rejestracja lub edycja dotyczą działalności nierejestrowanej tzn. **company.activityKind = NON_ACCOUNTED_ACTIVITY**

```
{
  "personName":{
    "firstName":"Jan",
    "lastName":"Kowalski"
  },
  "pesel":"69010583482",
  "documentData":{
    "number":"BKI332498",
    "type":"ID",
    "expirationDate":"2030-01-01",
    "country":"PL"
  },
}
```

```
"citizenship": "PL",
"birthData": {
  "date": "1951-06-19",
  "country": "PL"
}
}
```

- **Pole `currentData.company.address`**

Pole `address` w zależności od danych przechowywanych w Autopay może przyjąć dwie formy:

- wypełnione `address.address` oraz puste `address.street`, `address.houseNumber`, `address.flatNumber`

Przykład:

```
"address": {
  "address": "ul Testowa 77/3",
  "postalCode": "80-462",
  "city": "Gdańsk",
  "country": "PL",
  "street": null,
  "houseNumber": null,
  "flatNumber": null
}
```

- puste pole `address.address` i wypełnione `address.street`, `address.houseNumber`, `address.flatNumber`

Przykład:

```
"address": {
  "address": null,
  "postalCode": "80-462",
  "city": "Gdańsk",
  "country": "PL",
  "street": "ul Testowa",
  "houseNumber": 77,
  "flatNumber": 3
}
```

Natychmiastowe powiadomienie o zmianie statusu płatności kartowych (ICN Card)

Proces udostępnienia sklepowi płatności za pośrednictwem kart płatniczych, wymaga weryfikacji sklepu zarówno po stronie AP jak i po stronie administratora płatności kartowych, dlatego uruchomienie zazwyczaj trwa dłużej niż sama aktywacja sklepu.

System umożliwia wysyłkę natychmiastowego powiadomienia w przypadku włączenia lub wyłączenia możliwości wykonywania płatności przy użyciu kart płatniczych w sklepie, aby zapewnić spójność konfiguracji sklepu pomiędzy Platformą Integratora a AP. Zostało to zrealizowane poprzez wysyłkę komunikatu w formacie JSON przy użyciu REST API.

UWAGA: Poprawne działanie usługi wymaga zaimplementowania po stronie Integratora metody pozwalającej na przyjęcie wspomnianego komunikatu.

Opis pól komunikatu wysyłanego do Integratora

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceId	TAK	integer	Stały unikalny identyfikator sklepu nadany przez System płatności online.
2	orderId	TAK	string	Unikalny identyfikator zamówienia nadawany przez AP. Składnia orderId: <i>wartośćserviceId_unikalnylosowyciągznaków</i>
3	cardsStatus	TAK	string	Status uruchomienia płatności kartowych w sklepie. Dostępne wartości: ACTIVE INACTIVE
nd.	hash	TAK	string	Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wyłącznie wartości pól, bez nazw parametrów. Kolejność sklejania pól jest zgodna z kolejnością ich występowania na liście parametrów. Do powstałego w powyższy sposób łańcucha doklejany jest na jego końcu klucz, współdzielony między System Platformy i System płatności online. Z tak powstałego łańcucha obliczana jest wartość funkcji skrótu SHA256 i stanowi ona wartość pola Hash komunikatu. <i>Hash = SHA256(wartości_pól_komunikatu + klucz_współdzielony)</i>
nd.	currencies	TAK	string	Lista walut, dla których zmienił się status uruchomienia płatności kartowych

Przykład.

```
{
  "serviceId": "1111111",
  "orderId": "22222",
  "cardsStatus": "ACTIVE",
  "hash": "81eb70b8f2c4576bfb375a7ccbfcfb196b235556bcc329aa40a3dc8bfd",
  "currencies": ["PLN", "EUR", "GBP", "USD"]
}
```

Opis pól komunikatu zwrotnego do AP

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceId	TAK	integer	Stały unikalny identyfikator sklepu nadany przez System płatności online.
2	orderId	TAK	string	Unikalny identyfikator zamówienia nadawany przez AP.
3	confirmation	TAK	string	Status potwierdzenia: - CONFIRMED - NOTCONFIRMED
nd.	hash	TAK	string	Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wyłącznie wartości pól, bez nazw parametrów. Kolejność sklejania pól jest zgodna z kolejnością ich występowania na liście parametrów. Do powstałego w powyższy sposób łańcucha doklejany jest na jego końcu klucz, współdzielony między System Platformy i System płatności online. Z tak powstałego łańcucha obliczana jest wartość funkcji skrótu SHA256 i stanowi ona wartość pola Hash komunikatu. <i>Hash = SHA256(wartości_pól_komunikatu + klucz_współdzielony)</i>

Przykład.

```
{
  "serviceId": "1111111",
  "orderId": "22222",
  "confirmation": "CONFIRMED",
  "hash": "81eb70bcb8f2c4576bfb375a7ccbfcfb196b2355986556b29aa40a3dc8bfd"
}
```

Natychmiastowe powiadomienie o dokonanej zmianie polegającej na włączeniu oraz wyłączeniu danego kanału płatności na sklepie (ICN Gateway)

System umożliwia wysyłkę do Integratora natychmiastowego powiadomienia o włączeniu oraz wyłączeniu wskazanych kanałów płatności na danym sklepie. Zostało to zrealizowane poprzez wysyłkę komunikatu w formacie JSON. Poprawne działanie usługi wymaga zaimplementowania po stronie Integratora metody pozwalającej na przyjęcie wspomnianego komunikatu.

Opis Pól komunikatu wysyłanego do Integratora.

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceld	TAK	integer	Stały unikalny identyfikator sklepu nadany przez System płatności online.
2	orderId	TAK	string	Unikalny identyfikator zamówienia nadawany przez Autopay.
-	gatewayConfigurations	TAK		Obiekt przechowujący listę kanałów płatności, dla których nastąpiła zmiana aktywności.
3	gatewayConfiguration.gatewayId	TAK	integer	Identyfikator kanału płatności.
4	gatewayConfiguration.active	TAK	boolean	Informacja, czy wskazany kanał płatności został włączony czy wyłączony.
-	hash	TAK	string	Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wartości pól, bez nazw parametrów. Kolejność sklejania pól jest zgodna z kolejnością ich występowania na liście parametrów. Do powstałego w powyższy sposób łańcucha doklepany jest na jego końcu klucz, współdzielony między System Platformy i System płatności online. Z tak powstałego łańcucha obliczana jest wartość funkcji skrótu SHA256 i stanowi ona wartość pola Hash komunikatu.

UWAGA: Jeśli podczas integracji z Partnerem ustalono separator do liczenia hash'a dla komunikatów, zostanie on użyty. W takim przypadku liczenie hash będzie odbywać się według schematu:

Hash = SHA256(wartość_pola_1_komunikatu + separator + wartość_pola_2_komunikatu + separator + wartość_pola_3_komunikatu + separator + [kolejne pola oddzielone separatorem])

klucz_współdzielony)

Przykład:

```
{
  "serviceId": 111111,
  "gatewayConfigurations": [
    {
      "gatewayId": 101,
      "active": false
    },
    {
      "gatewayId": 100,
      "active": true
    }
  ],
  "orderId": "47781_0a25c3c9-e0a7-4bfa-9bdf-75ca75c2569d",
  "hash": "5f3dfeec2cea228e2f4db8840fdc0e4d91d224da195618c0c67c6a7ea53c1832"
}
```

Opis pól komunikatu zwrótnego do Autopay.

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceId	TAK	integer	Stały unikalny identyfikator sklepu nadany przez System płatności online.
2	orderId	TAK	string	Unikalny identyfikator zamówienia nadawany przez Autopay.
3	confirmation	TAK	string	Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wyłącznie wartości pól, bez nazw parametrów. Kolejność sklejania pól jest zgodna z kolejnością ich występowania na liście parametrów. Do powstałego w powyższy sposób łańcucha doklejany jest na jego końcu klucz, współdzielony między System Platformy i System płatności online. Z tak powstałego łańcucha obliczana jest wartość funkcji skrótu SHA256 i stanowi ona wartość pola Hash komunikatu. Hash = SHA256(wartości_pól_komunikatu + klucz_współdzielony)

Przykład:

```
Przykład.
{
  "serviceId": "1111111",
  "orderId": "22222",
  "confirmation": "CONFIRMED",
}
```

```
"hash": "81eb70bcb8f2c4576bfb375a7ccbfcfb196b2355986556b29aa40a3dc8bfd"  
}
```

Pobranie przez integratora danych Serwisu - GetAMLCompanyData

GetAMLCompanyDataReq

Komunikat służący do uzyskania aktualnych w Systemie danych Partnera. Ponieważ AP jest zobligowane, aby gwarantować aktualność danych obsługiwanych Partnerów, może nastąpić ich aktualizacja (w dowolnym momencie) na podstawie zaufanych źródeł zewnętrznych. W związku z powyższym, wymagane jest, aby przed każdym wyświetleniem na Platformie Integratora, nastąpiła ich aktualizacja na Platformie poprzez wywołanie metody GetAMLCompanyData.

```
<xsd:element name="GetAMLCompanyDataReq">  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element name="AcceptorID" type="xsd:int"/>  
      <xsd:element name="Header" type="tns:Header"/>  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>
```

Opis pól komunikatu

nazwa pola	typ	opis
AcceptorID	integer	ID akceptanta.
Header	header	Obiekt służący do przekazywania danych nagłówka dotyczących bezpieczeństwa komunikacji i poprawności przesyłanych danych.

```
<xsd:complexType name="Header">  
  <xsd:sequence>  
    <xsd:element name="PlatformId" type="xsd:string"/>  
    <xsd:element name="MessageTime" type="xsd:dateTime"/>  
    <xsd:element name="RequestId" type="xsd:long"/>  
    <xsd:element name="Hash" type="xsd:string"/>  
  </xsd:sequence>  
</xsd:complexType>
```

Opis pól komunikatu

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	PlatformId	TAK	string	Stały unikalny identyfikator Platformy nadany przez System płatności online.

kolejność Hash	nazwa	wymagany	typ	opis
2	MessageTime	TAK	dateTime	Czas wygenerowania komunikatu, komunikaty z czasem ustawionym późniejszym niż 5 minut od czasu serwera Systemu płatności online będą odrzucane. Warto ustawić czas komunikatu now()-1min, na wypadek rozszynchronizowania czasu na serwerach. Przykład: 2016-07-20T09:35:00.000 (komunikat wygenerowany w chwili 2016-07-20 09:36:00).
3	RequestId	TAK	long	Unikalny identyfikator żądania w ramach Platformy nadawany przez stronę inicjującą dany komunikat.
nd.	Hash	TAK	string{64}	Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wyłącznie wartości pól, bez nazw parametrów. Kolejność sklejania pól jest zgodna z kolejnością ich występowania na liście parametrów. Do powstałego w powyższy sposób łańcucha doklejany jest na jego końcu klucz, współdzielony między System Platformy i System płatności online. Z tak powstałego łańcucha obliczana jest wartość funkcji skrótu SHA256 i stanowi ona wartość pola Hash komunikatu. Hash = SHA256(wartości_pól_komunikatu + klucz_współdzielony) Przykład: SHA256(11112016-07-20T09:35:00.000111222333aaabbbccc) gdzie: <i>PlatformId = 1111</i> <i>MessageTime = 2016-07-20T09:35:00.000</i> <i>RequestId = 111222333</i> <i>Klucz współdzielony = aaabbbccc</i>

"GetAMLCompanyDataResp"

Komunikat jest odpowiedzią na GetAMLCompanyDataResp.

```

<xsd:element name="GetAMLCompanyDataResp">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Result" type="xsd:string"/>
      <xsd:element name="ErrorStatus" type="xsd:string"
nillable="true"/>
      <xsd:element name="Company" type="tns:Company"/>
      <xsd:element name="isServiceActive" type="xsd:boolean"

```

```

nillable="true"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

```

Opis pól komunikatu

nazwa pola	typ	opis
Result	string	OK – operacja zakończona sukcesem ERROR – błąd przy edycji Partnera
ErrorStatus	string	Status błędu.
Company	Company	Obiekt z danymi firmy Partnera.
isServiceActive	boolean	Informacja o tym, czy serwis jest aktywny.

WSKAZÓWKA: Typ komunikatu **Company**, oraz poszczególne jego składowe, opisano szczegółowo w części [Typy złożone](#).

Pobranie informacji o aktualnej liście dostępnych regulaminów - GetLegalData

System umożliwi Integratorowi pobranie aktualnej, pełnej listy regulaminów odpowiadających dostępnym dla Integratora stawkom prowizyjnym (CommissionModel) lub konkretnej listy regulaminów ograniczonej filtrami podanymi w parametrach żądania. W tym celu należy wywołać metodę getLegalData (https://domena_BMAutopay/getLegalData) z odpowiednimi parametrami. Wymiana komunikatów pomiędzy Autopay, a platformą Integratora odbywa się w formacie JSON. Wszystkie parametry przekazywane są metodą POST (Content-Type: application/json). Poniżej znajduje się lista dostępnych parametrów. Dostęp do usługi zabezpieczony jest poprzez filtrowanie adresów IP.

Opis pól komunikatu żądania

Kolejność do hash	Nazwa	Wymagany	Typ	Opis
.	header	TAK		Obiekt przechowujący pola autoryzacyjne komunikatu.
1	platformId	TAK	integer	Stały unikalny identyfikator Platformy nadany przez System płatności online.
2	messageId	TAK	string{32}	Unikalny identyfikator żądania w ramach Platformy nadawany przez stronę inicjującą dany komunikat.

Kolejność do hash	Nazwa	Wymagany	Typ	Opis
3	messageTime	TAK	dateTime	Czas wygenerowania komunikatu, komunikaty z czasem ustawionym późniejszym niż 5 minut od czasu serwera Systemu płatności online będą odrzucane. Warto ustawić czas komunikatu now()-1min, na wypadek roz synchronizowania czasu na serwerach. Przykład: 2016-07-20T09:35:00.000 (komunikat wygenerowany w chwili 2016-07-20 09:36:00).
-	hash	TAK	string	Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wyłącznie wartości pól, bez nazw parametrów. Kolejność sklejania pól jest zgodna z kolejnością ich występowania na liście parametrów. Do powstałego w powyższy sposób łańcucha doklejany jest na jego końcu klucz, współdzielony między System Platformy i System płatności online. Z tak powstałego łańcucha obliczana jest wartość funkcji skrótu SHA256 i stanowi ona wartość pola Hash komunikatu. Hash = SHA256(wartości_pól_komunikatu + klucz_współdzielony).
-	filterParams	NIE		Obiekt przechowujący parametry będące filtrami zapytania. Niewysłanie filtrów w zapytaniu komunikacie żądanie do Autopay spowoduje zwrócenie pełnej listy regulaminów dostępnej dla Integratora.
-	commissionmodellList	NIE		Lista stawek prowizyjnych, dla których mają zostać zwrócone regulaminy. Podanie konkretnej stawki spowoduje zwrócenie odpowiadającego jej regulaminu.
-	languageList	NIE		Lista języków, w których mają zostać zwrócone regulaminy. Przykład: Podanie języka PL spowoduje zwrócenie wszystkich regulaminów napisanych w języku polskim.

Przykład.

Odpytanie o listę regulaminów dla stawek prowizyjnych 1 oraz 2, w języku polskim oraz angielskim dla waluty PLN.

```
{
  "header":{
    "platformId":"111111",
    "messageId":22222,
    "messageTime":"2020-06-01 09:36:00",
    "hash":"31772235489560079037848456"
  },
  "filterParams":{
    "commissionmodelList":[
      1,
      2
    ],
    "languageList":[
      "PL",
      "EN"
    ]
  }
}
```

Opis pól komunikatu odpowiedzi

Poprawna odpowiedź - status http 200.

Kolejność do hash	Nazwa	Wymagany	Typ	Opis
1	status	TAK	string	Wartość: OK - w przypadku statusu http 200.
2	messageId	TAK string {32}	Pole zwracające wartość identyfikatora komunikatu żądania.	
-	hash	TAK	string	Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wyłącznie wartości pól, bez nazw parametrów. Kolejność sklejania pól jest zgodna z kolejnością ich występowania na liście parametrów.
-	regulationList	TAK		Obiekt przechowujący listę zwracanych regulaminów.

Kolejność do hash	Nazwa	Wymagany	Typ	Opis
-	regulationId	TAK	integer	Identyfikator regulaminu, odsyłany do Autopay w metodach RegisterCompany, UpdateCompany, UpdateService, UpdateConfiguration, będący informacją dla Autopay jaki regulamin został zaakceptowany na formularzu Integratora.
-	regulationLink	TAK	string	Link do pliku z regulaminem.
-	commissionmodel	TAK	integer	tawka prowizyjna, której dotyczy regulamin.
-	language	TAK	string	Język w jakim został napisany regulamin.
-	currency	TAK	string	Waluta stawki prowizyjnej, której dotyczy regulamin.

Odpowiedź z komunikatem błędu - status http 400.

Kolejność do hash	Nazwa	Wymagany	Typ	Opis
	errors	TAK		Obiekt przechowujący listę błędów zawartych w komunikacie żądania.
	field	TAK	string	Wskazuje na pole, którego dotyczy błąd.
	error	TAK	string	Słowny opis błędu.
	errorCode	integer	Kod błędu - pełen wykaz błędów znajduje poniżej	

Przykład.

Zwrócenie listy regulaminów

```
{
  "regulationList": [
    {
      "commissionModel": 1,
      "language": "PL",
      "regulationLink": "https://platnosci-accept.bm.pl/regulations/api/download/4eaf489b-c27f-40ac-b888-032f562077dd",
      "regulationId": 41,
      "currency": "PLN"
    },
    {
      "commissionModel": 3,
      "language": "PL",

```



```

"regulationLink": "https://platnosci-accept.bm.pl/regulations/api/download/70c5638e-7301-46f9-87e1-522dcff1db4e",
  "regulationId": 42,
  "currency": "PLN"
},
{
  "commissionModel": 6,
  "language": "PL",
  "regulationLink": "https://platnosci-accept.bm.pl/regulations/api/download/0ef22461-7245-4501-970f-e09f2929856f",
  "regulationId": 43,
  "currency": "PLN"
}
],
"messageId": "22121111112411112211111311212260",
"status": "OK",
"hash": "a2bd34888537a5c91c2700f12ec4e59786b03e9d4a92ad9ffd1dc49c8c8edad2"
}

```

Wykaz kodów błędów

Kod błędu	Opis	Parametr, którego dotyczy błąd
6000	Błędny hash	hash
6023	Paramert hash nie został podany, a jest wymagany	hash
6017	Parametr platformId nie został podany a jest wymagany	platformId
6018	Parametr messageId nie został podany a jest wymagany.	messageId
6019	Wartość parametru messageId ma niepoprawną długość.	messageId
6015	Błędny format daty	messageTime
6015	Data wysłana w odpytaniu o link do formularza jest nie poza dopuszczalnym zakresem.	messageTime
6015	Parametr messageTime nie został podany a jest wymagany.	messageTime
6027	Błędna wartość w liście.	commissionmodelList
6028	łędna wartość w liście.	languageList
6029	Nie podano obiektu header.	header

Zmiana konfiguracji Serwisu - "UpdateConfiguration"

"UpdateConfigurationReq"

Komunikat umożliwiający zmianę konfiguracji Sklepu przez Integratora bez konieczności wykonywania

przelewu weryfikacyjnego.

```

<xsd:element name="UpdateConfigurationReq">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="ServiceID" type="xsd:int"
minOccurs="0"/>
      <xsd:element name="Header" type="tns:Header"/>
      <xsd:element name="ConfigurationData"
type="tns:ConfigurationData"/>
      <xsd:element name="Currency" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

Opis pól komunikatu

nazwa	wymagany	typ	opis
ServiceID	NIE	integer	ID serwisu.
Header	NIE	Header	Obiekt służący do przekazywania danych nagłówka dotyczących bezpieczeństwa komunikacji i poprawności przesyłanych danych.
ConfigurationData	NIE	ConfigurationData	Obiekt służący do przekazania informacji o nowej konfiguracji sklepu.
Currency	NIE	string	Waluta.

```

<xsd:complexType name="Header">
  <xsd:sequence>
    <xsd:element name="PlatformId" type="xsd:string"/>
    <xsd:element name="MessageTime" type="xsd:dateTime"/>
    <xsd:element name="RequestId" type="xsd:long"/>
    <xsd:element name="Hash" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

```

Opis pól komunikatu

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	PlatformId	TAK	string	Stały unikalny identyfikator Platformy nadany przez System płatności online.

kolejność Hash	nazwa	wymagany	typ	opis
3	MessageTime	TAK	dateTime	Czas wygenerowania komunikatu, komunikaty z czasem ustawionym późniejszym niż 5 minut od czasu serwera Systemu płatności online będą odrzucane. Warto ustawić czas komunikatu now()-1min, na wypadek rozsynchronizowania czasu na serwerach. Przykład: 2016-07-20T09:35:00.000 (komunikat wygenerowany w chwili 2016-07-20 09:36:00).
4	RequestId	TAK	long	Unikalny identyfikator żądania w ramach Platformy nadawany przez stronę inicjującą dany komunikat.
nd.	Hash	TAK	string{64}	Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wyłącznie wartości pól, bez nazw parametrów. Kolejność sklejania pól jest zgodna z kolejnością ich występowania na liście parametrów. Do powstałego w powyższy sposób łańcucha doklejany jest na jego końcu klucz, współdzielony między System Platformy i System płatności online. Z tak powstałego łańcucha obliczana jest wartość funkcji skrótu SHA256 i stanowi ona wartość pola Hash komunikatu. <i>Hash = SHA256(wartości_pól_komunikatu + klucz_współdzielony)</i>

```

<xsd:complexType name="ConfigurationData">
  <xsd:sequence>
    <xsd:element name="isTransactionRefundAllowed"
type="tns:BooleanType" minOccurs="0"/>
    <xsd:element name="CommissionModel" type="xsd:long"
nillable="true"/>
    <xsd:element name="isCardsPaymentRequired"
type="tns:BooleanType" minOccurs="0"/>
    <xsd:element name="ServiceUrl" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

Opis pól komunikatu

nazwa pola	wymagany	typ	opis
isTransactionRefundAllowed	NIE	BooleanType	Informacja czy Partner chce udostępnić dla Serwisu opcję wypłat z Rachunku Płatniczego oraz zwrotów transakcji. Niewysłanie elementu bądź wysłanie wartości FALSE powoduje zablokowanie tej opcji. Wysłanie TRUE powoduje jej udostępnienie Serwisowi.
CommissionModel	NIE	long	Numer uzgodnionego w trakcie integracji modelu prowizyjnego.
isCardsPaymentRequired	NIE	BooleanType	Informacja czy Partner chce udostępnić dla Serwisu opcję płatności kartami. Niewysłanie elementu bądź wysłanie wartości false, powoduje wyłączenie kart. Wysłanie TRUE, powoduje rozpoczęcie procesu aktywacji kart, lub utrzymanie dostępności kart (w przypadku pozytywnie zakończonogo procesu aktywacji kart dla Serwisu).
ServiceUrl	NIE	string	Pole pozwalające na zmianę adresu tymczasowego podanego podczas rejestracji na adres docelowy.

UWAGA: Podany w **ServiceUrl** adres musi zawierać stałą część domeny.

Na przykład zmiana adresu: <https://nazwasklepu.integrator.pl> na <https://nazwasklepu.pl>

"UpdateConfigurationResp"

Komunikat jest odpowiedzią na UpdateConfigurationReq.

```
<xsd:element name="UpdateConfigurationResp">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Result" type="xsd:string"/>
      <xsd:element name="ErrorStatus" type="xsd:string"
nillable="true"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Opis pól komunikatu

nazwa pola	typ	opis
Result	xsd:string	OK - operacja zakończona sukcesem ERROR - błąd przy edycji serwisu
ErrorStatus	xsd:string	Status błędu, jeśli wystąpił ERROR.

nazwa pola	typ	opis
ActivationLink	xsd:string	Adres URL, prowadzący do Systemu skonfigurowanej pod proces weryfikacji poprzez przelew.

Anulowanie aktualnej edycji serwisu, która jest w trakcie weryfikacji - "CancelUpdate"

Komunikat służący do anulowania edycji danych serwisu, który jest aktualnie w trakcie weryfikacji. Metoda jest wykorzystywane w momencie, konfiguracja Integratora po stronie Systemu Autopay blokuje obsługę wielu weryfikacji na raz aby zachować spójność zmienianych danych. Poprawne wykonanie **CancelUpdate** umożliwi wysłanie danych z formularza celem edycji danych sklepu, bez oczekiwania na finalny status weryfikacji poprzedniej edycji danych.

UWAGA: Próby pobrania linka do formularza dla serwisu, który jest aktualnie w trakcie weryfikacji zakończą się błędem: SHOP_IN_VERIFICATION_STATUS.

CancelUpdateReq

```
<xsd:element name="CancelUpdateReq">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="ServiceID" type="xsd:int"
minOccurs="0"/>
      <xsd:element name="AcceptorID" type="xsd:int"
minOccurs="0"/>
      <xsd:element name="Header" type="tns:Header"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Opis pól

nazwa pola	typ	opis
ServiceID	int	ID serwisu.
AcceptorID	int	ID akceptanta.
Header	header	Obiekt służący do przekazywania danych nagłówka dotyczących bezpieczeństwa komunikacji i poprawności przesyłanych danych.

CancelUpdateResp

```
<xsd:element name="CancelUpdateResp">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Result" type="xsd:string"/>
      <xsd:element name="ErrorStatus" type="xsd:string"
nillable="true"/>
    </xsd:sequence>
  </xsd:complexType>
```

```
</xsd:element>
```

Opis pól

nazwa pola	typ	opis
Result	xsd:string	OK- operacja zakończona sukcesem ERROR - błąd przy edycji serwisu.
ErrorStatus	xsd:string	Gdy wystąpił ERROR.

Typy złożone

Header

Typ komunikatu służący do przekazywania danych nagłówka dotyczących bezpieczeństwa komunikacji i poprawności przesyłanych danych.

```
<xsd:complexType name="Header">  
  <xsd:sequence>  
    <xsd:element name="PlatformId" type="xsd:string"/>  
    <xsd:element name="MessageTime" type="xsd:date"/>  
    <xsd:element name="RequestId" type="xsd:long"/>  
    <xsd:element name="Hash" type="xsd:string"/>  
  </xsd:sequence>  
</xsd:complexType>
```

Opis pól komunikatu

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	PlatformId	TAK	string	Stały unikalny identyfikator Platformy nadany przez System płatności online.
3	MessageTime	TAK	dateTime	Czas wygenerowania komunikatu, komunikaty z czasem ustawionym późniejszym niż 5 minut od czasu serwera Systemu płatności online będą odrzucone. Warto ustawić czas komunikatu now()-1min, na wypadek rozsynchronizowania czasu na serwerach. Przykład: 2016-07-20T09:35:00.000 (komunikat wygenerowany w chwili 2016-07-20 09:36:00).
4	RequestId	TAK	long	Unikalny identyfikator żądania w ramach Platformy nadawany przez stronę inicjującą dany komunikat.

kolejność Hash	nazwa	wymagany	typ	opis
nd.	Hash	TAK	string{64}	Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wyłącznie wartości pól, bez nazw parametrów. Kolejność sklejania pól jest zgodna z kolejnością ich występowania na liście parametrów. Do powstałego w powyższy sposób łańcucha doklejany jest na jego końcu klucz, współdzielony między System Platformy i System płatności online. Z tak powstałego łańcucha obliczana jest wartość funkcji skrótu SHA256 i stanowi ona wartość pola Hash komunikatu. Hash = SHA256(wartości_pól_komunikatu + klucz_współdzielony)

Address

Ten typ komunikatu służy do przekazywania adresu danego podmiotu.

```
<xsd:complexType name="Address">
  <xsd:sequence>
    <xsd:element name="Address" type="xsd:string"/>
    <xsd:element name="PostalCode" type="xsd:string"/>
    <xsd:element name="City" type="xsd:string"/>
    <xsd:element name="Country" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```

Opis pól komunikatu

nazwa pola	wymagany	typ	opis
Address	TAK	string	Adres.
PostalCode	TAK	string	Kod pocztowy.
City	TAK	string	Miasto.
Country	TAK	string	Kraj.

NIPTyp

Typ opisujący numer NIP z ograniczeniem od 2 do 15 znaków.

```
<xsd:simpleType name="NIPTyp">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="2"/>
    <xsd:maxLength value="15"/>
  </xsd:restriction>
</xsd:simpleType>
```

```
</xsd:simpleType>
```

Currency

Typ opisujący walutę.

```
<xsd:simpleType name="Currency">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="PLN"/>  
    <xsd:enumeration value="EUR"/>  
    <xsd:enumeration value="USD"/>  
    <xsd:enumeration value="GBP"/>  
    <xsd:enumeration value="CZK"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

ForeignTransferModeType

System przelewu rozliczeniowego w przypadku rachunku zagranicznego.

```
<xsd:simpleType name="ForeignTransferModeType">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="SWIFT"/>  
    <xsd:enumeration value="SEPA"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

Beneficials

Obiekt jest listą typów beneficjentów rzeczywistych Partnera.

```
<xsd:complexType name="Beneficials">  
  <xsd:sequence>  
    <xsd:element name="Beneficial" type="tns:Beneficial"  
minOccurs="0" maxOccurs="2"/>  
  </xsd:sequence>  
</xsd:complexType>
```

Beneficial

Obiekt zawiera typ beneficjenta rzeczywistego oraz ewentualną listę beneficjentów tego typu.

```
<xsd:complexType name="Beneficial">  
  <xsd:sequence>  
    <xsd:element name="BeneficialType" type="xsd:int"/>  
    <xsd:element name="BeneficialOwners" type="tns:BeneficialOwners"  
minOccurs="0" />  
  </xsd:sequence>  
</xsd:complexType>
```


Opis pól komunikatu

nazwa pola	wymagany	typ	opis
BeneficialType	TAK	int	Typ beneficjenta: 11 - BENEFICJENT UDZIAŁOWIEC- osoba fizyczna, która ma pośrednio lub bezpośrednio więcej niż 25% udziałów/głosów 12 - BENEFICJENT KONTROLA- osoba fizyczna, sprawująca kontrolę poprzez jednostkę dominującą według przepisów o rachunkowości 13 - BENEFICJENT KIEROWNICTWO- osoby zajmujące wyższe stanowisko kierownicze w Spółce (zarząd lub rada nadzorcza) 14 - BENEFICJENT WSPÓLNIK-wspólnik spółki 15 - BENEFICJENT INNY- osoba nie będąca beneficjentem rzeczywistym; istnieje osoba fizyczna wywierająca na nią wpływ lub sprawująca nad nią kontrolę 16 - BENEFICJENT WŁAŚCICIEL
BeneficialOwners	NIE	BeneficialOwners	Lista beneficjentów danego typu.

BeneficialOwners

Typ opisuje listę beneficjentów rzeczywistych.

```
<xsd:complexType name="BeneficialOwners">
  <xsd:sequence>
    <xsd:element name="BeneficialOwner" type="tns:BeneficialOwner"
minOccurs="0" maxOccurs="8"/>
  </xsd:sequence>
</xsd:complexType>
```

BeneficialOwner

Typ opisuje dane pojedynczego beneficjenta rzeczywistego.

```
<xsd:complexType name="BeneficialOwner">
  <xsd:sequence>
    <xsd:element name="FirstName" type="xsd:string"
nillable="false"/>
    <xsd:element name="LastName" type="xsd:string"
nillable="false"/>
    <xsd:element name="Citizenship" type="xsd:string"/>
    <xsd:element name="Share" type="xsd:int" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

Opis pól komunikatu

nazwa pola	wymagany	typ	opis
FirstName	TAK	string	Imię beneficjenta.
LastName	TAK	string	Nazwisko beneficjenta.
Citizenship	TAK	string	Obywatelstwo beneficjenta.
Share	NIE	int	Ilość udziałów/akcji beneficjenta wyrażona w procentach. Udział musi być większy lub równy 25[%]. Suma udziałów wszystkich beneficjentów Partnera nie powinna przekraczać 100[%]. Udziały określone są tylko dla beneficjentów typu 11.

Plenipotentiary

Typ komunikatu opisujący pełnomocnika.

```
<xsd:complexType name="Plenipotentiary">
  <xsd:sequence>
    <xsd:element name="FirstName" type="xsd:string"/>
    <xsd:element name="LastName" type="xsd:string"/>
    <xsd:element name="Citizenship" type="xsd:string"/>
    <xsd:element name="Pesel" type="xsd:string" minOccurs="0"/>
    <xsd:element name="BirthDate" type="xsd:string" minOccurs="0"/>
    <xsd:element name="BirthCountry" type="xsd:string"
minOccurs="0"/>
    <xsd:element name="DocumentType" type="xsd:string"
minOccurs="0"/>
    <xsd:element name="Document" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DocumentExpirationDate" type="xsd:string"
minOccurs="0"/>
    <xsd:element name="DocumentCountryId" type="xsd:string"
minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

Opis pól komunikatu

nazwa pola	wymagany	typ	opis
FirstName	TAK	string	Imię pełnomocnika.
LastName	TAK	string	Nazwisko pełnomocnika.
Citizenship	TAK	string	Obywatelstwo pełnomocnika.
Pesel	NIE	string	Pesel pełnomocnika. W przypadku braku peselu należy podać datę oraz państwo urodzenia.
BirthDate	NIE	string	Data urodzenia pełnomocnika w formacie YYYY-mm-dd (np. 1990-01-01). Podawana, kiedy nie jest znany pesel lub pełnomocnik nie posiada numeru pesel.
BirthCountry	NIE	string	Państwo urodzenia pełnomocnika.

nazwa pola	wymagany	typ	opis
DocumentType	TAK	string	Typ dokumentu. Akceptowane wartości: PASSPORT lub ID.
Document	TAK	string	Numer dokumentu.
DocumentExpirationDate	TAK	string	Data ważności dokumentu.
DocumentCountryId	TAK	string	ID państwa dokumentu beneficjenta. Wymagane w przypadku podania DocumentType=PASSPORT. WSKAZÓWKA: Akceptowalne wartości zostały podane w części Zmiana konfiguracji Serwisu - "UPDATECONFIGURATION"

Partners

Typ opisuje listę współników.

```
<xsd:complexType name="Partners">
  <xsd:sequence>
    <xsd:element name="Partner" type="tns:Partner" minOccurs="0"
maxOccurs="7" />
  </xsd:sequence>
</xsd:complexType>
```

Partner

Typ komunikatu opisujący współnika.

```
<xsd:complexType name="Partners">
  <xsd:sequence>
    <xsd:element name="FirstName" type="xsd:string"/>
    <xsd:element name="LastName" type="xsd:string"/>
    <xsd:element name="Citizenship" type="xsd:string"/>
    <xsd:element name="Pesel" type="xsd:string" minOccurs="0"/>
    <xsd:element name="BirthDate" type="xsd:string" minOccurs="0"/>
    <xsd:element name="BirthCountry" type="xsd:string"
minOccurs="0" />
    <xsd:element name="DocumentType" type="xsd:string"
minOccurs="0" />
    <xsd:element name="Document" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DocumentExpirationDate" type="xsd:string"
minOccurs="0" />
    <xsd:element name="DocumentCountryId" type="xsd:string"
minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
```

Opis pól komunikatu

nazwa pola	wymagany	typ	opis
FirstName	TAK	string	Imię współnika.

nazwa pola	wymagany	typ	opis
LastName	TAK	string	Nazwisko wspólnika.
Citizenship	TAK	string	Obywatelstwo wspólnika.
Pesel	NIE	string	Pesel wspólnika. W przypadku braku peselu należy podać datę oraz państwo urodzenia.
BirthDate	NIE	string	Data urodzenia wspólnika w formacie YYYY-mm-dd (np. 1990-01-01). Podawana, kiedy nie jest znany pesel lub partner nie posiada numeru pesel.
DocumentType	TAK	string	Typ dokumentu. Akceptowane wartości: PASSPORT lub ID.
Document	TAK	string	Numer dokumentu. Wymagany dla activitykind: CIVIL_PARTNERSHIP. Dla innych form prawnych pole nie powinno być wypełniane.
DocumentExpirationDate	TAK	string	Data ważności dokumentu.
DocumentCountryId	TAK	string	ID państwa dokumentu beneficjenta. Wymagane w przypadku podania DocumentType=PASSPORT. WSKAZÓWKA: Akceptowalne wartości zostały podane w części Zmiana konfiguracji Serwisu - "UPDATECONFIGURATION"

Service

Typ komunikatu opisujący Serwis Partnera.

```

<xsd:complexType name="Service">
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:string" nillable="false"/>
    <xsd:element name="ServiceUrl" type="xsd:string"
nillable="false"/>
    <xsd:element name="UrlITN" type="xsd:string" nillable="false"/>
    <xsd:element name="ReturnUrl" type="xsd:string"
nillable="false"/>
    <xsd:element name="SettlementNRB" type="xsd:string" />
    <xsd:element name="SwiftCode" type="xsd:string" minOccurs="0"/>
    <xsd:element name="ForeignTransferMode"
type="tns:ForeignTransferModeType" minOccurs="0"/>
    <xsd:element name="CommissionModel" type="xsd:long"
nillable="true"/>
    <xsd:element name="isCardsPaymentRequired" type="xsd:boolean"
minOccurs="0" />
    <xsd:element name="AverageServiceTurnover" type="xsd:decimal"
minOccurs="0" />
    <xsd:element name="AverageTransactionAmount" type="xsd:decimal"
minOccurs="0" />
    <xsd:element name="EconomicPurpose" type="xsd:string"
minOccurs="0" />
    <xsd:element name="EconomicPurposeDescription" type="xsd:string"
minOccurs="0" />
    <xsd:element name="NumericTrade" type="xsd:int" minOccurs="0" />
    <xsd:element name="InvoiceEmail" type="xsd:string" minOccurs="0"
/>
  />

```

```

        <xsd:element name="ContactEmail" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="ComplaintEmail" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="ReportEmail" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="isTransactionRefundAllowed"
type="xsd:boolean" minOccurs="0" />
        <xsd:element name="Currency" type="tns:Currency" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

```

Opis pól komunikatu

nazwa pola	wymagany	typ	opis
Name	TAK	string	Nazwa Serwisu Partnera.
ServiceUrl	TAK	string	Adres URL Serwisu Partnera.
UrlITN	TAK	string	Adres URL, na który jest wysyłany ITN z powiadomieniem o zmianie statusu transakcji po otrzymaniu takiej informacji z Kanału Płatności.
ReturnUrl	TAK	string	Adres URL strony powrotu do płatności.
SettlementNRB	TAK	string	Numer konta bankowego do przekazania środków i weryfikacji poprzez Przelew weryfikacyjny.
SwiftCode	NIE	string	Kod SWIFT. Wymagany w przypadku podania numeru rachunku innego niż polski.
ForeignTransferMode	NIE	ForeignTransferModeType	System jakim wykonywane będą przelewy rozliczeniowe. Wymagany w przypadku podania numeru rachunku innego niż polski.
CommissionModel	NIE	long	Numer uzgodnionego w trakcie integracji modelu prowizyjnego.
isCardsPaymentRequired	NIE	boolean	Informacja czy Partner chce udostępnić dla Serwisu opcję płatności kartami. Niewystąpienie elementu bądź wystąpienie wartości FALSE, powoduje wyłączenie kart. Wystąpienie TRUE powoduje rozpoczęcie procesu aktywacji kart lub utrzymanie dostępności kart (w przypadku pozytywnie zakończonego procesu aktywacji kart dla Serwisu).
AverageServiceTurnover	NIE	decimal(2)	Średni obrót Serwisu. Jako separator dziesiętny używana jest kropka - '.' Format: 0.00; maksymalna długość: 14 cyfr przed kropką i 2 po kropce.
AverageTransactionAmount	NIE	decimal(2)	Średnia wartość transakcji. Jako separator dziesiętny używana jest kropka - '.' Format: 0.00; maksymalna długość: 14 cyfr przed kropką i 2 po kropce.
EconomicPurpose	TAK	string	Partner oświadcza, iż zawiera Umowę w następującym celu gospodarczym: DEVELOPMENT_ACTIVITY: rozwój działalności gospodarczej Partnera poprzez przyjmowanie od Klientów zapłaty za pośrednictwem Instrumentów Płatniczych (PBL, Karta, Szybki Przelew), o których mowa w Umowie, START_ACTIVITY: rozpoczęcie działalności gospodarczej przez Partnera w ramach realizowania sprzedaży Produktów na odległość, OTHER: opis szczegółowy w polu EconomicPurposeDescription
EconomicPurposeDescription	NIE	string	Opis celu gospodarczego wypełniany dla pola EconomicPurpose o wartości OTHER. Dopuszczalne jedynie wielkie litery alfabetu łacińskiego oraz znaki z zakresu: ĘǪÓĄąŚśŁłŻżŻćĆń

nazwa pola	wymagany	typ	opis
NumericTrade	TAK	int	<p>Branża, w której specjalizuje się Serwis Partnera. Pole jednokrotnego wyboru. Jeśli Serwis Partnera podlega pod kilka kategorii wybiera główną, która generuje największy obrót.</p> <p>Słownik wartości:</p> <ol style="list-style-type: none"> 1 Alkohol 2 Antyki, dzieła sztuki 3 Apteki i suplementy diety 4 Artykuły medyczne 5 Artykuły parafarmaceutyczne, suplementy, zioła 6 Artykuły spożywcze 7 Aukcje 8 Bielizna 9 Bilety 10 Biżuteria i zegarki 11 Militaria 12 Budowlane 13 Charytatywna 14 Chemia gospodarcza i przemysłowa 15 Dewocjonalia 16 Dom i ogród 17 Dziecko 18 E-book (książki elektroniczne) 19 Edukacja 20 Elektronika 21 E-papierosy 22 Filatelistyka 23 Finanse 24 Fundacja 25 Gadżety 26 Galanteria 27 Gastronomia 28 Gry online (nie hazard), loga, dzwonki do tel. komórkowych 29 Instytucje 30 Karty/kody pre-paid/telekarty 31 Kolekcjonerstwo 32 Komputery i sprzęt komputerowy, drukarki 33 Konta WWW i pocztowe 34 Kosmetyki i perfumy 35 Książki, gazety, czasopisma 36 Kwiaty i prezenty 37 Materiały biurowe 38 Motoryzacja 39 Multimedia i muzyka 40 Masowy Wystawca Faktur 41 Numizmatyka 42 Odzież 43 Ogłoszenia 44 Oprogramowanie, gry komputerowe i aplikacje 45 Prasa/Prenumerata 46 Rachunki 47 Rękodzieło 48 Sektor publiczny 49 Sprzęt AGD/RTV 50 Sprzęt fotograficzny 51 Sprzęt medyczny 52 Sprzęt sportowy 53 Szkolenia 54 Turystyka i hotelarstwo 55 Ubezpieczenia 56 Usługi 57 Usługi foto (odbitki) i poligraficzne 58 Usługi medyczne 59 VOD 60 Wędkarstwo 61 Wielobranżowość 62 Wyposażenie wnętrz, meble 63 Wypożyczalnia samochodów 64 Wyroby tytoniowe 65 Zabawki 66 Zoologia

nazwa pola	wymagany	typ	opis
InvoiceEmail	TAK	string	Adres email do wysyłki faktur oraz raportów miesięcznych.
ContactEmail	TAK	string	Adres email Partnera.
ComplaintEmail	TAK	string	Adres email do reklamacji.
ReportEmail	TAK	string	Adres email do wysyłki raportów dziennych.
isTransactionRefundAllowed	TAK	boolean	Informacja czy Partner chce udostępnić dla Serwisu opcję wypłat z Rachunku Płatniczego oraz zwrotów transakcji. Niewysłanie elementu bądź wysłanie wartości FALSE powoduje zablokowanie tej opcji. Wysłanie TRUE powoduje jej udostępnienie Serwisowi.
Currency	TAK	currency	Waluta serwisu.

Company

Typ komunikatu służący do przekazywania danych Partnera.

```

<xsd:complexType name="Company">
  <xsd:sequence>
    <xsd:element name="CompanyRemoteId" type="xsd:string"/>
    <xsd:element name="Name" type="xsd:string"/>
    <xsd:element name="Address" type="tns:Address"/>
    <xsd:element name="Nip" type="tns:NIPTYPE"/>
    <xsd:element name="Krs" type="xsd:string"/>
    <xsd:element name="Phone" type="xsd:string"/>
    <xsd:element name="RepresentingPersonFirstName"
type="xsd:string"/>
    <xsd:element name="RepresentingPersonLastName"
type="xsd:string"/>
    <xsd:element name="RepresentingPersonPesel" type="xsd:string"/>
    <xsd:element name="RepresentingPersonDateOfBirth"
type="xsd:string"/>
    <xsd:element name="RepresentingPersonBirthCountry"
type="xsd:string" minOccurs="0"/>
    <xsd:element name="RepresentingPersonCitizenship"
type="xsd:string"/>
    <xsd:element name="RepresentingPersonPersonalDocumentType"
type="xsd:string"/>
    <xsd:element name="RepresentingPersonPersonalDocument"
type="xsd:string"/>
    <xsd:element
name="RepresentingPersonPersonalDocumentExpirationDate" type="xsd:string"/>
    <xsd:element name="RepresentingPersonPersonalDocumentCountryId"
type="xsd:string" minOccurs="0"/>
    <xsd:element name="Service" type="tns:Service"/>
    <xsd:element name="ActivityKind" type="xsd:string"/>
    <xsd:element name="LegalForm" type="xsd:string" minOccurs="0"/>
    <xsd:element name="PanelAdministrator" type="xsd:string"
minOccurs="0" />
    <xsd:element name="isListedOnTheStockExchange"
type="tns:BooleanType" />
    <xsd:element name="Beneficials" type="tns:Beneficials"
minOccurs="0" />
    <xsd:element name="TradeRegisterName" type="xsd:string"
minOccurs="0"/>
    <xsd:element name="RegistrationDate" type="xsd:string"
minOccurs="0"/>
    <xsd:element name="Plenipotentiary" type="tns:Plenipotentiary"
minOccurs="0"/>

```

```

        <xsd:element name="Partners" type="tns:Partners" minOccurs="0"
/>
        <xsd:element name="PhysicalPerson" type="tns:PhysicalPerson"
minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

```

Opis pól komunikatu

nazwa pola	wymagany	typ	opis
CompanyRemoteld	TAK	string	ID reprezentujące Partnera.
Name	TAK	string	Pełna nazwa handlowa Partnera. Nazwa Akceptanta. Niedozwolone dla ActivityKind=NON_ACCOUNTED_ACTIVITY.
Address	TAK	address	Adres, pod którym jest zarejestrowana działalność handlowa Partnera (możliwość rejestracji wyłącznie z krajów należących do EOG). Niedozwolone dla ActivityKind=NON_ACCOUNTED_ACTIVITY.
Nip	TAK	NIPTyp	NIP Partnera. Niedozwolone dla ActivityKind=NON_ACCOUNTED_ACTIVITY.
Krs	TAK	string	KRS Partnera. Wymagany dla ActivityKind=GENERAL_PARTNERSHIP, LIMITED LIABILITY PARTNERSHIP, LIMITED PARTNERSHIP, LIMITED JOINT STOCK PARTNERSHIP, SP_ZOO, SA, SOCIETY, FOUNDATION, COOPERATIVE, CHURCH.
Phone	TAK	string	Telefon kontaktowy Partnera.
RepresentingPersonFirstName	TAK	string	Imię osoby reprezentującej Partnera. Niedozwolone dla ActivityKind=NON_ACCOUNTED_ACTIVITY.
RepresentingPersonLastName	TAK	string	Nazwisko osoby reprezentującej Partnera. Niedozwolone dla ActivityKind=NON_ACCOUNTED_ACTIVITY.
RepresentingPersonPesel	NIE	string	PESEL osoby reprezentującej Partnera. Niedozwolone dla ActivityKind=NON_ACCOUNTED_ACTIVITY.
RepresentingPersonDateOfBirth	NIE	string	Data urodzenia osoby reprezentującej Partnera w formacie YYYY-mm-dd (np. 1990-01-01). Niedozwolone dla ActivityKind=NON_ACCOUNTED_ACTIVITY. Wymagany, kiedy reprezentant nie posiada numeru PESEL.
RepresentingPersonBirthCountry	NIE	string	Państwo urodzenia. Niedozwolone dla ActivityKind=NON_ACCOUNTED_ACTIVITY. Wymagany, kiedy reprezentant nie posiada numeru PESEL.
RepresentingPersonCitizenship	TAK	string	Obywatelstwo. Niedozwolone dla ActivityKind=NON_ACCOUNTED_ACTIVITY.
RepresentingPersonPersonalDocumentType	TAK	string	Typ dokumentu potwierdzającego tożsamość osoby reprezentującej Partnera. Dopuszczalne wartości: PASSPORT – paszport ID – dowód osobisty Niedozwolone dla ActivityKind=NON_ACCOUNTED_ACTIVITY, GENERAL_PARTNERSHIP, LIMITED LIABILITY PARTNERSHIP, LIMITED PARTNERSHIP, LIMITED JOINT STOCK PARTNERSHIP, SP_ZOO, SA, SOCIETY, FOUNDATION, COOPERATIVE, GOVERNMENT, SELF_GOVERNMENTAL_UNIT, SELF_GOVERNMENTAL_CULTURAL_INSTITUTION, STATE_OWNED_ENTERPRISE, STATE_CULTURAL_INSTITUTION, PUBLIC_SECTOR_INSTITUTION, RESEARCH_INSTITUTION, CHURCH, FOREIGN

nazwa pola	wymagany	typ	opis
RepresentingPersonPersonalDocument	TAK	string	Seria i numer dowodu osobistego (lub nr paszportu - dokumentu potwierdzającego tożsamość osoby nie posiadającej dowodu osobistego). Niedozwolone dla ActivityKind=NON_ACCOUNTED_ACTIVITY, GENERAL_PARTNERSHIP, LIMITED_LIABILITY_PARTNERSHIP, LIMITED_PARTNERSHIP, LIMITED_JOINT_STOCK_PARTNERSHIP, SP_ZOO, SA, SOCIETY, FOUNDATION, COOPERATIVE, GOVERNMENT, SELF_GOVERNMENTAL_UNIT, SELF_GOVERNMENTAL_CULTURAL_INSTITUTION, STATE_OWNED_ENTERPRISE, STATE_CULTURAL_INSTITUTION, PUBLIC_SECTOR_INSTITUTION, RESEARCH_INSTITUTION, CHURCH, FOREIGN. Dopuszczalne jedynie cyfry oraz wielkie litery alfabetu łacińskiego.
RepresentingPersonPersonalDocumentExpirationDate	TAK	string	Data ważności dokumentu tożsamości (dowodu osobistego lub paszportu). Niedozwolone dla ActivityKind=NON_ACCOUNTED_ACTIVITY, GENERAL_PARTNERSHIP, LIMITED_LIABILITY_PARTNERSHIP, LIMITED_PARTNERSHIP, LIMITED_JOINT_STOCK_PARTNERSHIP, SP_ZOO, SA, SOCIETY, FOUNDATION, COOPERATIVE, GOVERNMENT, SELF_GOVERNMENTAL_UNIT, SELF_GOVERNMENTAL_CULTURAL_INSTITUTION, STATE_OWNED_ENTERPRISE, STATE_CULTURAL_INSTITUTION, PUBLIC_SECTOR_INSTITUTION, RESEARCH_INSTITUTION, CHURCH, FOREIGN. Dokument musi być ważny przynajmniej jeden dzień. Wymagany dla form prawnych: CIVIL_PARTNERSHIP, PROPRIETORSHIP.
RepresentingPersonPersonalDocumentCountryId	TAK	string	ID kraju dokumentu. Niedozwolone dla ActivityKind=NON_ACCOUNTED_ACTIVITY, GENERAL_PARTNERSHIP, LIMITED_LIABILITY_PARTNERSHIP, LIMITED_PARTNERSHIP, LIMITED_JOINT_STOCK_PARTNERSHIP, SP_ZOO, SA, SOCIETY, FOUNDATION, COOPERATIVE, GOVERNMENT, SELF_GOVERNMENTAL_UNIT, SELF_GOVERNMENTAL_CULTURAL_INSTITUTION, STATE_OWNED_ENTERPRISE, STATE_CULTURAL_INSTITUTION, PUBLIC_SECTOR_INSTITUTION, RESEARCH_INSTITUTION, CHURCH, FOREIGN. Wymagany dla form prawnych: CIVIL_PARTNERSHIP, PROPRIETORSHIP.
Service	TAK	Service	Serwis Partnera.
LegalForm	TAK	string	Nazwa zagranicznej formy prawnej. Wymagany dla ActivityKind = FOREIGN.

nazwa pola	wymagany	typ	opis
ActivityKind	TAK	string	Forma prawna. Poniżej prezentujemy dopuszczalne wartości. Integrator powinien prezentować Partnerowi formę prawną zgodną z wersją językową platformy: NON_ACCOUNTED_ACTIVITY - Działalność nieewidencjonowana osoby fizycznej PROPRIETORSHIP - Jednoosobowa działalność gospodarcza CIVIL_PARTNERSHIP - Spółka cywilna LIMITED_LIABILITY_PARTNERSHIP - Spółka partnerska GENERAL_PARTNERSHIP - Spółka jawna SP_ZOO - Spółka z ograniczoną odpowiedzialnością SA - Spółka akcyjna LIMITED_PARTNERSHIP - Spółka komandytowa LIMITED_JOINT_STOCK_PARTNERSHIP - Spółka komandytowo-akcyjna SOCIETY - Stowarzyszenie FOUNDATION - Fundacja COOPERATIVE - Spółdzielnia GOVERNMENT - Organ administracji rządowej, organ samorządu terytorialnego lub organ egzekucyjny (gminy, urzędy) SELF_GOVERNMENTAL_UNIT - Jednostka samorządu terytorialnego SELF_GOVERNMENTAL_CULTURAL_INSTITUTION - Samorządowa instytucja kultury STATE_OWNED_ENTERPRISE - Przedsiębiorstwo państwowe STATE_CULTURAL_INSTITUTION - Państwowa instytucja kultury PUBLIC_SECTOR_INSTITUTION - Instytucja gospodarki budżetowej RESEARCH_INSTITUTION - Instytut badawczy CHURCH - Kościół FOREIGN - Przedsiębiorca zagraniczny.
PanelAdministrator	NIE	string	Imię i nazwisko administratora panelu Systemu płatności online.
isListedOnTheStockExchange	TAK	boolean	Informacja czy Partner prowadzi spółkę, której papiery wartościowe są notowane na giełdzie w co najmniej jednym państwie członkowskim Unii Europejskiej lub w państwie równoważnym. Wymagane dla ActivityKind = FOREIGN.
Beneficials	TAK	Beneficials	Beneficjenci. Niedozwolone dla ActivityKind=GOVERNMENT, SELF_GOVERNMENTAL_UNIT, SELF_GOVERNMENTAL_CULTURAL_INSTITUTION, STATE_OWNED_ENTERPRISE, STATE_CULTURAL_INSTITUTION, PUBLIC_SECTOR_INSTITUTION, RESEARCH_INSTITUTION, CIVIL_PARTNERSHIP
TradeRegisterName	NIE	string	Nazwa rejestru handlowego. Wymagane w przypadku firmy zagranicznej.
RegistrationDate	NIE	string	Data rejestracji firmy.
Plenipotentiary	NIE	Plenipotentiary	Pełnomocnik.
Partners	NIE	-	Lista wspólników. Wymagana dla activitykind= CIVIL_PARTNERSHIP, GENERAL_PARTNERSHIP, LIMITED_LIABILITY_PARTNERSHIP, LIMITED_PARTNERSHIP, LIMITED_JOINT_STOCK_PARTNERSHIP
PhysicalPerson	NIE	PhysicalPerson	Osoba fizyczna. Wymagane dla ActivityKind=NON_ACCOUNTED_ACTIVITY

PhysicalPerson

Typ komunikatu służący do przekazywania danych osoby fizycznej. Wypełniany, kiedy ActivityKind = NON_ACCOUNTED_ACTIVITY.

```
<xsd:complexType name="PhysicalPerson">
  <xsd:sequence>
    <xsd:element name="FirstName" type="xsd:string"/>
    <xsd:element name="LastName" type="xsd:string"/>
    <xsd:element name="Pesel" type="xsd:string"/>
    <xsd:element name="DocumentType" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```

```

minOccurs="0"/>
        <xsd:element name="Document" type="xsd:string"/>
        <xsd:element name="DocumentExpirationDate" type="xsd:string"/>
        <xsd:element name="DocumentCountryId" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="Citizenship" type="xsd:string"/>
        <xsd:element name="BirthDate" type="xsd:string" minOccurs="0"/>
        <xsd:element name="BirthCountry" type="xsd:string"
minOccurs="0"/>
        </xsd:sequence>
</xsd:complexType>

```

Opis pól komunikatu

nazwa pola	wymagany	typ	opis
FirstName	TAK	string	Imię osoby fizycznej.
LastName	TAK	string	Nazwisko osoby fizycznej.
Pesel	NIE	string	PESEL osoby fizycznej. W przypadku braku peselu należy podać datę oraz państwo urodzenia.
DocumentType	TAK	string	Typ dokumentu. Akceptowane wartości: PASSPORT ID
Document	TAK	string	Numer dokumentu.
DocumentExpirationDate	TAK	string	Data ważności dokumentu.
DocumentCountryId	TAK	string	ID państwa dokumentu osoby fizycznej. Wymagany w przypadku DocumentType=PASSPORT.
Citizenship	TAK	string	Obywatelstwo osoby fizycznej.
BirthDate	NIE	string	Data urodzenia osoby fizycznej w formacie YYYY-mm-dd (np. 1990-01-01). Podawana, kiedy nie jest znany PESEL lub beneficjent nie posiada numeru PESEL.
BirthCountry	NIE	string	Państwo urodzenia osoby fizycznej.

Zasady weryfikacji na formularzu danych dotyczących beneficjentów

Weryfikacja danych na formularzu odbywa się głównie w oparciu o formę prawną firmy. W zależności od niej oczekiwane są typy beneficjentów, określone są ich wzajemne wykluczenia oraz maksymalna ilość wystąpień dla każdego typu.

Typ beneficjenta jest definiowany na formularzu poprzez zaznaczenie przez Klienta odpowiedniego oświadczenia.

Forma prawna	Oświadczenie	Typ Beneficjenta	Maks. wystąpień w formularzu
NON_ACCOUNTED_ACTIVITY - Działalność nieewidencjonowana osoby fizycznej	Jestem beneficjentem rzeczywistym - nie istnieje inna osoba fizyczna wywierająca na mnie wpływ lub sprawująca nade mną kontrolę UWAGA: Klient wypełnia dane beneficjanta tylko w przypadku zaznaczenia oświadczenia dla typu BENEFICJENT INNY.	BENEFICJENT WŁAŚCICIEL	0
	Nie jestem beneficjentem rzeczywistym - istnieje osoba fizyczna wywierająca na mnie wpływ lub sprawująca nade mną kontrolę.	BENEFICJENT INNY	1
PROPRIETORSHIP - Jednoosobowa działalność gospodarcza	Jestem beneficjentem rzeczywistym - nie istnieje inna osoba fizyczna wywierająca na mnie wpływ lub sprawująca nade mną kontrolę UWAGA: Klient wypełnia dane beneficjanta tylko w przypadku zaznaczenia oświadczenia dla typu BENEFICJENT INNY.	BENEFICJENT WŁAŚCICIEL	0
	Nie jestem beneficjentem rzeczywistym - istnieje osoba fizyczna wywierająca na mnie wpływ lub sprawująca nade mną kontrolę.	BENEFICJENT INNY	1

Forma prawna	Oświadczenie	Typ Beneficjenta	Maks. wystąpień w formularzu
CIVIL_PARTNERSHIP - Spółka cywilna	Jestem beneficjentem rzeczywistym - nie istnieje inna osoba fizyczna wywierająca na mnie wpływ lub sprawująca nade mną kontrolę UWAGA: Klient wypełnia dane beneficjanta tylko w przypadku zaznaczenia oświadczenia dla typu BENEFICJENT INNY.	BENEFICJENT WŁAŚCICIEL	0
	Nie jestem beneficjentem rzeczywistym - istnieje osoba fizyczna wywierająca na mnie wpływ lub sprawująca nade mną kontrolę.	BENEFICJENT INNY	8
GENERAL_PARTNERSHIP - Spółka jawna	Beneficjentem rzeczywistym są wspólnicy Spółki UWAGA: Klient wypełnia dane beneficjanta tylko w przypadku zaznaczenia oświadczenia dla typu BENEFICJENT INNY.	BENEFICJENT WSPÓLNIK	0
	Istnieje osoba fizyczna, inna niż wspólnicy, która wywiera wpływ lub sprawuje kontrolę nad Spółką	BENEFICJENT INNY	8
LIMITED_LIABILITY_PARTNERSHIP - Spółka partnerska	WSKAZÓWKA: Zasady weryfikacji danych beneficjanta analogiczne do GENERAL_PARTNERSHIP.		
LIMITED_PARTNERSHIP - Spółka komandytowa	WSKAZÓWKA: Zasady weryfikacji danych beneficjanta analogiczne do GENERAL_PARTNERSHIP.		

Forma prawna	Oświadczenie	Typ Beneficjenta	Maks. wystąpien w formularzu
LIMITED_JOINT_STOCK_PARTNERSHIP - Spółka komandytowo-akcyjna	Istnieje osoba fizyczna, która ma pośrednio lub bezpośrednio więcej niż 25% akcji/głosów UWAGA: Klient wypełnia dane beneficjanta tylko w przypadku zaznaczenia oświadczenia dla typu BENEFICJENT UDZIAŁOWIEC.	BENEFICJENT UDZIAŁOWIEC	3
	Nie istnieje osoba fizyczna, która ma pośrednio lub bezpośrednio więcej niż 25% akcji/głosów	BENEFICJENT WSPÓLNIK	0
SP_ZOO – Spółka z ograniczoną odpowiedzialnością	Istnieje osoba fizyczna, która ma pośrednio lub bezpośrednio więcej niż 25% akcji/głosów UWAGA: Typ BENEFICJENT KIEROWNICTWO nie może współistnieć z innymi typami beneficjentów. Typy BENEFICJENT UDZIAŁOWIEC oraz BENEFICJENT KONTROLA mogą ze sobą współistnieć więc klient może zaznaczyć na formularzu oba oświadczenia odpowiadające tym typom.	BENEFICJENT UDZIAŁOWIEC	3
	Istnieje osoba fizyczna, sprawująca kontrolę poprzez jednostkę dominującą według przepisów o rachunkowości	BENEFICJENT KONTROLA	8
	Osoby zajmujące wyższe stanowisko kierownicze w Spółce (zarząd lub rada nadzorcza)	BENEFICJENT KIEROWNICTWO	8
SA – Spółka akcyjna	WSKAZÓWKA: Zasady weryfikacji danych beneficjanta analogiczne do SP_ZOO.		

Forma prawna	Oświadczenie	Typ Beneficjenta	Maks. wystąpienie w formularzu
SOCIETY – Stowarzyszenie	Beneficjentem rzeczywistym jest zarząd stowarzyszenia UWAGA: W komunikacie może wystąpić tylko jeden z typów beneficjentów: BENEFICJENT KIEROWNICTWO lub BENEFICJENT INNY. Typy te się wzajemnie wykluczają.	BENEFICJENT KIEROWNICTWO	8
	Istnieje osoba fizyczna, inna niż zarząd stowarzyszenia, która wywiera wpływ lub sprawuje kontrolę nad stowarzyszeniem	BENEFICJENT INNY	2
FOUNDATION – Fundacja	Beneficjentem rzeczywistym jest zarząd stowarzyszenia UWAGA: Klient może zaznaczyć na formularzu tylko jedno z oświadczeń. Typy im odpowiadające wzajemnie wykluczają.	BENEFICJENT KIEROWNICTWO	8
	Istnieje osoba fizyczna, inna niż zarząd stowarzyszenia, która wywiera wpływ lub sprawuje kontrolę nad fundacją	BENEFICJENT INNY	2
COOPERATIVE – Spółdzielnia	Beneficjentem rzeczywistym jest zarząd spółdzielni UWAGA: Klient może zaznaczyć na formularzu tylko jedno z oświadczeń. Typy im odpowiadające wzajemnie wykluczają.	BENEFICJENT KIEROWNICTWO	8
	Istnieje osoba fizyczna, inna niż zarząd spółdzielni, która wywiera wpływ lub sprawuje kontrolę nad spółdzielnią	BENEFICJENT INNY	2

Forma prawna	Oświadczenie	Typ Beneficjenta	Maks. wystąpień w formularzu
GOVERNMENT - Organ administracji rządowej, organ samorządu terytorialnego lub organ egzekucyjny (gminy, urzędy)	WSKAZÓWKA: Informacje o beneficjentach nie są zbierane.		
SELF_GOVERNMENTAL_UNIT - Jednostka samorządu terytorialnego	WSKAZÓWKA: Informacje o beneficjentach nie są zbierane.		
SELF_GOVERNMENTAL_CULTURAL_INSTITUTION - Samorządowa instytucja kultury	WSKAZÓWKA: Informacje o beneficjentach nie są zbierane.		
STATE_OWNED_ENTERPRISE - Przedsiębiorstwo państwowe	WSKAZÓWKA: Informacje o beneficjentach nie są zbierane.		
STATE_CULTURAL_INSTITUTION - Państwowa instytucja kultury	WSKAZÓWKA: Informacje o beneficjentach nie są zbierane.		
PUBLIC_SECTOR_INSTITUTION - Instytucja gospodarki budżetowej	WSKAZÓWKA: Informacje o beneficjentach nie są zbierane.		
RESEARCH_INSTITUTION - Instytut badawczy	WSKAZÓWKA: Informacje o beneficjentach nie są zbierane.		
CHURCH - Kościelna osoba prawna lub jej jednostka organizacyjna	Beneficjentem jest organ osoby prawnej (np. dla parafii beneficjentem rzeczywistym jest proboszcz) UWAGA: Klient może zaznaczyć na formularzu tylko jedno z oświadczeń. Typy im odpowiadające wzajemnie wykluczają.	BENEFICJENT KIEROWNICTWO	8
	Istnieje inna osoba fizyczna, inna niż organ kościelnej osoby prawnej, która wywiera wpływ lub sprawuje nad nią kontrolę	BENEFICJENT INNY	2

Forma prawna	Oświadczenie	Typ Beneficjenta	Maks. wystąpień w formularzu
FOREIGN – Przedsiębiorca zagraniczny	Istnieje osoba fizyczna, która ma pośrednio lub bezpośrednio więcej niż 25% udziałów/głosów UWAGA: Klient może zaznaczyć na formularzu tylko jedno z oświadczeń. Typy im odpowiadające wzajemnie wykluczają.	BENEFICJENT UDZIAŁOWIEC	2
	Istnieje osoba fizyczna sprawująca kontrolę poprzez jednostkę dominującą według przepisów o rachunkowości	BENEFICJENT KONTROLA	8
	Osoby zajmujące wyższe stanowisko kierownicze w firmie (zarząd lub rada nadzorcza)	BENEFICJENT KIEROWNICTWO	8

Lista parametrów dla Przelewu weryfikacyjnego

Partner zakładający nowe konto w Systemie zobowiązany jest wykonać Przelew weryfikacyjny. Po jego wykonaniu AP przeprowadza ocenę wiarygodności, prawidłowości danych Partnera oraz ocenę AML.

Kiedy Partner zaktualizuje dane, AP oczekuje wykonania Przelewu weryfikacyjnego – wyłącznie w przypadku wrażliwych i kluczowych danych.

Lista parametrów, dla których rejestracja lub edycja danych serwisu spowoduje wygenerowanie linku Przelewu weryfikacyjnego

Nazwa pola

- Name
- Address
- Email
- Nip
- Krs
- ActivityKind
- ServiceUrl
- SettlementNRB
- NumericTrade
- TradeRegisterName
- RegistrationDate
- SwiftCode
- FirstName
- LastName

- Pesel
- DocumentType
- Document
- DocumentExpirationDate
- DocumentCountryId
- Citizenship
- BirthDate
- BirthCountry

UWAGA: ServiceUrl jest traktowany jako taki sam, jeśli:

a. różni się tylko protokołem (http/https)

b. w trakcie integracji z Partnerem został potwierdzony proces migracji adresu Serwisu z/do subdomeny CMSa Integratora (np. *przykladowastrona.integrator.pl = przykladowastrona.pl*)

c. została pominięta lub dodana składowa „www” adresu sklepu (np. *przykladowastrona.pl = www.przykladowastrona.pl*)

UWAGA: NIP, KRS oraz forma prawna są danymi, których nie można edytować. W przypadku konieczności zmiany tych danych należy ponownie zarejestrować Serwis.

Link do przelewu weryfikacyjnego

Link do Przelewu weryfikacyjnego oraz wszystkie jego notyfikacje ITN, zawierają wymieniony podczas integracji dedykowany dla tego procesu ServiceID oraz OrderID zbudowany według schematu:

```
OrderID = ActivationServiceID_UID
```

gdzie:

- **ActivationServiceID** – jest wartością ServiceID weryfikowanego Serwisu
- **UID** – jest ciągiem znaków, gwarantującym unikalność Przelewu weryfikacyjnego danego Serwisu.
- **Wszystkie komunikaty** (start oraz powrót z transakcji weryfikacyjnej, a także wszystkie jej notyfikacje ITN, przekazujące informacje o postępie procesu weryfikacji) podpisane są kluczem współdzielonym dla usługi Przelewu weryfikacyjnego (również wymienionym podczas integracji).

WSKAZÓWKA: Parametry ITN dedykowane do procesu weryfikacji opisane są w ramach części [Dodatkowe pola w komunikacie ITN/IPN transakcji wejściowej](#). W szczególności mowa tutaj o polach **verificationStatus** oraz **verificationStatusReasons**.

WSKAZÓWKA: Możliwe przejścia statusów płatności, weryfikacji oraz ich szczegółów zawierają części: *Szczegółowy opis zmiany statusu weryfikacji – dla transakcji zakończonej poprawnie (wynik pozytywny lub negatywny)*, *Szczegółowy opis zmiany statusu weryfikacji – dla transakcji nie zakończonej poprawnie*.

Rejestracja i obsługa Punktów Rozliczeń pomiędzy AP a Platformą Marketplace Partnera

Opis działania usługi automatycznej rejestracji i obsługi Punktów Rozliczeń

Niniejszy dokument opisuje zasady związane z wymianą komunikatów pomiędzy AP, a Platformą Marketplace Partnera w ramach funkcjonalności dodawania Punktów Rozliczeń (technologia REST).

Partner powinien na swojej Platformie Marketplace udostępnić przycisk, którego kliknięcie przez Klienta spowoduje wywołanie w systemie AP metody generującej aktualny link do formularza pozwalającego na rejestrację Punktu Rozliczeń.

Po otrzymaniu wspomnianego linku Platforma Marketplace powinna wykonać automatyczne przekierowanie na formularz znajdujący się pod tym linkiem.

Wypełnienie i wysłanie danych Sklepu przez Klienta spowoduje rejestrację Punktu Rozliczeń w systemie AP oraz zwrócenie strony z podziękowaniem oraz linkiem do weryfikacji online, celem sprawdzenia poprawności wprowadzonych danych. Link weryfikacyjny jest wysyłany również na adres email podany przez Klienta na formularzu.

Po wykonaniu płatności weryfikacyjnej przez Klienta oraz otrzymaniu przez AP niezbędnych danych z Kanału Płatności, Punkt Rozliczeń otrzymuje końcowy status weryfikacji. Jego pozytywna wartość powoduje aktywację Punktu Rozliczeń i gotowość do wykonywania płatności przy jego udziale, a do Klienta zostanie wysłana wiadomość z regulaminem zaakceptowanym przez niego na formularzu rejestracyjnym.

WAŻNE! Produkcyjna wersja usługi znajduje się za firewallem. Dostęp do niej jest przydzielany dla skończonej i zdefiniowanej w trakcie integracji puli IP. Nie dotyczy to środowiska testowego.

WAŻNE! Dla jednej Platformy Marketplace na danym środowisku (test/produkcja) przewidziany jest jeden identyfikator Platformy (MarketplaceID) oraz klucz współdzielony dla usługi rejestracji.

WAŻNE! Niedopuszczalne jest udostępnianie w jakiegokolwiek formie (również w kodzie aplikacji uruchamianej na serwerze osób trzecich) danych autoryzacyjnych do usługi tj. MarketplaceID i klucza współdzielonego.

Dane wymieniane podczas integracji obsługi Punktów Rozliczeń

Dane wymieniane w środowisku testowym

Dane	Przekazywane przez AP do Partnera	Przekazywane przez Partnera do AP
Adres REST	✓	✗
MarketplaceID	✓	✗
Metoda : link	✓	✗

Dane	Przekazywane przez AP do Partnera	Przekazywane przez Partnera do AP
ServiceID (dla usługi Przelewu weryfikacyjnego)	✓	✗
Klucz współdzielony dla usługi rejestracji Punktu Rozliczeń	✓	✗
Klucz współdzielony dla usługi Przelewu weryfikacyjnego	✓	✗
Mechanizm funkcji skrótu	✓	✗
Adres testowego formularza	✓	✗
Adres IP, z którego wysyłane są ITNy informujące o statusie weryfikacji Punktu Rozliczeń	✓	✗
Adres do panelu administracyjnego dla Partnera (opcja)	✓	✗
Login	✓	✗
Hasło	✓	✗
Adres ITN po przelewie weryfikacyjnym	✗	✓
Adres powrotu po przelewie weryfikacyjnym	✗	✓

Dane przekazywane w środowisku produkcyjnym

Dane	Przekazywane przez AP do Partnera	Przekazywane przez Partnera do AP
Adres REST	✗	✗
MarketplaceID	✓	✗
Metoda : link	✓	✗
ServiceID (dla usługi Przelewu weryfikacyjnego)	✓	✗
Klucz współdzielony dla usługi rejestracji Punktu Rozliczeń	✓	✗
Klucz współdzielony dla usługi Przelewu weryfikacyjnego	✓	✗
Mechanizm funkcji skrótu	✓	✗
Adres IP, z którego wysyłane są ITNy informujące o statusie weryfikacji Punktu Rozliczeń	✓	✗
Adres do panelu administracyjnego dla Partnera (opcja)	✓	✗
Login	✓	✗
Hasło	✓	✗

Dane	Przekazywane przez AP do Partnera	Przekazywane przez Partnera do AP
Adres IP, z którego następuje połączenie do WS	✘	✓
Adres ITN po przelewie weryfikacyjnym	✘	✓
Adres powrotu po przelewie weryfikacyjnym	✘	✓

Proces dodawania Punktów Rozliczeń

Opis procesu

Wymiana komunikatów pomiędzy AP, a Serwisem Partnera realizujących funkcjonalności dodawania Punktów Rozliczeń odbywa się za pomocą technologii REST. Wszystkie parametry w komunikatach przekazywane są metodą POST. Protokół rozróżnia wielkość liter, zarówno w nazwach, jak i wartościach parametrów. Wartości przekazywanych parametrów powinny być kodowane w UTF-8.

Pobranie aktywnego linku do formularza rejestracyjnego

Po kliknięciu przez klienta przycisku rejestracji na Platformie Marketplace, na adres https://adres_uslugi/api/marketplace/link powinno zostać wysłane odpytanie o aktualny link do formularza rejestracyjnego. Komunikat powinien zostać wysłany z nagłówkiem Content-Type: application/json oraz parametrami wskazanymi w tabeli poniżej:

Opis pól komunikatu

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	marketplaceID	TAK	string	Stały unikalny identyfikator Marketplace nadany przez System płatności online.
2	messageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łańciskiego (np. na bazie UID). Wartość pola musi być unikalna dla każdego odpytania o link do formularza.

kolejność Hash	nazwa	wymagany	typ	opis
nd.	hash	TAK	string{64}	Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wyłącznie wartości pól, bez nazw parametrów. Kolejność sklejania pól jest zgodna z kolejnością ich występowania na liście parametrów. Parametry w łańcuchu są oddzielane od siebie znakiem " ". Do powstałego w powyższy sposób łańcucha doklejany jest na jego końcu klucz przekazany podczas integracji. Z tak powstałego łańcucha obliczana jest wartość funkcji skrótu SHA256 i stanowi ona wartość pola Hash komunikatu. Hash=SHA256(wartości_pól_komunikatu + klucz_współdzielony)

Przykład wyliczenia Hash dla poniższego komunikatu:

```
{
  "marketplaceID":1,
  "messageID":"12345678901234567890123456789000",
  "hash":"wartosc_hash"
}
```

gdzie

$$\text{Hash} = \text{SHA256}(1 | 12345678901234567890123456789000 | \text{klucz_wspoldzielony}) = \text{wartosc_hash}$$

Odpowiedź na żądanie

W odpowiedzi na powyższe żądanie zwracany jest (w tej samej sesji HTTP) komunikat zawierający link do formularza lub w przypadku błędu jego opis wraz ze wskazaniem pola, którego dotyczy.

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	typ	opis
1	link	string	Wygenerowany link do formularza rejestracyjnego.
2	messageID	string{32}	Pseudolosowy identyfikator komunikatu. Jego wartość odpowiada wartości otrzymanej w komunikatu wysłanego z Platformy Marketplace.

kolejność Hash	nazwa	typ	opis
nd.	hash	string{64}	Wartość funkcji skrótu, służąca do autentykacji komunikatu, obliczana jest od łańcucha zawierającego sklejone pola komunikatu (konkatenacja pól). Sklejane są wyłącznie wartości pól, bez nazw parametrów. Kolejność sklejania pól jest zgodna z kolejnością ich występowania na liście parametrów. Do powstałego w powyższy sposób łańcucha doklepany jest na jego końcu klucz, współdzielony między System Platformy i System płatności online. Z tak powstałego łańcucha obliczana jest wartość funkcji skrótu SHA256 i stanowi ona wartość pola Hash komunikatu. <i>Hash = SHA256(wartości_pól_komunikatu + klucz_współdzielony)</i>
nd.	errors	-	Lista błędów w komunikacie wysłanym z Platformy Marketplace.
nd.	errors -> field	string	Nazwa pola, którego dotyczy błąd.
nd.	errors -> error	string	Opis błędu.

UWAGA: Po otrzymaniu przez Platformę Marketplace linku do formularza powinno zostać wykonane automatyczne przekierowanie na tenże formularz.

UWAGA: Każda próba rejestracji powinna być poprzedzona pobraniem linka do formularza.

UWAGA: Link do formularza jest aktywny domyślnie przez 24 godziny (wartość może być zmieniona na prośbę Partnera) od jego wygenerowania. Po upłygnięciu tego okresu wyświetlenia formularza rejestracyjnego nie jest możliwe.

Przykłady wymienianych komunikatów podczas pobierania linka do formularza

a) Poprawnie przetworzony request.

REQUEST

```
{
  "marketplaceID":1,
  "messageID": "12345678901234567890123456789000",
  "hash": "437c941b5c88b9d0bfa58a1d073d724f55cfed794c1cf9648ba2340c3dcc803f"
}
```

RESPONSE

```
{
  "link":
  "https://adres_usługi/marketplace/ee0e88d81c11b110c1a35a740996453d745dcde0be449686be6ec0777a02be5d",
  "messageID": "12345678901234567890123456789000",
  "hash": "d01e20cc8ab2c3fbe907e805329d915a20058e96f1d6d91a863361444cbd6379"
}
```

b) MessageID został był już w użyciu, nie jest unikatowy.

REQUEST

```
{
  "marketplaceID":1,
  "messageID":"12345678901234567890123456789012",
  "hash":"e43ff5f4b3121d1fd267bd66aaf446b079c28292f2a966e23eb1e8a8c0aaf81a"
}
```

RESPONSE

```
{
  "errors": [
    {
      "field": "messageID",
      "error": "messageId not unique"
    }
  ]
}
```

c) Unikatowe MessageID musi mieć 32 znaki.

REQUEST

```
{
  "marketplaceID":1,
  "messageID":"1234567890123456789012345678900022222",
  "hash":"dc8d14b5e3491a700e7f2479f8c196cc0db76f41b2b2ffac1f042a6cad1e5914"
}
```

RESPONSE

```
{
  "errors": [
    {
      "field": "messageID",
      "error": "size must be between 32 and 32"
    }
  ]
}
```


d) Podany hash jest niepoprawny.

REQUEST

```
{
  "marketplaceID": 1,
  "messageID": "12345678901234567890123456789001",
  "hash": "dc8d14b5e3491a700e7f2479f8c196cc0db76f41b2b2ffac1f042a6cad1eaaaa"
}
```

RESPONSE

```
{
  "errors": [
    {
      "field": "hash",
      "error": "Invalid hash"
    }
  ]
}
```

Dodatkowe rozszerzenia

Alternatywne modele rozpoczęcia transakcji

Preautoryzacja kartowa

Opis ogólny działania usługi preautoryzacji kartowej

Obsługa preautoryzacji kartowej dostarcza funkcjonalność blokowania środków na karcie Klienta na pewien (np. z góry ustalony w trakcie zakładania blokady) czas, a następnie dokonywania obciążenia. Szczególnym przypadkiem jest sytuacja, w której blokada jest zdejmowana bez potrącania jakiegokolwiek kwoty (np. usługa na rzecz klienta nie została wykonana).

Wszystkie te operacje (blokowanie środków, obciążanie, wycofywanie blokady) powinny być zlecone przez API Systemu Płatności Autopay. Jeśli w czasie ważności transakcji zakładającej blokadę, nie nastąpi zlecenie obciążenia karty, System dokonuje zwolnienia środków, powiadamiając o tym fakcie standardowym komunikatem o zmianie statusu transakcji (ITN).

Również pozostałe operacje (skuteczne obciążenie, założenie blokady na karcie oraz jej późniejsze obciążenia) skutkują wysyłką komunikatu ITN. Komunikat ten jest jedyną wiążącą informacją o zmianie statusu transakcji oraz (stosowany wraz z usługą **transactionStatus**; patrz część [Odpytanie o status transakcji](#)), pomaga obsłużyć transakcję nie zrywając sesji z użytkownikiem (nawet w przypadku różnych problemów sieciowych). Synchroniczne potwierdzenia operacji (węzeł **confirmation**, służą jedynie do prezentowania wstępnej informacji o zleceniu).

Etapy transakcji preautoryzacji kartowej

Założenie blokady na wniosek Partnera

Można wyróżnić 3 podstawowe sposoby zakładania blokady na karcie:

a) Zakładanie blokady podczas autoryzacji płatności jednorazowej (Patrz [Schemat A dla Preautoryzacji](#)). Klient wypełnia formatkę kartową, po wystartowaniu transakcji, w której Partner wskazuje w parametrach startowych:

- kartowy kanał płatności (**GatewayID=1500**) oraz
- chęć zabezpieczenia środków, zamiast obciążenia (**Hold=true**)

b) Zakładanie blokady podczas inicjowania płatności automatycznej (zapisywania karty w Serwisie lub Aplikacji Mobilnej) (Patrz [Schemat B dla Preautoryzacji](#)).

WSKAZÓWKA: Cały cykl i wszystkie zdarzenia płatności automatycznej zgodnie z dokumentacją, zmodyfikowany o moment faktycznego obciążenia karty oraz założenia płatności automatycznej – w modelu preautoryzacji to operacja **transactionClear** powoduje te zdarzenia.

Klient wypełnia formatkę kartową, po wystartowaniu transakcji, w której Partner wskazuje w parametrach startowych:

- kartowy kanał płatności (**GatewayID=1503**),
- fakt zaakceptowania regulaminu usługi płatności automatycznej dostarczonego przez AP (**RecurringAcceptanceState=ACCEPTED**, lub po ustaleniach biznesowych wartości **PROMPT/FORCE**)
- wybór inicjalizacji płatności automatycznej wraz z potencjalnym obciążeniem karty (**RecurringAction=INIT_WITH_PAYMENT**)
- chęć zabezpieczenia środków, zamiast obciążenia (**Hold=true**)

c) Zakładanie blokady z wykorzystaniem wcześniej zapisanej karty (Patrz [Schemat C dla Preautoryzacji](#)).

WSKAZÓWKA: Cały cykl i wszystkie zdarzenia płatności automatycznej zgodnie z dokumentacją, zmodyfikowany o moment faktycznego obciążenia karty oraz założenia płatności automatycznej – w modelu preautoryzacji to operacja **transactionClear** powoduje te zdarzenia.

Klient nie wypełnia formatki kartowej, następuje natomiast backendowa (bez przekierowania) przedtransakcja, w której Partner wskazuje w parametrach startowych:

- kartowy kanał płatności (**GatewayID=1503**)
- wskazanie wcześniej dodanej karty (**ClientHash** pochodzący z **RPAN**)
- wybór metody płatności automatycznej (**RecurringAction=MANUAL**)
- chęć zabezpieczenia środków, zamiast obciążenia (**Hold=true**)

Każda z tych metod zakładania blokady skutkuje komunikatem ITN, którego status wskazuje wynik autoryzacji transakcji. Poza standardowymi statusami, w wypadku blokowania środków, System może dostarczyć w ITN status **paymentStatus=ON_HOLD**, który stanowi potwierdzenie założenia blokady środków na karcie Klienta. Ponadto ITN będzie standardowo zawierać globalny identyfikator transakcji (**remotelD**), który będzie niezbędny do późniejszego obciążenia założonej blokady.

Obciążenie karty na wniosek Partnera

Opis obciążenia karty na wniosek Partnera

Po założeniu blokady może nastąpić zlecenie przez Partnera obciążenia wcześniej zautoryzowanej karty (Patrz [Schemat D dla Preautoryzacji](#)). W tym celu należy wywołać dedykowaną usługę **transactionClear** (https://{host_bramki}/webapi/transactionClear) z odpowiednimi parametrami. Wszystkie parametry przekazywane są metodą POST (Content-Type: application/x-www-form-urlencoded). Protokół rozróżnia wielkość liter zarówno w nazwach jak i wartościach parametrów. Wartości przekazywanych parametrów powinny być kodowane w UTF-8.

Opis dostępnych parametrów dla obciążenia karty na wniosek Partnera

kolejność Hash	nazwa	wymagany	typ	opis
1	ServiceID	TAK	string{1,10}	Identyfikator Serwisu Partnera.
2	MessageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łacińskiego (np. na bazie UID), wartość pola musi być unikalna i wskazywać konkretne zlecenie wypłaty w Serwisie Partnera.
3	RemotelD	TAK	string{1,20}	Alfanumeryczny identyfikator transakcji nadany przez System oraz przekazywany do Partnera w komunikacie ITN transakcji wejściowej. Jego podanie spowoduje obciążenie karty zautoryzowanej w transakcji o wskazanym RemotelD , jeśli jest w stanie blokady (status ON_HOLD).
4	Amount	TAK	amount	Kwota obciążenia (nie może być większa niż kwota blokady); jako separator dziesiętny używana jest kropka - '.' Format: 0.00.

kolejność Hash	nazwa	wymagany	typ	opis
5	Products	TAK	string{1,10000}	Informacje o produktach wchodzących w skład transakcji, przekazywany w postaci zakodowanego protokołem transportowym Base64 XMLa. Struktura musi zawierać wszystkie podane w preautoryzacji produkty, jednak może być uproszczona (w celu identyfikacji produktu, którego kwota ma być zaktualizowana, brane będą pod uwagę jedynie productID oraz idBalancePoint, a nową kwotę należy podać w subAmount). /Wymagane dla wielu produktów podanych w preautoryzacji.
nd.	Hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis Partnera.

Potwierdzenie wykonania operacji dla obciążenia karty na wniosek Partnera

Do poprawnego odpytania należy, wraz z przekazywanymi parametrami, przesłać zdefiniowany nagłówek HTTP o odpowiedniej treści. Dołączony nagłówek powinien nosić nazwę 'BmHeader' i posiadać następującą wartość 'pay-bm', w całości powinien prezentować się następująco 'BmHeader: pay-bm'. W przypadku poprawnego komunikatu zwracany jest (w tej samej sesji HTTP) tekst w formacie XML, zawierający potwierdzenie wykonania operacji lub opis błędu.

Struktura potwierdzenia (XML)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <balancePayoff>
    <serviceID>ServiceID</serviceID>
    <messageID>MessageID</messageID>
    <remoteOutID>RemoteOutID</remoteOutID>
    <hash>Hash</hash>
  </balancePayoff>
```

Struktura potwierdzenia (XML)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <balancePayoff>
    <balancePointID>BalancePointID</balancePointID>
    <messageID>MessageID</messageID>
    <remoteOutID>RemoteOutID</remoteOutID>
    <hash>Hash</hash>
  </balancePayoff>
```

Opis zwracanych parametrów dla obciążenia karty na wniosek Partnera

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceID	TAK	string{1,32}	Identyfikator Serwisu Partnera. Pochodzi z żądania metody. Wymagany dla confirmation=CONFIRMED.
2	messageID	TAK	string{1,20}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łańciskiego (np. na bazie UID). Pochodzi z żądania metody. Wymagany dla confirmation=CONFIRMED.
3	confirmation	TAK	string{1,100}	Status potwierdzenia przyjęcia zlecenia. Może przyjmować dwie wartości: - CONFIRMED – operacja powiodła się. UWAGA: Nie oznacza to wykonania obciążenia! System asynchronicznie dostarczy ITN z paymentStatus=SUCCESS. - - NOTCONFIRMED – operacja nie powiodła się.
4	reason	NIE	string{1,1000}	Wyjaśnienie szczegółów przetwarzania żądania.
nd.	hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis Partnera. Wymagany dla confirmation=CONFIRMED.

Zwolnienie blokady na wniosek Partnera

Po założeniu blokady może nastąpić zlecenie przez Partnera jej zwolnienia (bez potrącania jakichkolwiek środków) (Patrz [Schemat E dla Preautoryzacji](#)). Do tej operacji należy użyć usługi **releaseHold** (Patrz [Anulowanie nieopłaconej transakcji](#)). Po skutecznym zainicjowaniu zwolnienia blokady (poprawnej odpowiedzi na anulowanie transakcji), System asynchronicznie dostarczy ITN z **paymentStatus=FAILURE** oraz **paymentStatusDetails=CANCELLED**.

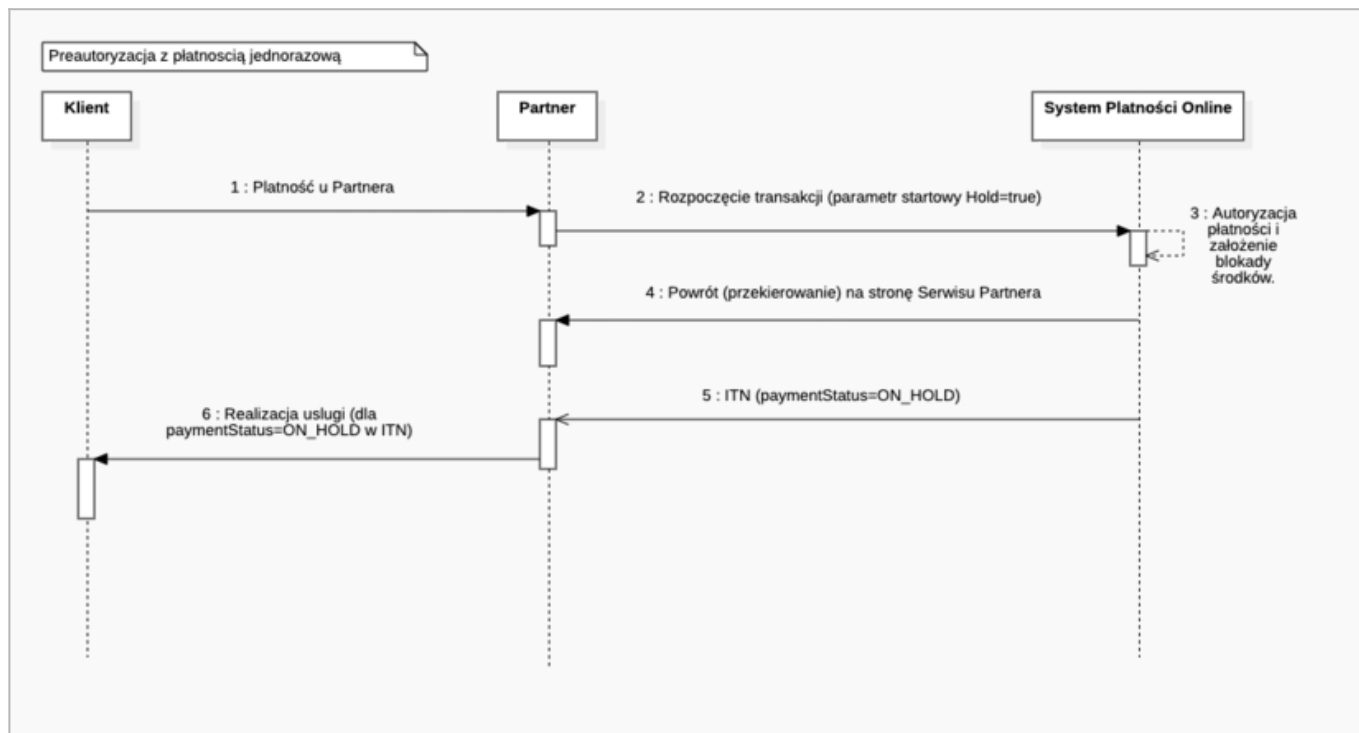
Zwolnienie blokady po przeterminowaniu transakcji

W przypadku bezczynności Partnera (po założeniu blokady) przez ustalony czas ważności transakcji, następuje jej zwolnienie przez System (bez potrącania jakichkolwiek środków) (Patrz [Schemat F dla Preautoryzacji](#)). System dokona anulowania transakcji, zdjęcia blokady oraz dostarczy ITN z

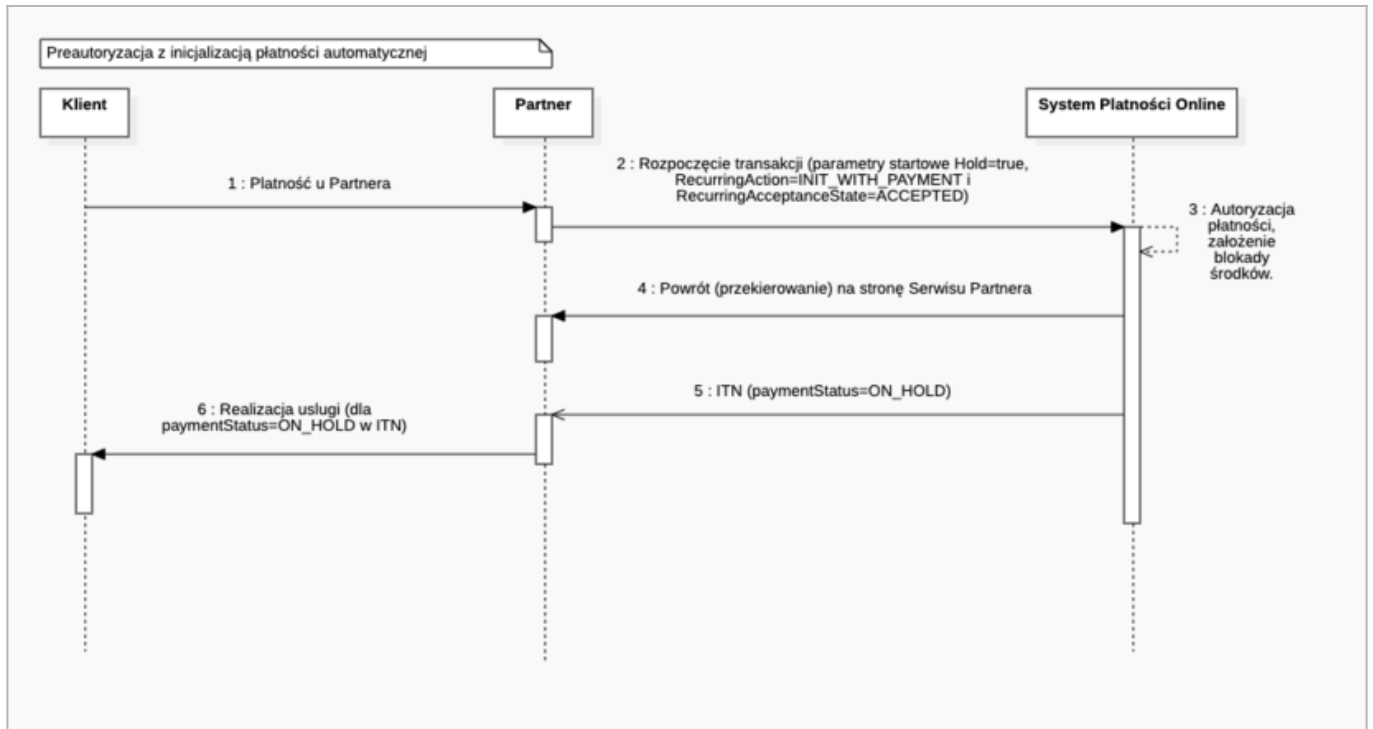
paymentStatus=FAILURE oraz **paymentStatusDetails=CANCELLED**.

Schematy dla Preautoryzacji

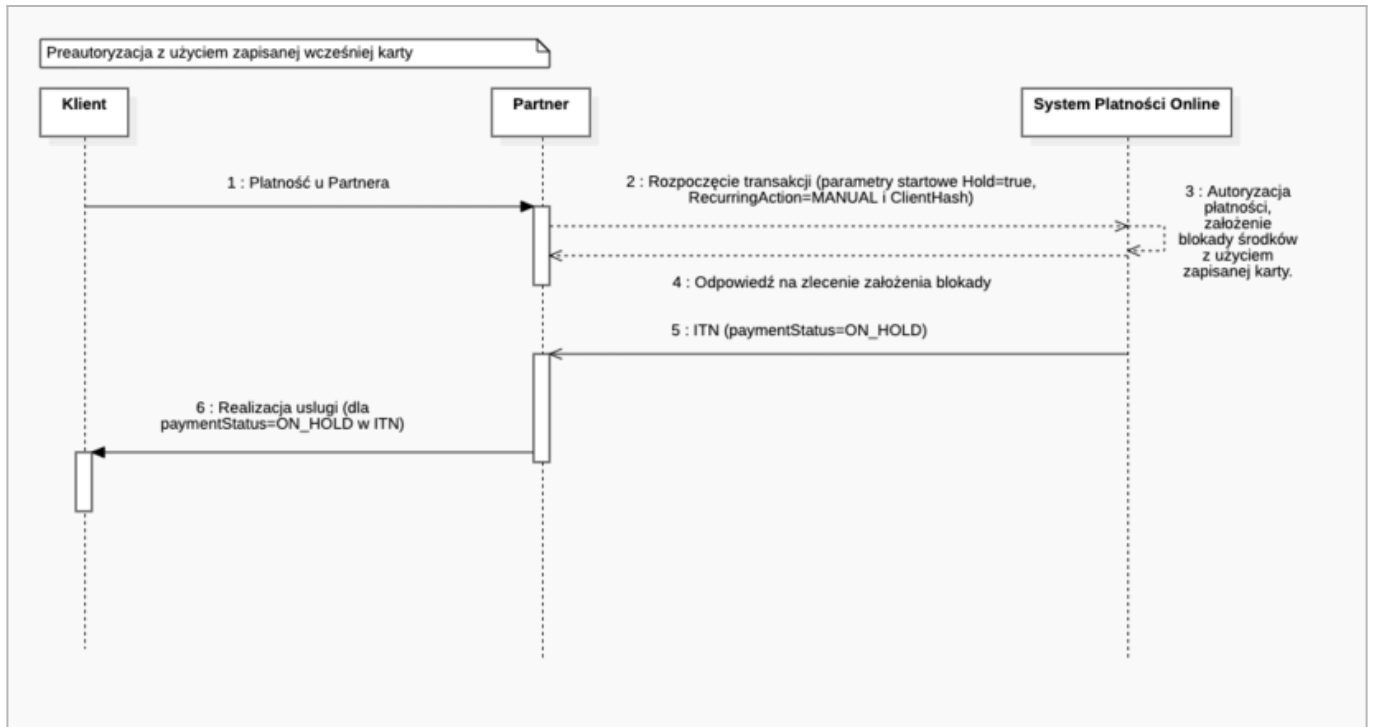
Schemat A dla Preautoryzacji: Zakładanie blokady podczas autoryzacji płatności jednorazowej



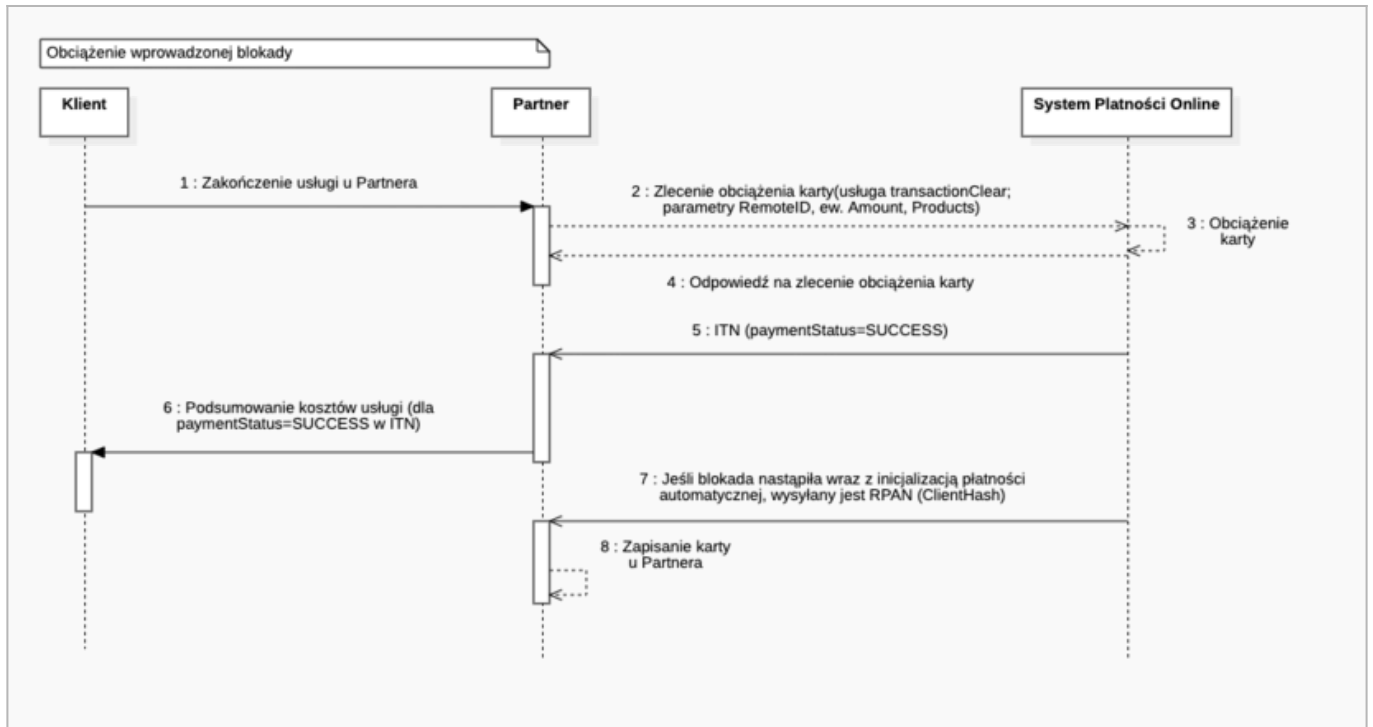
Schemat B dla Preautoryzacji: Zakładanie blokady podczas inicjowania płatności automatycznej (zapisywania karty)



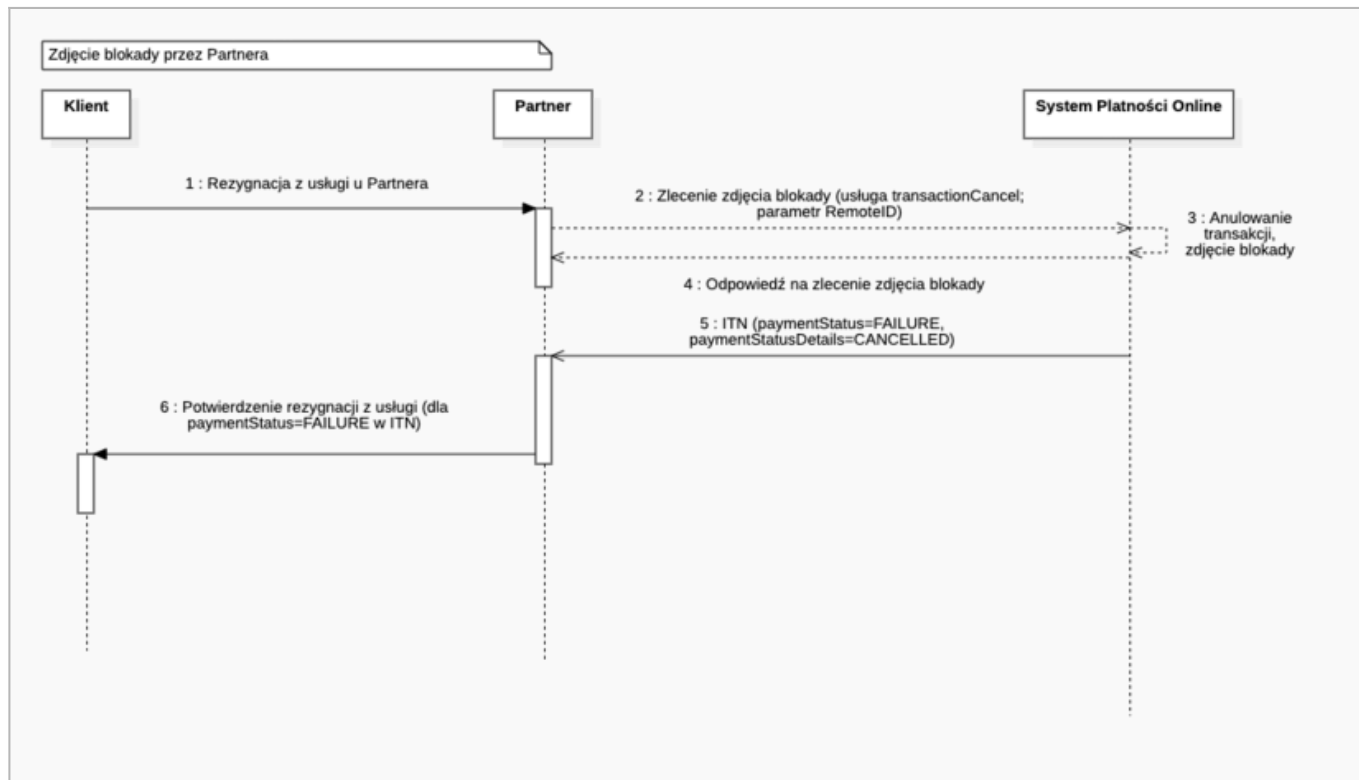
Schemat C dla Preautoryzacji: Zakładanie blokady z wykorzystaniem wcześniej zapisanej karty



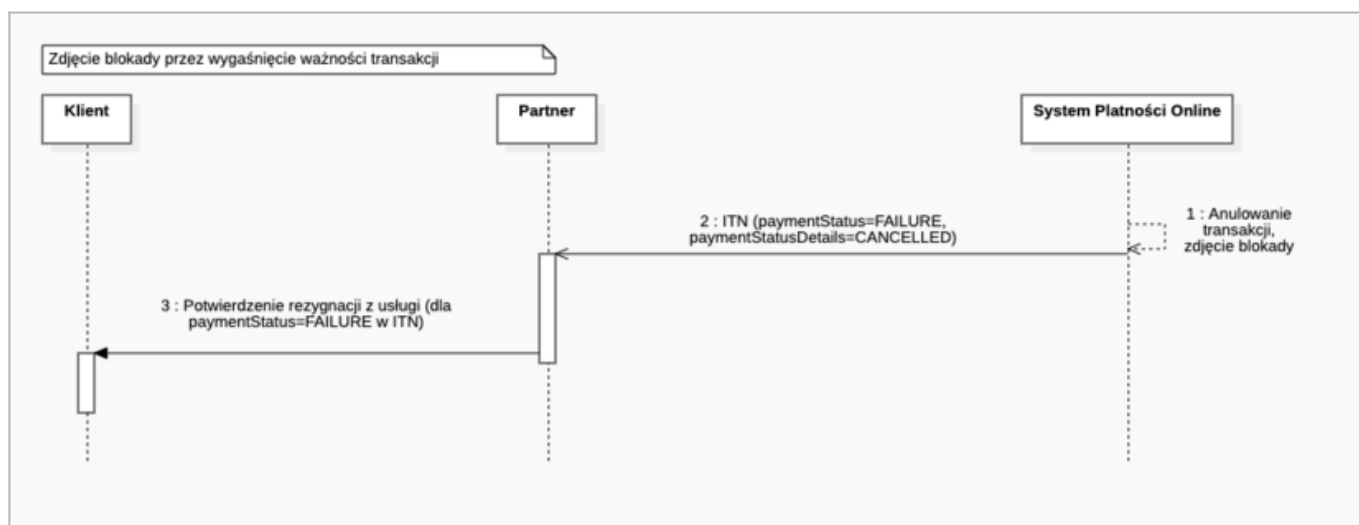
Schemat D dla Preautoryzacji: Zlecenie przez Partnera obciążenia wcześniej zautoryzowanej karty



Schemat E dla Preautoryzacji: Zlecenie przez Partnera zwolnienia blokady (bez potrącania środków)



Schemat F dla Preautoryzacji: Zwolnienie blokady przez System (bez potrącania środków)



Przedtransakcja

Opis Przedtransakcji

Przedtransakcja rozszerza standardowy model rozpoczęcia transakcji o obsługę określonych potrzeb:

- zamówienia linku do płatności na podstawie przesłanych parametrów
- obciążenia Klienta (jeśli nie jest wymagana dodatkowa autoryzacja dokonana przez Klienta)

- zweryfikowania poprawności linku płatności, zanim Klient zostanie przekierowany do Systemu – wywołanie powoduje walidację parametrów i konfiguracji Systemu
- skrócenia linka płatności – zamiast kilku/kilkunastu parametrów, link zostaje skrócony do dwóch identyfikatorów
- ukrycia danych wrażliwych parametrów linku transakcji – przedtransakcja odbywa się backendowo, a link do kontynuacji transakcji nie zawiera danych wrażliwych, a jedynie identyfikatory kontynuacji
- użycia SDK mobilnego w wariancie mieszanym – start transakcji wykonuje backend aplikacji mobilnej, a nie samo SDK z użyciem tokena transakcyjnego

WSKAZÓWKA: Szczegóły na temat wariantów SDK w [Dokumentacji SDK](#).

Szczególne przypadki użycia Przedtransakcji, to obciążenia:

- BLIK 0
Aby użyć tej usługi należy podać **GatewayID=509** oraz przekazać kod autoryzacji transakcji w parametrze **AuthorizationCode**.
- BLIK 0 OneClick
- Obciążenia „Płatności automatycznej”
Aby użyć tej usługi należy podać jeden z **GatewayID** o **gatewayType="Płatność automatyczna"** oraz niezbędne parametry.
- Autoryzacje poprzez portfele Visa
Aby użyć tej usługi należy podać **GatewayID=1511** oraz przekazać zakodowany token w parametrze **PaymentToken**. W przypadku braku tokena, autoryzacja odbędzie się na stronie Systemu.
- Autoryzacje poprzez portfele Google Pay

UWAGA: Usługa umożliwia obciążenie karty zapisanej w portfelu Klienta bez przekierowania do Systemu. Często następuje wymuszenie dodatkowej autoryzacji w postaci 3DS (domyślne zachowanie środowiska testowego, które można przekonfigurować).

W modelu **Whitelabel** należy zintegrować się zgodnie z opisem, a następnie podać **GatewayID=1512** oraz zakodowany token w parametrze **PaymentToken**. W przypadku braku

tokena (lub model inny, niż **Whitelabel**) wystarczy podać **GatewayID=1512** - autoryzacja odbędzie się na stronie Systemu.

- Autoryzacje poprzez portfele Apple Pay
Aby użyć tej usługi należy podać **GatewayID=1513**. Autoryzacja odbędzie się na stronie Systemu.
- Autoryzacja poprzez natywną formatkę SDK mobilnego

UWAGA: Usługa umożliwi obciążenie karty, której szczegóły podano na bezpiecznej formatce kartowej SDK, a sam start transakcji wykonuje backend aplikacji mobilnej.

Oprócz odpowiedniego GatewayID - 1500 dla płatności jednorazowej lub 1503 dla aktywacji płatności automatycznej (oraz innych parametrów) - należy podać uzyskany z SDK PaymentToken oraz parametr WalletType=SDK_NATIVE (opis w części [Rozpoczęcie transakcji z dodatkowymi parametrami](#))

Wywołanie Przedtransakcji

Obowiązkowym elementem w przypadku przedtransakcji jest przesłanie backendowo (używając np. cURL) standardowego komunikatu startu transakcji (patrz [Rozpoczęcie transakcji](#)), z nagłówkiem 'BmHeader' o wartości: 'pay-bm-continue-transaction-url':

Przykład nagłówka

'BmHeader: pay-bm-continue-transaction-url')

Dodatkowo zalecane jest przekazywanie parametru **CustomerIP** (do celów reklamacyjnych, sprawozdawczych).

Przykład startu Przedtransakcji (PHP)

```
$data = array(
    'ServiceID' => '100047',
    'OrderID' => '20161017143213',
    'Amount' => '1.00',
    'Description' => 'test bramki',
    'GatewayID' => '0',
    'Currency' => 'PLN',
    'CustomerEmail' => 'test@bramka.pl',
    'CustomerIP' => '127.0.0.0',
    'Title' => 'Test title',
    'Hash' => 0c5ca136e8833e40efbf42a4da7c148c50bf99f8af26f5c9400681702bd72056
);

$fields = (is_array($data)) ? http_build_query($data) : $data;

$curl = curl_init('https://{host_bramki}/test_ecommerce');
curl_setopt($curl, CURLOPT_HTTPHEADER, array('BmHeader: pay-bm-continue-transaction-url'));
curl_setopt($curl, CURLOPT_POSTFIELDS, $fields);
curl_setopt($curl, CURLOPT_POST, 1);
```

```

curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, true);
$response = curl_exec($curl);
$code = curl_getinfo($curl, CURLINFO_HTTP_CODE);
$response = curl_getinfo($curl);
curl_close($curl);

echo htmlspecialchars_decode($curlResponse);

```

Odpowiedź na Przedtransakcję - link do kontynuacji transakcji

W przypadku poprawnej walidacji parametrów (i konfiguracji) oraz potrzeby wykonania przez Klienta dodatkowej akcji (wybrania kanału płatności - jeśli podano **GatewayID=0**, wykonania/zatwierdzenia przelewu, podania kodu CVC/CVV, wykonania 3DS) - zostanie zwrócony XML z linkiem kontynuacji transakcji.

Przykład pliku z linkiem kontynuacji transakcji (XML)

```

<?xml version="1.0" encoding="UTF-8"?>
  <transaction>
    <status>PENDING</status>
    <redirecturl>
      https://{host_bramki}/payment/continue/96VSD39Z6E/L6CGP5BH
    </redirecturl>
    <orderId>20180824105435</orderId>
    <remoteID>96VSD39Z6E</remoteID>
    <hash>
      1c6eae2127f0c3f81fbed3b6372f128040729a4d4e562fb696c22e0db68dbbe1
    </hash>
  </transaction>

```

Obiekt transaction dla Przedtransakcji

Obiekt **transaction** reprezentuje wpływ lub wypłatę środków z konta AP, np. zrealizowany zakup lub zwrot.

Atrybuty obiektu transaction dla Przedtransakcji

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	status	TAK	string{1,32}	Status transakcji. W tym wypadku stała PENDING.
2	redirecturl	TAK	string{1,100}	Adres do kontynuacji transakcji rozpoczętej przez komunikat przedtransakcji.
3	orderId	TAK	string{1,32}	Identyfikator transakcji nadany w Serwisie Partnera i przekazany w starcie transakcji.

kolejność Hash	nazwa	wymagany	typ	opis
4	remoteID	TAK	string{1,20}	Unikalny identyfikator transakcji nadany w Systemie AP.
nd.	hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez herwis.

Odpowiedź na Przedtransakcję - brak kontynuacji transakcji

W przypadku niepoprawnej walidacji lub nieskutecznego obciążenia nie jest generowany link kontynuacji. Zwracany jest (w tej samej sesji HTTP) tekst w formacie XML, informujący o statusie przetwarzania żądania.

Przykład statusu przetwarzania żądania (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction>
  <orderID>OrderID</orderID>
  <remoteID>RemoteID</remoteID>
  <confirmation>ConfStatus</confirmation>
  <reason>Reason</reason>
  <blikAMList>
    <blikAM>
      <blikAMKey>Klucz1</blikAMKey>
      <blikAMLabel>Etykieta1</blikAMLabel>
    </blikAM>
    <blikAM>
      <blikAMKey>Klucz2</blikAMKey>
      <blikAMLabel>Etykieta2</blikAMLabel>
    </blikAM>
  </blikAMList>
  <paymentStatus>PaymentStatus</paymentStatus>
  <hash>Hash</hash>
</transaction>
```

Wynik Przedtransakcji

Parametry zwracane dla wyniku Przedtransakcji.

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	orderID	TAK	string{1,32}	Identyfikator transakcji nadany w Serwisie Partnera i przekazany w starcie transakcji. Wymagany dla confirmation=CONFIRMED.

kolejność Hash	nazwa	wymagany	typ	opis
2	remoteID	TAK	string{1,20}	Unikalny identyfikator transakcji nadany w Systemie AP. Wymagany dla confirmation=CONFIRMED.
3	confirmation	TAK	string{1,100}	Status potwierdzenia przyjęcia zlecenia. Może przyjmować dwie wartości: - CONFIRMED - operacja powiodła się. UWAGA: Nie oznacza obciążenia. - NOTCONFIRMED - operacja nie powiodła się.
4	reason	NIE	string{1,1000}	Wyjaśnienie przyczyny odrzucenia zlecenia (dla confirmation=NOTCONFIRMED), jeśli jest ona dostępna.
5	blikAMList	NIE	string{1,10000}	Lista dostępnych aplikacji mobilnych banków w opcji BLIK 0 OneClick (dla confirmation=NOTCONFIRMED oraz reason=ALIAS_NONUNIQUE).
				Format dla blikAMList: <pre> <blikAM> <blikAMKey>Klucz1</blikAMKey> <blikAMLabel>Etykieta1</blikAMLabel> </blikAM> ... <blikAM> <blikAMKey>KluczN</blikAMKey> <blikAMLabel>EtykietaN</blikAMLabel> </blikAM> </pre>
6	paymentStatus	NIE	enum	Status autoryzacji transakcji, przyjmuje wartości: - PENDING - transakcja rozpoczęta - SUCCESS - poprawna autoryzacja transakcji - FAILURE - transakcja nie została zakończona poprawnie
nd.	hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis. Wymagany dla confirmation=CONFIRMED.

Poprawna walidacja parametrów

W przypadku poprawnej walidacji parametrów (i konfiguracji) oraz braku potrzeby wykonania przez

Klienta dodatkowej akcji – zwracane jest potwierdzenie zlecenia obciążenia.

Ma to miejsce w przypadkach, gdzie dane są wystarczające do wykonania obciążenia dla danego kanału płatności, na przykład: BLIK 0 bez wymaganego kodu BLIK (ani wskazania aliasu aplikacji mobilnej banku), płatność cykliczna, płatność Kartą OneClick bez wymaganego CVC/CVV/3DS.

Result

confirmation=CONFIRMED

Niepoprawna walidacja parametrów

W przypadku niepoprawnej walidacji parametrów (i konfiguracji) – zwracany jest błąd.

Result

confirmation=NOTCONFIRMED

Błąd może być również zwrócony w przypadku synchronicznej odpowiedzi z Kanału Płatności (np. błąd specyficzny dla próby inicjalizacji płatności automatycznej BLIK, tj. reason=RECCURRENCY_NOT_SUPPORTED).

UWAGA: Błąd może być również zwrócony w przypadku synchronicznej odpowiedzi z Kanału Płatności (np. błąd specyficzny dla próby inicjalizacji płatności automatycznej BLIK, tj. reason=RECCURRENCY_NOT_SUPPORTED). Innym znanym przypadkiem jest też błąd walidacji adresu podanego w parametrze startowym CustomerEmail (INVALID_EMAIL).

Obsługa odpowiedzi dla Przetransakcji

Status potwierdzenia przyjęcia zlecenia (confirmation)	Status Płatności (paymentStatus)	Opis zachowania Partnera
CONFIRMED	SUCCESS	Przyjęto transakcję do przetwarzania, status poprawny. Nie należy ponawiać próby obciążenia. Można wyświetlić potwierdzenie płatności, ale procesy biznesowe powinny być wstrzymane do potwierdzenia w ITN (zostanie ono wysłane po otrzymaniu przez AP poprawnego statusu transakcji z Kanału Płatności).
CONFIRMED	FAILURE	Przyjęto transakcję do przetwarzania, status niepoprawny. Można ponowić próbę obciążenia z tym samym OrderID . Po otrzymaniu przez AP statusu transakcji z Kanału Płatności, wysłany zostanie komunikat ITN. UWAGA: Nie można ponawiać próby obciążenia z tym samym OrderID , jeśli w trakcie integracji uzgodniony zostanie model blokowania przez System startów transakcji z tym samym OrderID . Domyślnie zachowanie przez Partnera unikalności OrderID jest tylko zaleceniem i nie podlega weryfikacji w starcie transakcji.

Status potwierdzenia przyjęcia zlecenia (confirmation)	Status Płatności (paymentStatus)	Opis zachowania Partnera
CONFIRMED	PENDING	Przyjęto transakcję do przetwarzania, ale nieznanym jest jeszcze jej status. Nie należy ponawiać próby obciążenia. Dalsza obsługa, jak w przypadku Timeout.
NOTCONFIRMED	-	Nie zlecono transakcji (przyczyna wskazana w węźle reason). Można ponowić próbę obciążenia z tym samym OrderID. Komunikat ITN nie powinien nigdy być wysłany.
Timeout (lub inna odpowiedź, jak niepoprawna struktura, brak wymaganych pól, inny status potwierdzenia)	-	Należy poczekać na ITN do terminu ważności transakcji (w tym celu zaleca się stosowanie krótkiego czasu ważności, np. 15 min), informując Klienta o wyniku w ramach odrębnego procesu (mail/sms). Po tym czasie zaleca się odpytać o status transakcji (transactionStatus). Jeśli metoda zwróci brak zarejestrowanej transakcji (lub same statusy płatności FAILURE), można ponowić zlecenie obciążenia z takim samym OrderID . Alternatywnie można spróbować unieważnić transakcję, przyspieszając tym samym proces uzyskiwania ostatecznego statusu transakcji i ew. proces ponowienia komunikatu startu transakcji. Należy w tym celu użyć usługi anulowania transakcji (transactionCancel) oraz potwierdzić jej działanie poprzez odpytanie o status transakcji (jak opisano wyżej).

Zamówienie danych do przelewu w transakcji typu Szybki Przelew

Opis zamawiania danych do przelewu w transakcji typu Szybki Przelew

Szybki Przelew to forma płatności, która wymaga od Klienta samodzielnego przepisania danych do przelewu dostarczanych przez System. Jakiego typu jest dany Kanał płatności, mówi parametr gatewayType w odpowiedzi na wywołanie usługi „Odpytywanie o listę aktualnie dostępnych Kanałów Płatności”. Dane do przelewu mogą być Klientowi wyświetlone:

- na stronie AP (realizacja transakcji w oparciu o standardowy model startu transakcji opisany w części [Rozpoczęcie transakcji](#))
- w serwisie Partnera (realizację transakcji bez przekierowania Klienta na stronę AP opisano poniżej)

Wywołanie

Dla poprawnego nadania komunikatu należy przesłać backendowo (np. cURLeM) standardowy

komunikat startu transakcji, z nagłówkiem 'BmHeader' o wartości: 'pay-bm' (w całości nagłówek powinien prezentować się następująco 'BmHeader: pay-bm'). W przypadku błędnego zdefiniowania nagłówka lub jego braku, komunikat zostanie błędnie odczytany. Dodatkowo zalecane jest przekazywanie parametru CustomerIP zgodnie z opisem w punkcie IP użytkownika (potrzebne do procesów reklamacyjnych, sprawozdawczych) oraz **wymagane** jest przekazanie niezerowego parametru **GatewayID** (o **gatewayType** „Szybki Przelew”).

Implementacja startu transakcji w tle (PHP)

```
$data = array(
    'ServiceID' => '100047',
    'OrderID' => '20150723144517',
    'Amount' => '1.00',
    'Description' => 'test bramki',
    'GatewayID' => '71',
    'Currency' => 'PLN',
    'CustomerEmail' => 'test@bramka.pl',
    'CustomerIP' => '127.0.0.0',
    'Title' => 'Test title',
    'ValidityTime' => '2016-12-19 09:40:32',
    'LinkValidityTime' => '2016-07-20 10:43:50',
    'Hash' => 'e627d0b17a14d2faee669cad64e3ef11a6da77332cb022bb4b8e4a376076daaa'
);

$fields = (is_array($data)) ? http_build_query($data) : $data;

$curl = curl_init('https://{host_bramki}/test_ecommerce');
curl_setopt($curl, CURLOPT_HTTPHEADER, array('BmHeader: pay-bm'));
curl_setopt($curl, CURLOPT_POSTFIELDS, $fields);
curl_setopt($curl, CURLOPT_POST, 1);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, true);
$curlResponse = curl_exec($curl);
$code = curl_getinfo($curl, CURLINFO_HTTP_CODE);
$response = curl_getinfo($curl);
curl_close($curl);

echo htmlspecialchars_decode($curlResponse);
```

Odpowiedź - dane do przelewu

W przypadku płatności tego typu, System generuje komplet danych potrzebnych do wykonania wewnątrzbankowego (a więc szybkiego) przelewu na rachunek bankowy AP. Dane te umieszczane są w odpowiedzi na start transakcji, w dokumencie xml.

Odpowiedź systemu płatności na start transakcji (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
  <transaction>
    <receiverNRB>47 1050 1764 1000 0023 2741 0516</receiverNRB>
    <receiverName>Autopay</receiverName>
    <receiverAddress>81-718 Sopot, ul. Powstańców Warszawy
6</receiverAddress>
    <orderID>9IMYEH2AV3</orderID>
    <amount>1.00</amount>
    <currency>PLN</currency>
```

```

<title>9IMYEH2AV3 - weryfikacja rachunku</title>
<remoteID>9IMYEH2AV3</remoteID>
<bankHref>https://ssl.bsk.com.pl/bskonl/login.html</bankHref>
<hash> fe685d5e1ce904d059eb9b7532f9e06a64c34c1ea9fcf29b62afefdb7aad7b75
</hash>
</transaction>

```

Lista zwracanych parametrów dla odpowiedzi

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	receiverNRB	TAK	string{32}	Numer rachunku odbiorcy przelewu (AP).
2	receiverName	TAK	string{1,100}	Nazwa odbiorcy przelewu (AP).
3	receiverAddress	TAK	string{1,100}	Dane adresowe odbiorcy przelewu (AP).
5	orderID	TAK	string{1,32}	Identyfikator transakcji nadany w Serwisie Partnera i przekazany w starcie transakcji.
6	amount	TAK	amount	Kwota transakcji. Jako separator dziesiętny używana jest kropka - '.' Format: 0.00; maksymalna długość: 14 cyfr przed przecinkiem i 2 po przecinku. UWAGA: Dopuszczalna wartość pojedynczej Transakcji w Systemie produkcyjnym wynosi min. 0.01 PLN, max. 100000.00 PLN (lub do wysokości indywidualnego limitu pojedynczej transakcji w Banku dla przelewu wewnątrzbankowego).
7	currency	TAK	string{1,3}	Waluta transakcji.
8	title	TAK	string{1,140}	Pełny tytuł przelewu (ID wraz z doklejonym polem Description ze startu transakcji).
9	remoteID	TAK	string{1,20}	Unikalny identyfikator przelewu nadany w Systemie AP.
10	bankHref	TAK	string{1,100}	Adres logowania w systemie bankowości internetowej, który można wykorzystać do stworzenia przycisku „Przejdź do banku”.

kolejność Hash	nazwa	wymagany	typ	opis
nd.	hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis.

UWAGA: Powyższe informacje należy wykorzystać do wyświetlenia danych przelewowych oraz przekierowania użytkownika do strony logowania banku.

Płatność BLIK 0 OneClick

Opis płatności BLIK 0 OneClick

Jest to rozwiązanie dedykowane dla płatności BLIK, pozwalające zrealizować płatność bez podawania kodu BLIK (oraz bez konieczności opuszczania Serwisu). Jej skuteczne zainicjowanie w Systemie powoduje automatyczne uruchomienie/wybudzenie aplikacji mobilnej banku i zaprezentowanie transakcji do potwierdzenia Użytkownikowi.

Potencjalne korzyści:

- udostępnienie pierwszej wygodnej i bezpiecznej metody płatności w mCommerce niewymagającej podania numeru karty otwiera ten segment na nowych klientów,
- lepsze doświadczenie zakupowe Klienta - płaci szybciej i wygodniej,
- częstotliwość zakupów i wartość klienta w czasie - Klienci chętniej i częściej kupują w tych sklepach, w których proces zakupowy jest wygodniejszy,
- współczynnik konwersji - Serwis ma większą kontrolę nad procesem zakupu i płatności (Klient go nie opuszcza), eliminowane jest ryzyko utraty koszyka,
- szybka decyzja transakcyjna - w krótkim czasie transakcja podlega autoryzacji, odmowie lub unieważnieniu,
- Serwis ma możliwość objęcia analizą samego etapu dokonywania płatności.

Warunkiem udostępnienia Klientowi BLIK 0 OneClick jest zautoryzowanie w Serwisie (posiadanie konta oraz wcześniejsze do niego zalogowanie). Jeśli podczas wcześniej wykonywanej płatności BLIK, wraz z innymi informacjami o płatności, Serwis przesłał dedykowany Alias UID (opis parametrów **BlikUIDKey** oraz **BlikUIDLabel** w innej części dokumentu), a Klient potwierdzając płatność w

aplikacji mobilnej zaznaczył, że chce zapamiętać sklep, to skutkiem było trwałe powiązanie (typowo na okres 2 lat) Klienta Serwisu z jego aplikacją, czyli zarejestrowanie Aliasu UID. Kolejne jego użycie będzie skutkowało autoryzacją transakcji bez podania kodu.

Wywołanie płatności BLIK 0 OneClick

Zaleca się, aby przy wyborze Kanału Płatności BLIK nie wymuszać podania kodu BLIK. Warto natomiast wyświetlić hiperłącze „Chcę wprowadzić kod BLIK” pod przyciskiem „Kupuję i płacę”, aby umożliwić wpisanie kodu w pierwszej próbie (na wypadek, gdyby Klient chciał dokonać płatności BLIK z innej aplikacji mobilnej niż ta, w której wcześniej zapamiętał dany Serwis).

Serwis powinien wykonać Przedtransakcję, ze zwróceniem szczególnej uwagi na:

- podanie parametru **GatewayID** = 509 – wskazanie kanału płatności BLIK,
- podanie parametrów **BlikUIDKey** oraz **BlikUIDLabel** – wskazanie wymaganego w usłudze BLIK 0 OneClick Alias UID (identyfikatora użytkownika)
- podanie parametru **AuthorizationCode** – jeśli Klient podał kod BLIK,
- podanie parametru **BlikAMKey** – jeśli Klient wskazał etykietę aplikacji mobilnej banku spośród zaprezentowanej w Serwisie listy,
- obsługę możliwych odpowiedzi na przedtransakcję, w tym obsłużyć „Odpowiedź – brak kontynuacji” oraz błędy specyficzne dla BLIK 0 OneClick:

a) błąd wielu aplikacji mobilnych banku (**confirmation=NOTCONFIRMED** oraz **reason=ALIAS_NONUNIQUE**) – wyświetlenie listy etykiet zwróconej w przedtransakcji listy aliasów (pary klucz + etykieta zawarte w strukturze **BlikAMList**), w celu pobrania wybranego klucza i podania go w parametrze **BlikAMKey** kolejnej próby przedtransakcji

b) błędy autoryzacji (**confirmation=NOTCONFIRMED** oraz **reason** o jednej z wartości: **ALIAS_DECLINED, ALIAS_NOT_FOUND, WRONG_TICKET, TICKET_EXPIRED, TICKET_USED**) – wyświetlenie pola Kod Blik, w celu pobrania go i podania w parametrze **AuthorizationCode** kolejnej próby przedtransakcji

Przelewy do Urzędu Skarbowego

Opis

System umożliwia realizację przelewów do Urzędu Skarbowego.

Walidacja tytułów przelewów do Urzędu Skarbowego

nazwa walidatora: US_TITLE_VALIDATOR

Title="/TI/_____/OKR/_____/SFP/_____/TXT/_____ {idtransremote_out}", gdzie:

a) TI: oznacza identyfikator podatnika (P dla PESEL lub N dla NIP lub R dla REGON). Wpływa na:

- Y w NRB (0/1)
- X w NRB, który jest PESELelem, lub NIPem.

b) OKR: Rok, typ okresu i numer okresu, za który dokonywana jest płatność podatku

- R - rok w formacie dwucyfrowym
- P - półrocze
- K - kwartał
- M - miesiąc
- D - dekada
- J - dzień
- 0 (zero) - dla należności niezwiązanych z okresem rozliczeniowym

Numer okresu:

- dla R znaki 4-7 nie powinny być wypełnione
- dla P znaki 4-5 = 01 lub 02, znaki 6-7 nie wypełnione
- dla K znaki 4-5 = 01,02,03 lub 04, znaki 6-7 nie wypełnione
- dla M znaki 4-5 = 01 do 12, znaki 6-7 nie wypełnione
- dla D znaki 4-5 = 01,02 lub 03, znaki 6-7 = 01 do 12
- dla J znaki 4-5 = 01 do 31, znaki 6-7 = 01 do 12

c) SFP: symbol formularza płatności:

symbol formularza płatności	mikrorachunek	Budowa NRB, na który należy przekazać kwotę	Okres, za który dokonywana jest płatność podatku
CIT	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	
CIT-10Z	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	Należności niezwiązane z okresem rozliczeniowym.
CIT-11R	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	Należności niezwiązane z okresem rozliczeniowym.
CIT-6R	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	Należności niezwiązane z okresem rozliczeniowym.
CIT-6AR	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	Należności niezwiązane z okresem rozliczeniowym.
CIT-8	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	roczny
CIT-8A	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	roczny
CIT-8B	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	roczny
CIT-9R	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	Należności niezwiązane z okresem rozliczeniowym.

symbol formularza płatności	mikrorachunek	Budowa NRB, na który należy przekazać kwotę	Okres, za który dokonywana jest płatność podatku
CIT-CFC	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	
KP	NIE	XX XXXX XXXX XXXX XXX0 0007 0000	
SD	NIE	XX XXXX XXXX XXXX XXX0 0007 0000	Należności niezwiązane z okresem rozliczeniowym.
SD-2	NIE	XX XXXX XXXX XXXX XXX0 0007 0000	Należności niezwiązane z okresem rozliczeniowym.
PCC	NIE	XX XXXX XXXX XXXX XXX0 0007 0000	Należności niezwiązane z okresem rozliczeniowym.
PCC-2	NIE	XX XXXX XXXX XXXX XXX0 0007 0000	Należności niezwiązane z okresem rozliczeniowym.
PCC-3	NIE	XX XXXX XXXX XXXX XXX0 0007 0000	Należności niezwiązane z okresem rozliczeniowym.
PIT	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	
PIT-28	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	roczny
PIT-36	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	roczny
PIT-36L	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	roczny
PIT-37	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	roczny
PIT-38	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	roczny
PIT-39	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	roczny
PIT-4	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	miesięczny
PIT-4R	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	roczny
PPL	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	
PIT-7	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	
PIT-8A	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	miesięczny

symbol formularza płatności	mikrorachunek	Budowa NRB, na który należy przekazać kwotę	Okres, za który dokonywana jest płatność podatku
PIT-8AR	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	roczny
PIT-CFC	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	
PU1	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	Należności niezwiązane z okresem rozliczeniowym.
PPD	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	
PPE	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	
PPW	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	
VAT-7	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	miesięczny
VAT-7K	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	kwartalny. Od 1.10 te symbole będą zastąpione przez JPK_V7K oraz JPK_V7M.
VAT-7D	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	kwartalny
VAT-8	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	miesięczny
VAT-9M	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	miesięczny
VAT-10	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	Należności niezwiązane z okresem rozliczeniowym.
VAT-12	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	
VAT-14	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	
VAP-1	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	
VAI	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	Należności niezwiązane z okresem rozliczeniowym.
VAT-IM	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	
VAT-Z	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	Należności niezwiązane z okresem rozliczeniowym.

symbol formularza płatności	mikrorachunek	Budowa NRB, na który należy przekazać kwotę	Okres, za który dokonywana jest płatność podatku
VAT-In	TAK	LK 1010 0071 222Y XXXX XXXX XXXX	Należności niezwiązane z okresem rozliczeniowym.

d) TXT: dodatkowy tekst (max 29 znaków)

e) {idtransremote_out} → stała tytułu, która musi znajdować się na końcu ciągu. W to miejsce nie należy wklejać żadnych dodatkowych wartości.

Google Pay

Opis

Google Pay to błyskawiczny i intuicyjny system płatności od Google. Pozwala on użytkownikowi na przeprowadzenie procesu płatności bez wypełniania formularza kartowego, ponieważ dane karty są przechowywane bezpiecznie na serwerach firmy.

Google Pay to produkt umożliwiający uzyskanie zaszyfrowanych danych karty płatniczej klienta pozwalających na jej obciążenie.

W celu dokonania płatności przez Google Pay należy zapisać kartę płatniczą na swoim koncie Google, używając jakiegokolwiek platformy Google (np. kupując aplikacje w Google Play) lub bezpośrednio na stronie [Google Pay](#).

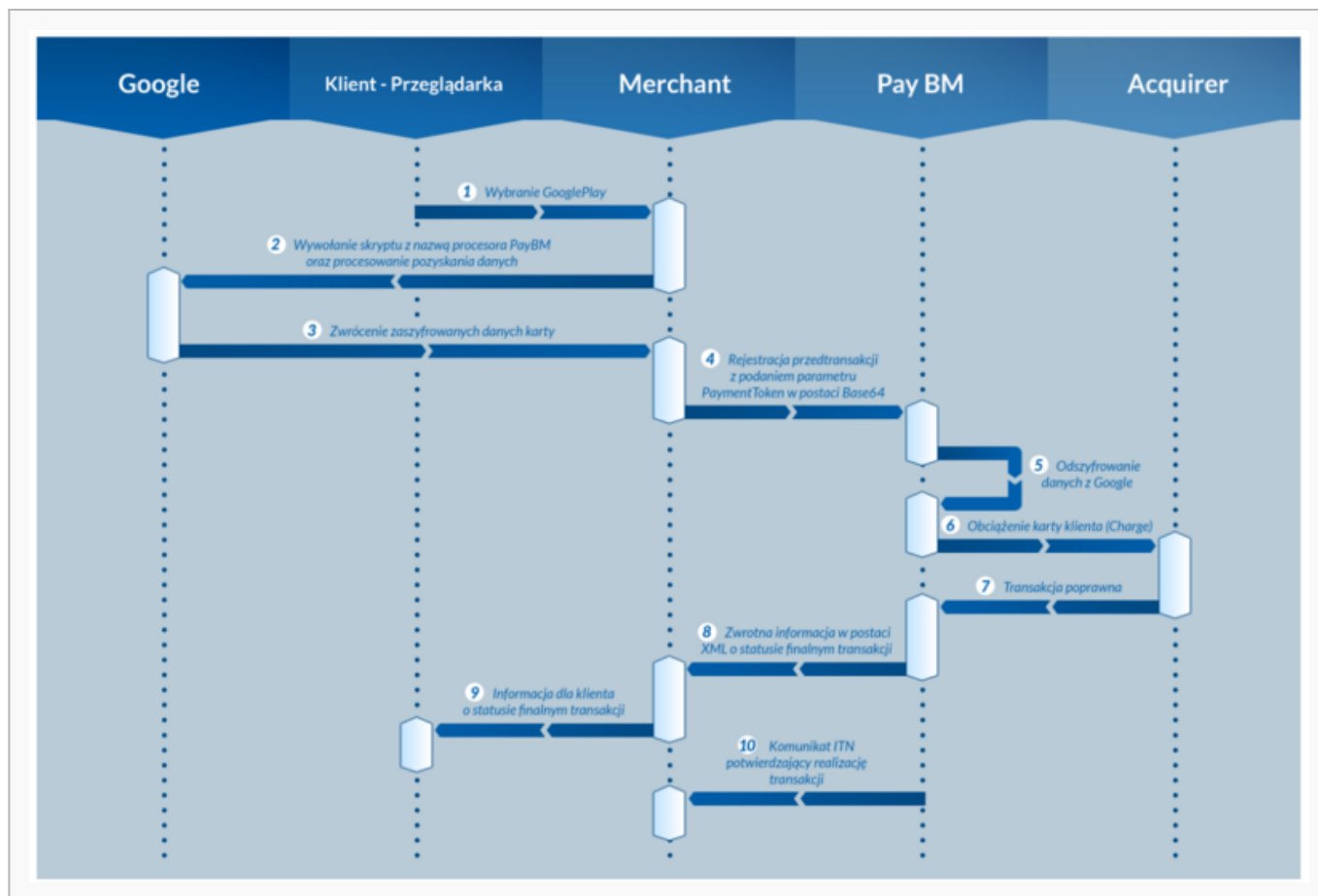
UWAGA: Usługa wymaga wcześniejszego podpisania umowy z operatorem kartowym. Po szczegółowe informacje należy zwrócić się do Działu Biznesu Autopay.

Schemat komunikacji

Po kliknięciu „Zapłać przez Google Pay” na stronie sklepu pojawia się formularz Google Pay. Klient potwierdza w nim swoje konto Google i kartę, którą zamierza zapłacić (na tym etapie może też zmienić kartę na inną lub dodać nową). Skrypt przekazuje zakodowane dane karty w tle poprzez funkcję **postMessage**, następnie sklep musi je przyjąć i zakodować przez funkcję base64 i w końcu wysłać w parametrze **PaymentToken** wraz z pozostałymi parametrami (danymi transakcji).

Na swojej stronie sklep musi wywołać skrypt udostępniony przez Google z podmienionymi danymi Procesora płatności.

WSKAZÓWKA: Szczegóły w [dokumentacji deweloperskiej Google](#).



Szczegółowy schemat komunikacji i wymiany danych

Rejestracja Transakcji Google Pay

1. Sklep na swojej stronie musi wysłać zapytanie do Systemu Płatności Online AP, aby pobrać dane potrzebne do realizacji płatności Google Pay (**paybmApiResponse**).

WSKAZÓWKA: Przykład wysłania zapytania dostępny jest na [GitHubie Autopay](#).

2. Następnie, sklep musi wywołać skrypt udostępniony w [części Samouczek dokumentacji deweloperskiej Google](#), zawierający:

a) Podmienione dane Procesora płatności:

```
const tokenizationSpecification = {
  type: 'PAYMENT_GATEWAY',
  parameters: {
    'gateway': 'bluemediia',
    'gatewayMerchantId': paybmApiResponse.acceptorId
  }
};
```

b) Dane zwrócone przez System Płatności Online AP przekazane w obiekcie **PaymentDataRequest.merchantInfo**:

```
PaymentDataRequest.merchantInfo = {
```

```
merchantId: paybmApiResponse.merchantId,  
merchantOrigin: paybmApiResponse.merchantOrigin,  
merchantName: paybmApiResponse.merchantName,  
authJwt: paybmApiResponse.authJwt,  
};
```

3. Po kliknięciu „Zapłać przez Google Pay” na stronie sklepu pojawia się formularz Google Pay. Klient potwierdza w nim swoje konto Google i kartę, którą ma zamiar zapłacić (na tym etapie może też zmienić kartę na inną lub dodać nową). Skrypt w tle przekazuje zakodowane dane karty, które sklep musi przyjąć, a następnie zakodować funkcją Base64 i wysłać w parametrze **PaymentToken** wraz z pozostałymi parametrami startowymi transakcji (tj. danymi transakcji Systemu Płatności Online AP - szczegóły opisane są w części [Rozpoczęcie transakcji z dodatkowymi parametrami](#)).

WSKAZÓWKA: Kompletny przykład integracji z Google Pay dostępny jest na [GitHubie Autopay](#).

Informację dodatkowe

Aby zachować integralność estetyczną stylistyki stosowanej na stronie www oraz w aplikacji mobilnej należy skorzystać ze wskazówek, które znajdują się w [części Brand Guidelines dokumentacji deweloperskiej Google](#) w przypadku opisów styli oraz przycisków dla stron www, oraz w [części Tutorial dokumentacji deweloperskiej Google](#), gdzie znajdują się informacje potrzebne przy tworzeniu aplikacji mobilnej.

Apple Pay

Implementacja Apple Pay na stronie sklepu.

Prośba o kontakt z produktem infrastruktury płatniczej - potrzebne wsparcie IT.

1. Utworzenie konta przez Partnera i uzyskanie certyfikatu przetwarzania płatności zgodnie z [dokumentem Configure Apple Pay \(iOS, watchOS\)](#)
 - **certyfikat komunikacyjny** - do tzw. przedstawienia się - *merchant identifier*
 - **certyfikat obciążający** - *payment processing certificate*
2. Implementacja Web zgodnie z [dokumentem Apple Pay on the Web](#)
3. Przygotowanie 2 końcówek po stronie Partnera, w domenie zgłoszonej w Apple (z wykorzystaniem 2 certyfikatów od Apple):
 - do rozpoczęcia sesji
 - do obciążenia Klienta na podstawie tokena od Apple

WSKAZÓWKA: safari (przeglądarka Klienta) puka o sesję do końcówki (o której mowa powyżej) oni wtedy trafiają do nas - my zwracamy sesję.

4. Przetwarzanie płatności Apple Pay

- W ramach rejestracji usługi w Apple wygenerować swój certyfikat *merchant identity*.
- Wygenerować certyfikat *payment processing* na bazie certyfikatu dostarczonego przez AP CSR

UWAGA: Certyfikaty AP CSR dla acceptu i dla produkcji różnią się).

- Po wykorzystaniu go w procesie rejestracji Apple dostarczyć AP certyfikat podpisany przez Apple i wysłać poprzez [formularz Autopay](#)

WSKAZÓWKA: Klient powinien podać kraj, miasto, domenę strony www, email osoby kontaktowej.

- W ramach realizacji płatności na stronie Partnera, rozpocząć sesję API Apple.
- Następnie zwrócić Autopay token w parametrze startowym PaymentToken.

UWAGA: Odszyfrowanie tokena to obowiązek AP.

Format payment tokena: wycinek obiektu w formacie json, który zwraca api ApplePay:

```
EncryptedPaymentData {
    String version;
    String data;
    String signature;
    Header header;
}
Header {
    String ephemeralPublicKey;
    String publicKeyHash;
    String transactionId;
    String applicationData;
}
```

UWAGA: Przy wysyłce `ApplePayPaymentRequest`, trzeba uzupełnić pole `applicationData` o wartość `orderId` zakodowaną Base64, zgodnie z opisem w [dokumencie applicationData](#).

Widget Autopay (model WhiteLabel)

Partner który chciałby osadzić część startów transakcji bezpośrednio w na swoim serwisie / w swoim

koszyku (w tzw. modelu WhiteLabel), może to zrobić integrując Widget Autopay. Aktualnie Widget Autopay wspiera zbieranie danych kartowych (w ramach PaywayId 1500/1503) oraz starty Visa Mobile (PaywayId 1523)

WAŻNE! Partner nie jest upoważniony do przechowywania danych Karty (w szczególności numeru karty, kodu zabezpieczającego CVC, CVV2), z wyjątkiem parametrów przekazywanych przy realizacji płatności automatycznych przez AP, opisanych w tej części.

WAŻNE! Serwis Partnera w którym wykorzystywana jest funkcjonalność Widget Autopay musi być szyfrowana a HTML IFRAME z widgetem musi być osadzony na adresie HTTPS z użyciem protokołu TLS.

WAŻNE! Partner zobowiązuje się przedłożyć AP, w formie elektronicznej, następujące dokumenty:

- a) jednorazowo (przed zawarciem Umowy): wypełniony kwestionariusz SAQ-A PCI (Section 2); Dokument zostanie dostarczony przez AP lub jest możliwy do pobrania na stronie: <https://www.pcisecuritystandards.org>
- b) kwartalnie: wynik kwartalnego audytu PCI ASV obejmującego skanowanie zewnętrznych (publicznych) adresów IP/sieci/ domen – protokołu IPv4 oraz/lub IPv6. Audyt taki musi zostać przeprowadzony przez jednego z autoryzowanych wykonawców, z listy dostępnej pod adresem: https://www.pcisecuritystandards.org/assessors_and_solutions/approved_scanning_vendors

Autopay WidgetJS SDK

Do osadzenia i komunikowania się z widget'em Autopay należy użyć Autopay WidgetJS SDK. W skrócie, sprowadza się to będzie do osadzenia HTML IFRAME z widget'em oraz skonfigurowania JS SDK w celu obsługi komunikatów (event'ów) produkowanych podczas interakcji Cardholder'a z widget'em. Komunikatem finalnym jest event ze statusem FORM_SUCCESS zawierający paymentToken niezbędny do backend'owego startu transakcji na API Systemu Płatności Online AP.

Osadzenie SDK

Poniżej przykłady, jak w prosty sposób, przy użyciu Autopay WidgetJS SDK, osadzić i skomunikować widget, zarówno dla kanałów kartowych jak i dla kanału VisaMobile.

Autopay WidgetJS SDK jest dostępny pod adresem `widget-new/widget-communication.min.js` po umieszczeniu go w sekcji `<head>`

```
<script
src="https://testcards.autopay.eu/widget-new/widget-communication.min.js"></script>
```

uzyskujemy dostęp do obiektu `WidgetConnection`

```
var widgetConfigObject = { ... };
var widget = new WidgetConnection(widgetConfigObject)
```

który po uzupełnieniu o konfigurację w formie obiektu JSON umożliwi pełną komunikację z API Autopay, a w rezultacie zapewni otrzymywanie event'u o statusie FORM_SUCCESS z paymentToken'em.

Przykłady konfiguracji

Przykład konfiguracji dla danych kartowych

Konfiguracja:

```
{ language: 'pl', amount: 1.23, currency: 'PLN', serviceId: 123456 }
```

Zwrócony PaymentToken:

```
{status: 'FORM_SUCCESS', message: 'ey...9', id: 'OGF1ZTYyYTMtN2U2OS00MTU1LTgyNDctNmMwMGI2NjE5ZDQy' }
```

Przykład konfiguracji dla danych VisaMobile

Konfiguracja:

```
{ language: 'pl', amount: 1.23, currency: 'PLN', serviceId: 123456, merchantName: 'ShopName' }
```

Zwrócony PaymentToken:

```
{status: 'FORM_SUCCESS', message: 'ey...9', prefix: '48', phoneNumber: '666666666', id: 'OGF1ZTYyYTMtN2U2OS00MTU1LTgyNDctNmMwMGI2NjE5ZDQy' }
```

Szczegółowe omówienie konfiguracji obiektu WidgetConnection

Język

Determinuje wersję językową widget'u w jakiej zostanie zaprezentowany.

Nazwa pola: language Format string Wartości: domyślnie pl, aktualnie wspierane są następujące języki: cs, de, eł, en, es, fr, hr, hu, it, pl, ro, se, sk, sl, uk

Kwota transakcji

Kwota transakcji

Nazwa pola: amount Format float Wartości: kwota zapisana w formacie float, czyli np: "1,23 PLN" to

1.23

amount: 1.23, currency: 'PLN', serviceId: 123456, merchantName: 'ShopName'

Waluta transakcji

Waluta transakcji

Nazwa pola: currency Format string Wartości: domyślnie PLN, inne waluty w formacie zgodnym z aktualną konfiguracją serwisu

Numer serwisu

Numer serwisu otrzymany od Autopay (zależny od środowiska developerskiego)

Nazwa pola: serviceId Format integer Wartości: zazwyczaj sześciocyfrowy

Typ rekurencji (tylko dla kart)

Wskazanie rodzaju inicjacji rekurencji (Tylko dla kanału 1503 związanego z inicjacją rekurencji)

Nazwa pola: recurringAction Format string Wartości: 'INIT_WITH_REFUND', 'INIT_WITH_PAYMENT'

Nazwa sklepu (tylko dla VisaMobile)

Nazwa wyświetlana użytkownikowi w powiadomieniu VisaMobile w aplikacji mobilnej banku (Tylko dla kanału 1523 VisaMobile)

Nazwa pola: merchantName Format string Wartości: Nazwa sklepu

Widget Kartowy - Przykład implementacji na stronie partnera

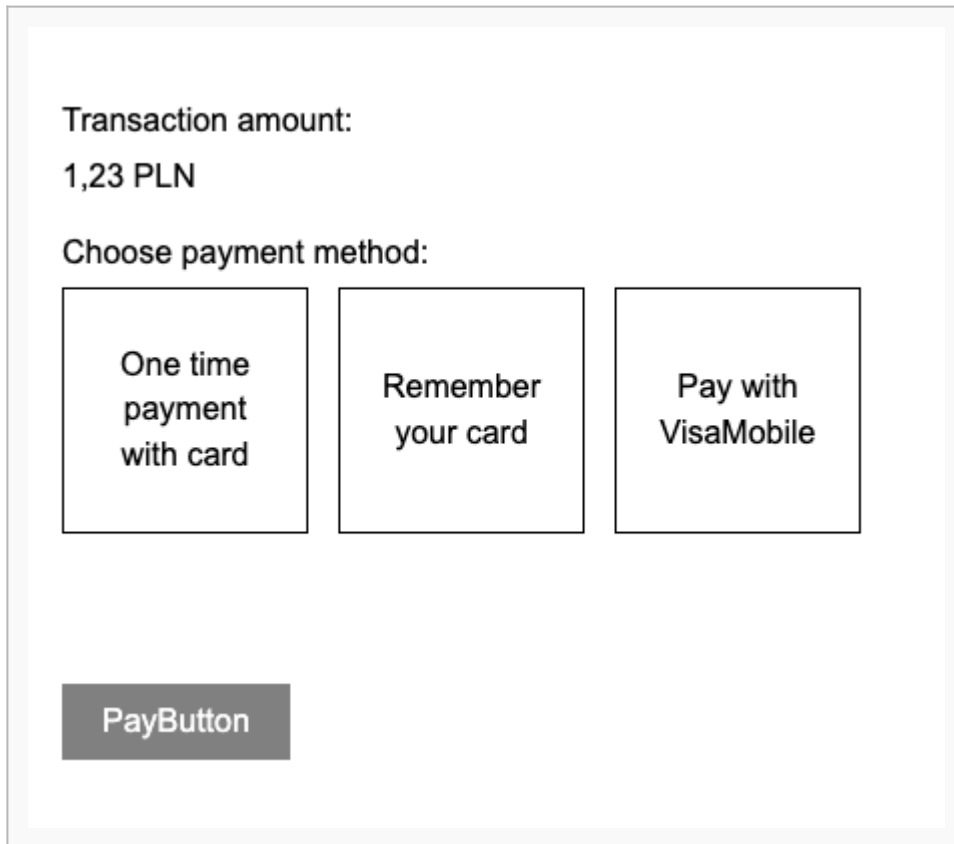
WAŻNE! [Poniższy przykład kodu HTML](#) powstał głównie w celach poglądowych. Aby go faktycznie uruchomić na swoim lokalnym komputerze, poniższy plik HTML musi zostać umieszczony pod jakąś lokalną domeną (dowolną, może być test.local). Ten HTML nie może być odpalany w przeglądarce w formie lokalnego pliku ponieważ eventyJS wymieniane pomiędzy IFRAME a stroną są weryfikowane pod kątem zgodności domeny (a więc jakaś domena musi być obecna).

Poniższa strona ma za zadanie imitować Front Merchanta, pokazując jakie elementy należy zaimplementować aby dokonać integracji z Widget'em Autopay.

W przeglądarce poniższa, przykładowa strona składa się z trzech sekcji:

- sekcja górna zawiera możliwość wyboru konkretnych kanałów płatności
- sekcja środkowa zawiera miejsce, w którym osadzony będzie HTML IFRAME, do którego, w razie potrzeby, umieszczany będzie adres widget'u (visamobile lub standardowego kartowego wedle potrzeby)

- sekcja dolna zawiera (domyślnie nieaktywny) przycisk PayButton spięty z SDK JS sterujący uruchomieniem procesu w widget'cie (w tym przykładzie przycisk uaktywnia się dopiero, gdy otrzymuje komunikat o poprawnej walidacji i uzyskaniu kompletu danych potrzebnych do wystartowania procesu)



Transaction amount:
1,23 PLN

Choose payment method:

One time payment with card	Remember your card	Pay with VisaMobile
----------------------------	--------------------	---------------------

PayButton

Kiedy zostanie wybrany kanał kartowy (payway: 1500 lub 1503), wczyta się dedykowany widok formatki kartowej (oparty o HTML IFRAME). W momencie wprowadzenia w widżecie pełnych, prawidłowych danych kartowych, (dzięki eventom walidacyjnym) nastąpi uaktywnienie się przycisku "Zapłać" na Froncie Merchanta.

Transaction amount:
1,23 PLN

Choose payment method:

One time
payment
with card

Remember
your card

Pay with
VisaMobile

3D SECURE VISA MasterCard SecureCode

Card payment is encrypted and secure. The transaction will be authorized using 3D Secure on your bank's website.

Your card data

Card number _____

First name _____

Last name _____ Valid to _____

CVV code _____

PayButton

Podpowiedź: Jak widać w przykładzie w modelu WhiteLabel jest również możliwa obsługa kanału VisaMobile. Implementacja/osadzenie są analogiczne do widgeta kartowego dlatego poniższy kod zawiera już oba przypadki.

Walidacja i skompletowanie danych

W momencie wpisywania danych karty Autopay Widget JS SDK otrzymuje od widget'u event'y VALIDITY_STATUS z wartością valid: false

Transaction amount:
1,23 PLN

Choose payment method:

One time
payment
with card

Remember
your card

Pay with
VisaMobile

3D SECURE VISA MasterCard
SecureCode

Card payment is encrypted and secure. The transaction will be authorized using 3D Secure on your bank's website.

Your card data

! Enter card number
Card number

! Complete first name
First name

! Complete last name ! Enter expiration date
Last name Valid to

! Enter CVV number
CVV code

PayButton

Gdy uzyskamy pełne dane karty, ostatnim event'em będzie VALIDITY_STATUS z wartością valid: true

```
{status: 'VALIDITY_STATUS', message: null, valid: true, id: 'M2Zl...mU2'}
```

O ten event można oprzeć uaktywnianie przycisku PayButton

Transaction amount:
1,23 PLN

Choose payment method:

One time payment with card

Remember your card

Pay with VisaMobile

3D SECURE **VISA** **MasterCard SecureCode**

Card payment is encrypted and secure. The transaction will be authorized using 3D Secure on your bank's website.

Your card data **VISA**

Card number
4444 4444 4444 4000 ✓

First name
Test ✓

Last name
Test ✓

Valid to
11 / 55 ✓

CVV code
555 ✓

PayButton

Podpowiedź: Na środowisku testowym płatności kartowe są oparte o mock 3ds i mock autoryzacyjny. Poszczególnym scenariuszom odpowiadają dedykowane numery kart testowych. Pełna lista przypadków testowych znajduje się w oddzielnym załączniku.

Ekran DCC

W przypadku gdy dany scenariusz i karta spełnia warunki uzyskania oferty DCC pojawi dodatkowy ekran z propozycją przewalutowania dla Cardholdera

Transaction amount:
1,23 PLN

Choose payment method:

One time payment with card	Remember your card	Pay with VisaMobile
----------------------------	--------------------	---------------------

Select the currency in which you want to pay.
The selection you made cannot be changed in the next step

EUR
You will pay 0,29 EUR
Current exchange rate: 1,00 PLN = 0,231853 EUR
Currency rate provider: MockApi - enum
The currency conversion fee is 4,00%

PLN
You will pay 1,23 PLN

[PayButton](#)

Cardholder może zdecydować się na skorzystanie z obciążenia karty w natywnej dla niej walucie lub pozostawić oryginalną walute. Na tym ekranie również występuje walidacja.

Transaction amount:
1,23 PLN

Choose payment method:

One time payment with card	Remember your card	Pay with VisaMobile
----------------------------	--------------------	---------------------

Select the currency in which you want to pay.
The selection you made cannot be changed in the next step

EUR
You will pay 0,29 EUR
Current exchange rate: 1,00 PLN = 0,231853 EUR
Currency rate provider: MockApi - enum
The currency conversion fee is 4,00%

PLN
You will pay 1,23 PLN

PayButton

Wybór waluty Cardholdera nie będzie miał wpływu na Merchanta i oryginalną kwotę samej transakcji, ale będzie miał wpływ na kwotę jaką zostanie obciążona karta. Jeśli Cardholder nie chce skorzystać z oferty przewalutowania DCC, zaznacza oryginalną walutę (czyli w tym przypadku PLN).

Transaction amount:
1,23 PLN

Choose payment method:

One time payment with card	Remember your card	Pay with VisaMobile
----------------------------	--------------------	---------------------

Select the currency in which you want to pay.
The selection you made cannot be changed in the next step

EUR
You will pay 0,29 EUR
Current exchange rate: 1,00 PLN = 0,231853 EUR
Currency rate provider: MockApi - enum
The currency conversion fee is 4,00%

PLN
You will pay 1,23 PLN

PayButton

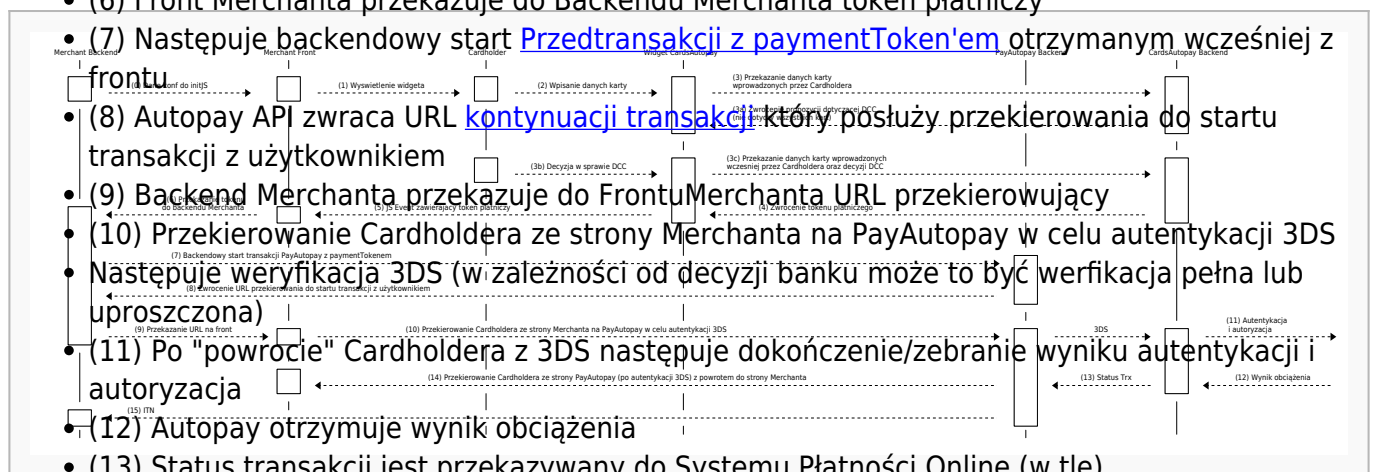
Uzyskanie tokenu

Przycisk należy powiązać z Autopay Widget JS SDK tak, aby jego kliknięcie triggerowało wywołanie metody `widget.sendForm()`; w obiekcie `WidgetConnection Co` finalnie, zaowocuje uzyskaniem event'u `FORM_SUCCESS`, czyli uzyskaniem wartości `paymentToken`'u (w polu `message`).

```
{status: 'FORM_SUCCESS', message: 'eyJ...n19', id: 'M2Z...ZmU2'}
```


Bezpieczne przekazania danych karty do systemu Autopay oraz pełny flow transakcji:

- (0) Przekazanie do frontu danych konfiguracyjnych inicjujących osadzenie widget'a
- (1) Wyświetlenie formatki kartowej Widgeta osadzonej na froncie Merchanta (dane karty nie są podawane na frontendzie merchanta tylko na frontendzie widget'a Autopay)
- (2) Start poprzez wpisanie przez Cardholdera danych karty
- (3) Przesłanie danych kartowych z użyciem połączenia TLS zabezpieczonego certyfikatem typu Extended Validation do backendu CardsAutopay ** Dotyczy tylko przypadku gdy jest możliwe zaproponowanie DCC ** (3a) Zwrócenie szczegółów propozycji przewalutowania DCC ** (3b) Cardholder podejmuje decyzję, czy chce skorzystać z DCC ** (3c) Następuje przekazanie danych karty wprowadzonych wcześniej przez Cardholdera oraz decyzji DCC
- (4) WidgetJS otrzymuje z CardsAutopay i przesyła do Frontu Merchanta (za pomocą JS) wartość paymentToken 'a
- (5) Front Merchanta poprzez Event JavaScript otrzymuje z Widget'a token płatniczy
- (6) Front Merchanta przekazuje do Backendu Merchanta token płatniczy



- (7) Następuje backendowy start [Przedtransakcji z paymentToken'em](#) otrzymanym wcześniej z frontu
- (8) Autopay API zwraca URL [kontynuacji transakcji](#) który posłuży przekierowania do startu transakcji z użytkownikiem
- (9) Backend Merchanta przekazuje do Frontu Merchanta URL przekierowujący
- (10) Przekierowanie Cardholdera ze strony Merchanta na PayAutopay w celu autentykacji 3DS
- Następuje weryfikacja 3DS (w zależności od decyzji banku może to być weryfikacja pełna lub uproszczona)
- (11) Po "powrocie" Cardholdera z 3DS następuje dokończenie/zebranie wyniku autentykacji i autoryzacja
- (12) Autopay otrzymuje wynik obciążenia
- (13) Status transakcji jest przekazywany do Systemu Płatności Online (w tle)
- (14) Następuje przekierowanie Cardholdera ze strony PayAutopay (po autentykacji 3DS) z powrotem [do strony Merchanta](#)
- (15) Asynchronicznie do Backendu Merchanta przychodzi komunikat [ITN](#) ze statusem transakcji ** (w przypadku transakcji inicjującej rekurencje, Backend Merchanta dostanie też dodatkowy komunikat [RPAN](#))

Widget Visa Mobile - Przykład implementacji na stronie partnera

Poniżej przykład prostej implementacji HTML/JS z użyciem widget'u VisaMobile (oraz Widget'u kartowego)


```
{ 'status': 'FORM_SUCCESS', 'message': 'eyJrZ...', ... }
```

Kluczowe dla tej integracji jest miejsce w kodzie JS, które odpowiada za odebranie event'ów, a szczególnie event'u o statusie FORM_SUCCESS, zawiera on bowiem w polu message wartość paymentToken'u, którą merchant musi przekazać do swojego backend'u w celu skompletowania parametrów do Autopay API umożliwiających start płatności w Autopay.

Przykładowa strona

W przeglądarce poniższa, przykładowa strona składa się z trzech sekcji:

- sekcja górna zawiera ikony/przyciski konkretnych kanałów płatności (z wykorzystaniem reprezentacji graficznych z Autopay)
- sekcja środkowa zawiera miejsce, w którym znajduje się HTML IFRAME, do którego, w razie potrzeby, umieszczany będzie adres widget'u (visamobile lub standardowego kartowego wedle potrzeby)
- sekcja dolna zawiera (domyślnie nieaktywny) przycisk PayButton spięty z SDK JS sterujący uruchomieniem procesu w widget'cie (w tym przykładzie przycisk uaktywnia się dopiero, gdy otrzymuje komunikat o poprawnej walidacji i uzyskaniu kompletu danych potrzebnych do wystartowania procesu)

Transaction amount:

1,23 PLN

Choose payment method:

One time
payment
with card

Remember
your card

Pay with
VisaMobile

PayButton

Kiedy zostanie wybrany kanał przeznaczony dla VisaMobile, wyświetla się dedykowany widok (oparty o HTML IFRAME), w którym podanie pełnego numeru telefonu (dzięki komunikatom walidacji) skutkuje uaktywnieniem przycisku "Zapłać".


Transaction amount:
1,23 PLN

Choose payment method:

One time
payment
with card

Remember
your card

Pay with
VisaMobile






Wprowadź numer telefonu ⓘ


+48

Nie będziemy wykorzystywać Twojego numeru telefonu w celach marketingowych.


ⓘ
Aby zapłacić Visa Mobile







Włącz usługę Visa Mobile w aplikacji banku dostępnego w Moje ING, Nest Bank oraz SGB Mobile





Jeśli posiadasz kartę Visa w jednym z poniższych banków, zarejestruj ją w aplikacji Visa Mobile




















Szybko, wygodnie, bezpiecznie
Po założeniu konta i dodaniu karty Visa, jedynie co musisz zrobić, to wpisać numer telefonu i potwierdzić płatność w aplikacji - bez potrzeby wprowadzania numeru karty i innych informacji przy każdej transakcji.

🔒 Bezpieczna płatność dzięki szyfrowaniu SSL

PayButton

Walidacja i skompletowanie danych

W momencie wpisywania numeru telefonu Autopay Widget JS SDK otrzymuje od widget'u event'y `VALIDITY_STATUS` z wartością `valid: false` Gdy uzyskamy pełen numer telefonu, ostatnim event'em będzie `VALIDITY_STATUS` z wartością `valid: true`

```
{status: 'VALIDITY_STATUS', message: null, valid: true, id: 'M2ZL...mU2'}
```

O ten event można oprzeć uaktywnianie przycisku PayButton


Transaction amount:
1,23 PLN

Choose payment method:


One time payment with card

Remember your card

Pay with VisaMobile



Wprowadź numer telefonu ⁱ

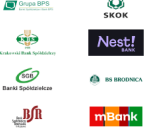
+48 666 666 666 

Nie będziemy wykorzystywać Twojego numeru telefonu w celach marketingowych.


Aby zapłacić Visa Mobile


Włącz usługę Visa Mobile w aplikacji banku dostępną w Moje ING, Nestl Bank oraz SGB Mobile


Jeśli posiadasz kartę Visa w jednym z poniższych banków, zarejestruj ją w aplikacji Visa Mobile



Szybko, wygodnie, bezpiecznie
Po założeniu konta i dostaniu karty Visa, jedyną co musisz zrobić, to wpisać numer telefonu i potwierdzić płatność w aplikacji - bez potrzeby wpisywania numeru karty i innych informacji przy każdej transakcji.

 Bezpieczna płatność dzięki szyfrowaniu SSL

PayButton 



Uzyskanie tokenu

Przycisk należy powiązać z Autopay Widget JS SDK tak, aby jego kliknięcie triggerowało wywołanie metody `widget.sendForm()`; w obiekcie `WidgetConnection Co` finalnie, zaowocuje uzyskaniem event'u `FORM_SUCCESS`, czyli uzyskaniem wartości `paymentToken`'u.

```
{status: 'FORM_SUCCESS', message: 'eyJ...n19', prefix: '48', phoneNumber: '666666666', id: 'M2Z...ZmU2'}
```


Transaction amount:
1,23 PLN

Choose payment method:

One time payment with card

Remember your card

Pay with VisaMobile



Wprowadź numer telefonu ⁱ




+48 666 666 666



Nie będziemy wykorzystywać Twojego numeru telefonu w celach marketingowych.


Aby zapłacić Visa Mobile

Wiąca usługa Visa Mobile w aplikacji banku dostępne w Mop, ING, Ivoo Bank oraz SGB Mobile

Jeśli posiadasz kartę Visa w jednym z poniższych banków, zarejestruj ją w aplikacji Visa Mobile.



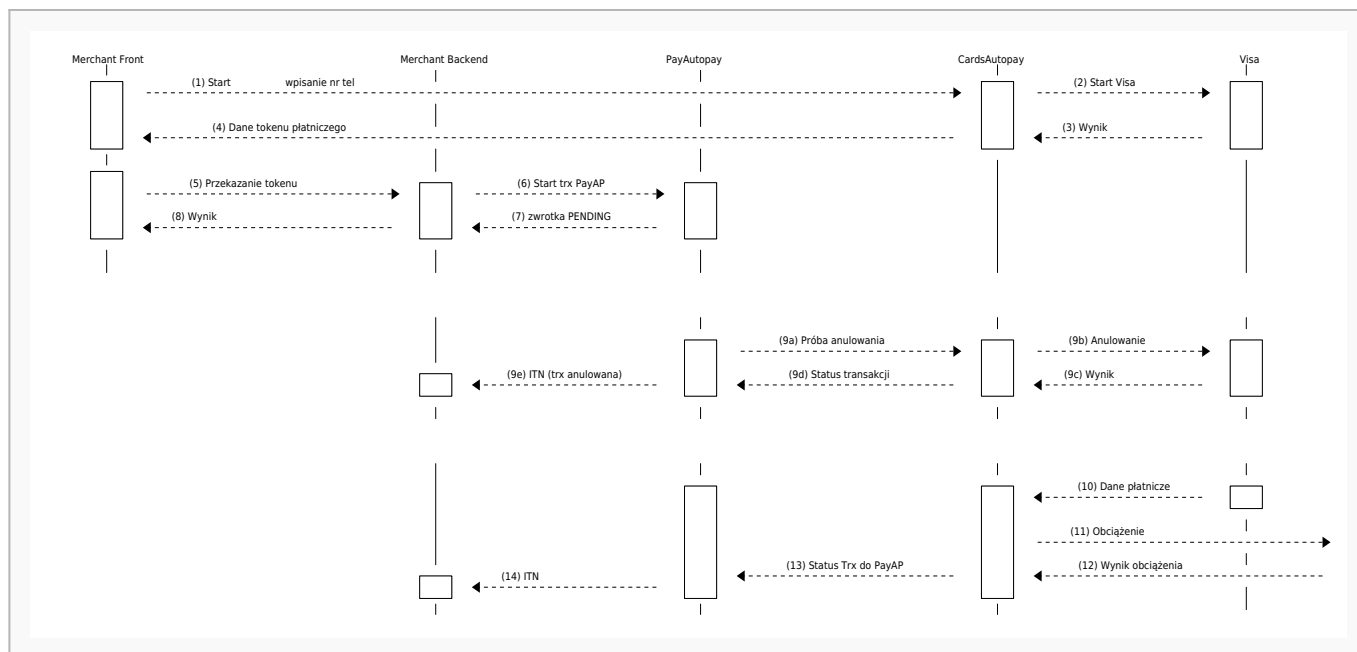
Szybko, wygodnie, bezpiecznie

Po założeniu konta i dołączeniu karty Visa, jedynie cała kwota przebieg, to wpisać numer telefonu i potwierdzić płatność w aplikacji - bez potrzeby wyjawienia numeru karty i innych informacji przy każdej transakcji.

Bezpieczna płatność dzięki szyfrowaniu SSL

PayButton

Widget Visa Mobile - Szczegółowy schemat komunikacji i wymiany danych



Start transakcji:

- (1) Start poprzez wpisanie numeru telefonu w widget'cie VisaMobile
- (2) Wystartowanie transakcji w systemie VisaMobile
- (3) Zwrotka z VisaMobile
- (4) Zwrócenie z widget'u do Merchanta (za pomocą JS) wygenerowanego paymentToken 'a
- (5) Front Merchanta przekazuje do Backendu Merchanta token płatniczy
- (6) Następuje backendowy start [Przedtransakcji z paymentToken'em](#) otrzymanym wcześniej paymentToken'em
- (7) Zwrotka dostaje od razu odpowiedź XML PENDING (ponieważ jest procesowana w tle)
- (8) Na froncie Merchanta zaprezentowane zostaje ekran oczekiwania na wynik

Ewentualne anulowanie transakcji (z poziomu paywall'a PayAutopay):

- (9a) Jeśli użytkownik nie dostanie powiadomienia w aplikacji mobilnej lub zmieni zdanie ma możliwość anulowania transakcji z poziomu paywall'a
- (9b) Z systemu CardsAutopay wysyłany jest request anulujący do Visa
- (9c) System CardsAutopay otrzymuje wynik anulowania z Visa
- (9d) Wynik anulowanie zostaje przyjęty przez system PayAutopay oraz zaprezentowany użytkownikowi na paywall'u PayAutopay
- (9e) Równoległe jest wysyłany [ITN](#) (ze statusem negatywnym) do Merchanta

Obciążenie i zwrócenie wyniku:

- (10) Oczekiwanie na dane tokenu płatniczego z Visa (jeśli klient VisaMobile potwierdzi w aplikacji mobilnej chęć zapłacenia konkretną kartą za dane zamówienie)
- (11) Następuje zlecenie obciążenia
- (12) Otrzymujemy pozytywny lub negatywny wynik obciążenia
- (13) Następuje przesłanie stanu transakcji do bramki PayAutopay
- (14) Bramka Autopay wysyła wynik transakcji do Merchanta w formie [ITN'a](#)

Omówienie przykładowego kodu HTML JS (Widget Kartowy i VisaMobile)

Poniższy kod został wykorzystany do wygenerowania przykładowych integracji, o których była mowa wyżej w sekcjach z przykładami implementacji Widget'a Kartowego jak i Widget'a Visa Mobile

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Autopay Widget Integration Example</title>
  <script
src="r=quot;https://testcards.autopay.eu/widget-new/widget-communication.min.js"></scri
pt>
  <style>/* pominięte w przykładzie */</style>
</head>
<body>
<div>
  <form onsubmit="submitForm(event)" novalidate>
    <div class="form-group"><p>Transaction amount:</p><span>1,23 PLN</span></div>

    <!-- przykładowa implementacja mechanizmu wyboru kanału płatności po stronie
merchanta -->
    <p>Choose payment method:</p>
    <ul>
      <li onclick="setPayway(event, 1500)">One time payment with card</li>
      <li onclick="setPayway(event, 1503)">Remember your card</li>
      <li onclick="setPayway(event, 1523)">Pay with VisaMobile</li>
    </ul>

    <!-- miejsce, w które wstrzyknięty zostanie HTML IFRAME z widget'em -->
    <div class="form-group" id="iframe-wrapper">
      <iframe id="iframe"></iframe>
    </div>

    <!-- przycisk wywołujący akcję w widget'cie -->
    <button type="submit" id="button" disabled="disabled">PayButton</button>
  </form>
</div>
<script type="text/javascript">
window.addEventListener('load', () => {
  // pomocnicze zmienne (tylko na potrzeby przykładu)
  var currentPayway = null;
  var widget = null;

  // przykładowe konfiguracje zależne od środowiska developerskiego (tylko na potrzeby
przykładu)
  var AUTOPAY_CARDS_DOMAIN_ENV_PROD = 'https://cards.autopay.eu';
  var AUTOPAY_CARDS_DOMAIN_ENV_TEST = 'https://testcards.autopay.eu';

  var MERCHANTS_SERVICE_ID_ENV_PROD = 903555;
  var MERCHANTS_SERVICE_ID_ENV_TEST = 903555;

  // pomocnicza metoda obsługująca wybór kanału płatności i osadzenie widget'u
function setPayway (event, paywayId) {
  if (currentPayway === paywayId) {
    return;
  }
  currentPayway = paywayId
  removeWidget();
  disableSubmitButton();
  markActiveIcon(event);

  if (paywayId === 1500) {
```



```

        // obsługa event'u związanego z walidacją w momencie wprowadzania danych
w widget'cie przez użytkownika/cardholder'a
        widget.on(widgetEvents.validationResult, function (message, eventData) {
            console.log('form validation result =>', eventData);
            if (eventData.valid) {
                enableSubmitButton();
            } else {
                disableSubmitButton();
            }
        })

        // obsługa event'u showModal
        widget.on(widgetEvents.showModal, function () {
            console.log('show modal');
        })
    })
}

// pomocnicza metoda (tylko na potrzeby przykładu) ustawiająca usuwanie widget'u dla
innych, niż kartowe, kanałów płatności (w przykładzie jest kanał 106 PBL)
function removeWidget () {
    if (!widget) {
        return;
    }
    widget.stopConnection();
}

// pomocnicza metoda (tylko na potrzeby przykładu)
function enableSubmitButton () {
    document.getElementById('button').removeAttribute('disabled');
}

// pomocnicza metoda (tylko na potrzeby przykładu)
function disableSubmitButton () {
    document.getElementById('button').setAttribute('disabled', 'disabled');
}

// pomocnicza metoda (tylko na potrzeby przykładu) powiązująca naćciśnięcie aktywnego
przycisku zapłacić z wywołaniem sendForm() w obiekcie widget'u
function submitForm (event) {
    event.preventDefault();
    if (!widget || widget.invalid) {
        return;
    }
    disableSubmitButton();
    widget.sendForm();
}

window.setPayway = setPayway;
window.submitForm = submitForm
});
</script>
</body>
</html>

```

Płatność kartą osadzoną w Serwisie (iFrame)

iFrame kartowy (PayBmCheckout) nie jest już wspierany, zamiast niego do płatności kartą osadzoną w

serwisie Partnera należy użyć [Widgeta Kartowego](#).

Płatność automatyczna

Opis płatności automatycznej

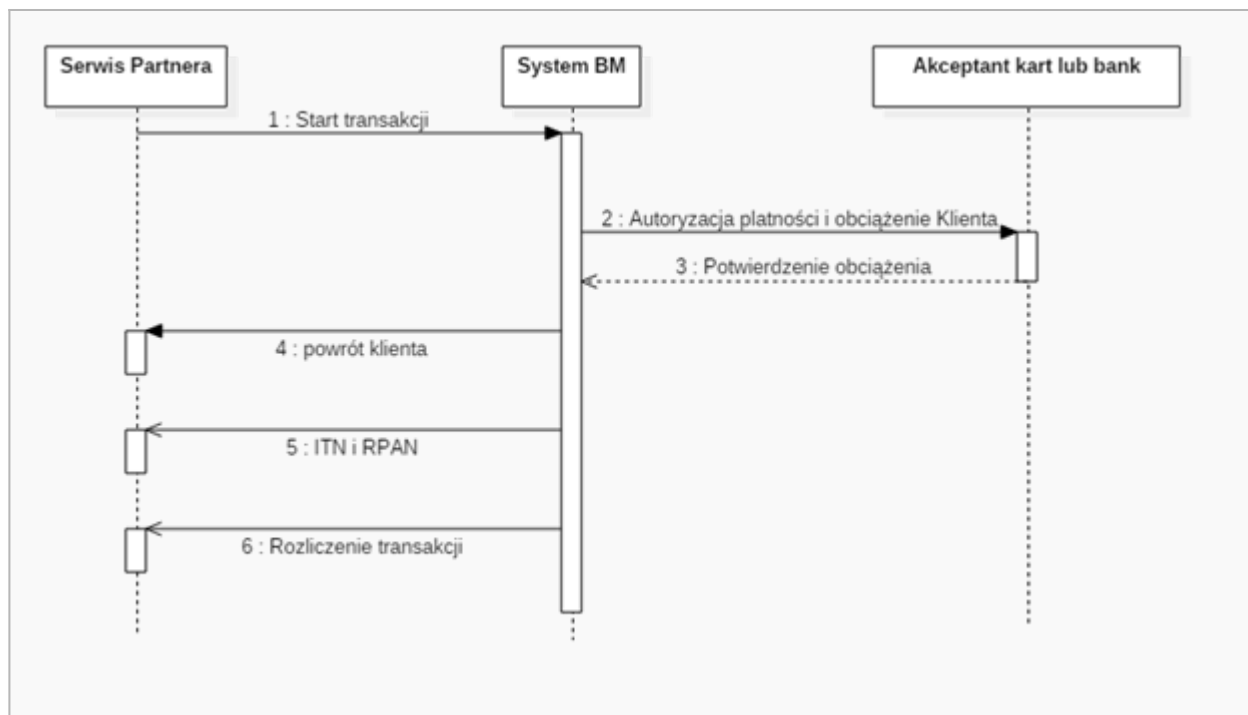
Płatności automatyczne to wyjątkowo wygodny i bezpieczny sposób dokonywania powtarzalnych transakcji. Polega ona na automatycznym pobieraniu należności od Klienta w jej terminach płatności. Usługę należy najpierw aktywować. W przypadku kart odbywa się to poprzez przekierowanie klienta do formatki aktywacyjnej usługi. W przypadku BLIK natomiast poprzez akceptację płatności automatycznej w Aplikacji Mobilnej. Po skutecznym zautoryzowaniu takiej transakcji aktywacyjnej, AP przekazuje do Partnera standardowy komunikat o zmianie statusu transakcji (ITN) oraz komunikat o uruchomieniu usługi płatności automatycznej (RPAN). Komunikat RPAN zawiera pole **clientHash**, którym Partner będzie identyfikować konkretną płatność automatyczną podczas późniejszych obciążeń oraz dezaktywacji usługi.

Wszystkie transakcje w ramach cyklu życia płatności automatycznej (aktywacja i obciążenia) są realizowane w ramach dedykowanych Kanałów Płatności (BLIK - **GatewayID=522**, Karty - **GatewayID = 1503**) o **gatewayType="Płatność automatyczna"**. W przypadku integrowania płatności automatycznej BLIK, można podać (w danych podawanych przed integracją) długość życia aktywowanych płatności automatycznych (domyślnie bezterminowe) lub podawać ją w transakcji inicjalizującej (parametr **RecurringValidityTime**).

Aktywacja płatności automatycznej

Aktywacja płatności automatycznej składa się z autoryzacji transakcji aktywacyjnej, komunikacji ITN oraz RPAN. Po otrzymaniu RPAN, Partner jest gotowy do wykonywania obciążeń cyklicznych (lub jednym kliknięciem).

Jest to przypadek aktywacji usługi płatności automatycznej podczas płatności za usługę/towar (a więc **RecurringAction=INIT_WITH_PAYMENT** i rozliczenie transakcji do Partnera).



Proces aktywacji usługi płatności automatycznych

Komunikat ITN wysyłany po płatności automatycznej jest podobny do tych, otrzymywanych po płatnościach jednorazowych (rozszerzony jest jedynie o węzeł **RecurringData**, oraz - dla płatności kartowej - **CardData**). Pozostałe dwa elementy procesu aktywacji usługi to start transakcji oraz RPAN.

Komunikat startu transakcji automatycznej

Proces aktywacji usługi inicjowany jest z Serwisu Partnera, poprzez rozpoczęcie transakcji z parametrem **RecurringAction** pozwala sterować zachowaniem Systemu:

- a) wartość **INIT_WITH_PAYMENT** - odpowiada aktywacji usługi płatności automatycznej podczas płatności za usługę/towar (karta lub rachunek obciążane są kwotą należności, a środki z płatności przekazywane są do Partnera); na liście dostępnych kanałów płatności System prezentuje tylko płatności automatyczne (o ile nie wybrano kanału płatności w Serwisie),
- b) wartość **INIT_WITH_REFUND** - odpowiada aktywacji usługi płatności automatycznej poza procesem płatności za usługę/towar (karta lub rachunek obciążane są kwotą 1 PLN, po czym następuje automatyczny zwrot środków na rachunek Klienta); na liście dostępnych kanałów płatności System prezentuje tylko płatności automatyczne (o ile nie wybrano kanału płatności w Serwisie),
- c) brak parametru (lub pusty) - o ile nie wybrano kanału płatności w Serwisie, System wyświetli wszystkie dostępne dla Serwisu kanały płatności (wraz z automatycznymi) oraz pozostawi Klientowi decyzję: płatność jednorazowa, czy uruchomienie płatności automatycznej. Jeśli Klient wybierze płatność automatyczną, to transakcja zostanie w standardowy sposób rozliczona na rzecz Partnera (a w RPAN wróci parametr **RecurringAction=INIT_WITH_PAYMENT**).

UWAGA: Niedozwolone jest rozpoczynanie transakcji aktywacyjnych z wybranym kanałem płatności automatycznej, ale bez wybranego **RecurringAction**.

W niektórych przypadkach (jeśli wynika to z odpowiedzi metody *legalData*) wymagane jest również podanie parametrów *RecurringAcceptanceState* (o wartości **ACCEPTED**, co oznacza to, że Klient przeczytał i zaakceptował regulamin płatności automatycznej na Serwisie Partnera) oraz *RecurringAcceptanceID*.

Aktywacja usługi kartowej płatności automatycznej odbywa się na formatkach dostarczanych przez AP. Klient jest zobowiązany do podania danych karty: Imię, Nazwisko, Numer karty, Datę ważności oraz kod CVV. W przypadku automatycznej płatności z rachunku bankowego (BLIK), autoryzacja następuje bez podawania danych kartowych: np. kodem BLIK (transportowanym w parametrach startowych w *AuthorizationCode*), lub poprzez Alias BLIK OneClick (transportowanym w parametrach startowych w *BlikUIDKey/BlikUIDLabel*).

Po zautoryzowaniu transakcji, System AP przekazuje do Serwisu Partnera komunikat o zmianie statusu transakcji (ITN) oraz komunikat o uruchomieniu usługi płatności automatycznej (RPAN). Komunikat RPAN jest dedykowany dla zdarzeń aktywacji płatności automatycznej i zawiera jej identyfikator (*ClientHash*), którym Partner będzie się posługiwać podczas późniejszych obciążeń oraz dezaktywacji usługi. RPAN zawiera też informację o akcji w procesie płatności automatycznej (*RecurringAction*, opisany wyżej).

Powiadomienie o uruchomieniu płatności automatycznej (RPAN)

Po otrzymaniu pozytywnego statusu płatności dla aktywacji płatności automatycznej, wysyłany jest do Serwisu dedykowany komunikat. Powiadomienie to polega na wysłaniu przez System AP dokumentu XML zawierającego dane o uruchomionej płatności automatycznej. Dokument wysyłany jest protokołem HTTPS (domyślnie port 443), metodą POST, jako parametr HTTP o nazwie *recurring*. Parametr ten jest zapisany mechanizmem kodowania transportowego Base64.

Format dokumentu (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
  <recurringActivation>
    <serviceID>ServiceID</serviceID>
    <transaction>
      <orderID>OrderID</orderID>
      <remoteID>RemoteID</remoteID>
      <amount>999999.99</amount>
      <currency>PLN</currency>
      <gatewayID>GatewayID</gatewayID>
      <paymentDate>YYYYMMDDhhmmss</paymentDate>
      <paymentStatus>PaymentStatus</paymentStatus>
    <paymentStatusDetails>PaymentStatusDetails</paymentStatusDetails>
      <startAmount>999998.99</startAmount>
      <invoiceNumber>InvoiceNumber</invoiceNumber>
      <customerNumber>CustomerNumber</customerNumber>
      <customerEmail>CustomerEmail</customerEmail>
      <customerPhone>CustomerPhone</customerPhone>
    </transaction>
    <recurringData>
      <recurringAction>RecurringAction</recurringAction>
      <clientHash>ClientHash</clientHash>
      <expirationDate>YYYYMMDDhhmmss</expirationDate>
    </recurringData>
    <cardData>
      <index>Index</index>
      <validityYear>ValidityYear</validityYear>
```

```

        <validityMonth>ValidityMonth</validityMonth>
        <issuer>Issuer</issuer>
        <bin>BIN</bin>
        <mask>Mask</mask>
    </cardData>
    <hash>Hash</hash>
</recurringActivation>

```

Wartości elementów: **orderID**, **serviceID**, **amount** dotyczące każdej z aktywowanych płatności automatycznych są identyczne z wartościami odpowiadających im pól podanymi przez Serwis przy rozpoczęciu danej płatności inicjalizacyjnej.

Opis zwracanych parametrów dla uruchomienia płatności automatycznej

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1.	serviceID	TAK	string{1,10}	Identyfikator Serwisu Partnera, nadawany w trakcie rejestracji usługi, jednoznacznie identyfikuje Serwis Partnera w Systemie płatności online.
2.	transaction -> orderID	TAK	string{1,32}	Identyfikator transakcji nadany w Serwisie Partnera i przekazany w starcie transakcji.
3.	transaction -> remoteID	TAK	string{1,20}	Alfanumeryczny identyfikator transakcji nadany przez System płatności online.

kolejność Hash	nazwa	wymagany	typ	opis
5.	transaction -> amount	TAK	amount	<p>Kwota transakcji. Jako separator dziesiętny używana jest kropka - '.'</p> <p>Format: 0.00; maksymalna długość: 14 cyfr przed kropką i 2 po kropce. UWAGA:</p> <p>Dopuszczalna wartość pojedynczej Transakcji w Systemie produkcyjnym wynosi odpowiednio:</p> <ul style="list-style-type: none"> - dla PBL - min. 0.01 PLN, max. 100000.00 PLN (lub do wysokości ustalonej przez Bank wydający instrument płatniczy) - dla Kart płatniczych - min. 0.10 PLN, max. 100000.00 PLN (lub do wysokości indywidualnego limitu pojedynczej transakcji w Banku wydawcy Karty) - dla Szybkich przelewów - min. 0.01 PLN, max. 100000.00 PLN (lub do wysokości indywidualnego limitu pojedynczej transakcji w Banku dla przelewu wewnątrzbankowego) - dla BLIK - min. 0.01 PLN, max. 75000.00 PLN (lub do wysokości indywidualnego limitu pojedynczej transakcji w Banku dla przelewu wewnątrzbankowego) - dla OTP - min. 100.00 PLN, max. 2000.00 PLN - dla Alior Rat - min. 50.00 PLN, max. 7750.00 PLN
6.	transaction -> currency	TAK	string{1,3}	Waluta transakcji.
7.	transaction -> gatewayID	TAK	string{1,5}	Identyfikator Kanału Płatności, za pomocą, którego klient uregulował płatność.
8.	transaction -> paymentDate	TAK	string{14}	Moment zautoryzowania transakcji, przekazywany w formacie YYYYMMDDhhmmss. (Czas CET)

kolejność Hash	nazwa	wymagany	typ	opis
9.	transaction -> paymentStatus	TAK	enum	Status autoryzacji transakcji. Przyjmuje wartości (przejścia statusów identyczne z analogicznym polem w ITN): PENDING - transakcja rozpoczęta SUCCESS - poprawna autoryzacja transakcji, Serwis otrzyma środki za transakcję FAILURE - transakcja nie została zakończona poprawnie
10.	transaction -> paymentStatusDetails	TAK	enum	Szczegółowy status transakcji, wartość może być ignorowana przez Serwis.
11.	transaction -> startAmount	NIE	amount	Kwota transakcji podana w Linku Płatności (nie uwzględnia ew. kwoty prowizji naliczonej Klientowi). Suma prowizji Klienta i startAmount znajduje się w polu amount, gdyż jest to wynikowa wartość transakcji). Jako separator dziesiętny używana jest kropka - '.' Format: 0.00; maksymalna długość: 14 cyfr przed kropką i 2 po kropce.
12.	transaction -> invoiceNumber	NIE	string{1,100}	Numer dokumentu finansowego w serwisie.
13.	transaction -> customerNumber	NIE	string{1,35}	Numer klienta w serwisie.
14.	transaction -> customerEmail	NIE	string{1,60}	Adres email klienta.
15.	transaction -> customerPhone	NIE	string{9-15}	Numer telefonu użytkownika.
16.	recurringData -> recurringAction	NIE	string{1,100}	Akcja w procesie płatności automatycznej.
17.	recurringData -> clientHash	TAK	string{1,64}	Identyfikator płatności automatycznej.
18.	recurringData -> expirationDate	NIE	string{14}	Moment wygaśnięcia ważności płatności automatycznej, przekazywany w formacie YYYYMMDDhhmmss. (Czas CET)

kolejność Hash	nazwa	wymagany	typ	opis
19.	cardData -> index	NIE	string{1, 64}	Index karty płatniczej używanej w płatności automatycznej (jeśli użyto karty).
20.	cardData -> validityYear	NIE	string{4}	Ważność karty w formacie YYYY (jeśli użyto karty).
21.	cardData -> validityMonth	NIE	string{2}	Ważność karty w formacie mm (jeśli użyto karty).
22.	cardData -> issuer	NIE	string{64}	Wystawca karty, możliwe wartości: - VISA - MASTERCARD - MAESTRO - AMERICAN EXPRESS (obecnie nie wspierane) - DISCOVER (obecnie nie wspierane) - DINERS (obecnie nie wspierane) - UNCATEGORIZED (nierozpoznany wystawca)
23.	cardData -> bin	NIE	string{6}	Pierwsze 6 cyfr numeru karty.
24.	cardData -> mask	NIE	string{4}	Ostatnie 4 cyfry numeru karty.
nd.	hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis.

WSKAZÓWKA: Element **hash** (komunikatu) służy do autentykacji dokumentu. Wartość tego elementu obliczana jest jako wartość funkcji skrótu z łańcucha zawierającego sklejone wartości wszystkich pól dokumentu oraz dołączonego klucza współdzielonego. Opis sposobu obliczania skrótu znajduje się w części [Bezpieczeństwo transakcji](#).

Odpowiedź na powiadomienie

W odpowiedzi na powiadomienie oczekiwany jest status HTTP 200 (OK) oraz tekst w formacie XML (nie kodowany Base64), zwracany przez Serwis Partnera w tej samej sesji HTTP, zawierający potwierdzenie otrzymania komunikatu.

Struktura potwierdzenia (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
  <confirmationList>
```



```

<serviceID>ServiceID</serviceID>
<recurringConfirmations>
  <recurringConfirmed>
    <clientHash>ClientHash</clientHash>
    <confirmation>Confirmation</confirmation>
  </recurringConfirmed>
</recurringConfirmations>
<hash>Hash</hash>
</confirmationList>

```

Element confirmation płatności automatycznej

Element **confirmation** służy do przekazania stanu weryfikacji autentyczności transakcji przez Serwis Partnera. Wartość elementu wyznaczana jest przez sprawdzenie poprawności wartości parametru **serviceID**, porównanie wartości pól **orderID** i **amount** w komunikacie powiadomienia oraz w komunikacie rozpoczynającym transakcję, a także weryfikację zgodności wyliczonego skrótu z parametrami komunikatu z wartością przekazaną w polu hash komunikatu.

Przewidziano dwie wartości elementu **confirmation**:

- a) **CONFIRMED** – wartości parametrów w obu komunikatach oraz parametr hash są zgodne – transakcja autentyczna;
- b) **NOTCONFIRMED** – wartości w obu komunikatach są różne lub niezgodność hash – transakcja nieautentyczna;

WSKAZÓWKA: Element **hash** (w odpowiedzi na komunikat) służy do autentykacji odpowiedzi i liczony jest z wartości parametrów odpowiedzi. Opis sposobu obliczania skrótu znajduje się w części [Bezpieczeństwo transakcji](#).

W wypadku braku poprawnej odpowiedzi na wysłane powiadomienia, System płatności online podejmie kolejne próby jego przekazania po upływie określonego czasu. Serwis Partnera powinien wykonywać własną logikę biznesową (np. uruchomienie usługi płatności automatycznej, mailingu itp.) jedynie po pierwszym komunikacie o danym **ClientHash**.

Schemat ponawiania komunikatów ITN/ISTN/IPN/RPAN/RPDN

Poniżej schemat opisujący planowe ponawianie komunikatów (zastrzegamy jednak możliwość ponowienia każdego z nich w dowolnym momencie).

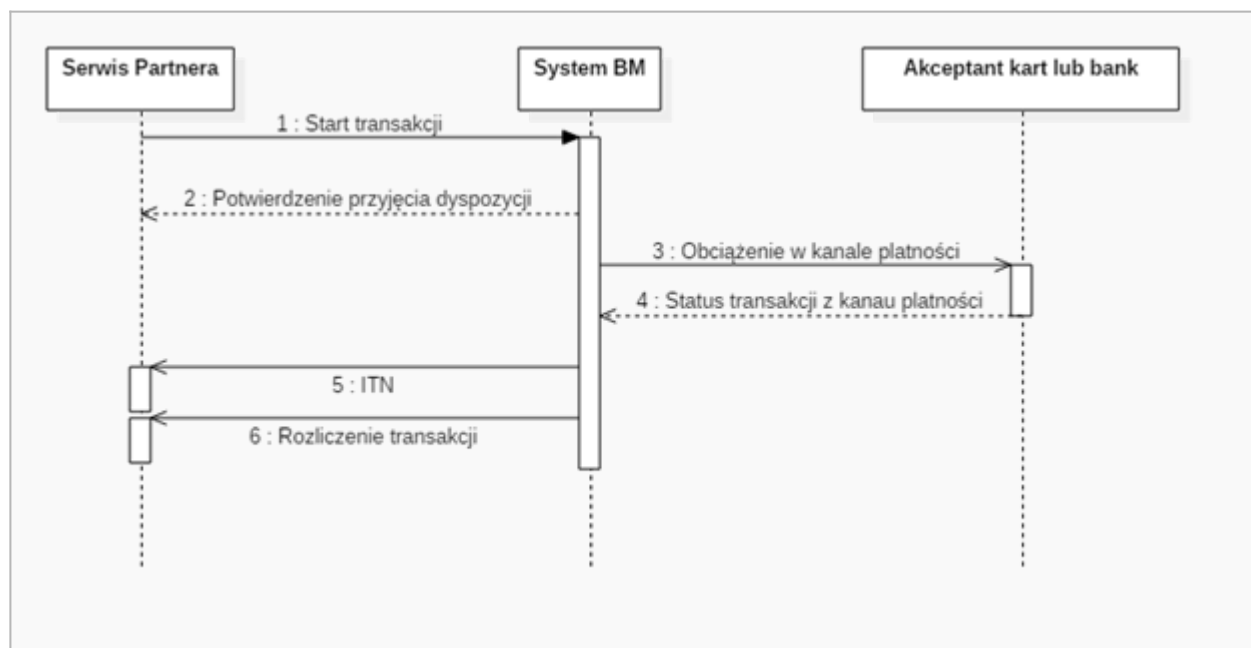
Nr ponowienia	Odstęp do kolejnego ponowienia
1-12	3 min
13-156	10 min
157-204	1 godzina
205-209	1 dzień

UWAGA: Ciągłe ponawianie przez System identycznego komunikatu oznacza brak lub nieprawidłową na niego odpowiedź z Serwisu, oraz wymaga od Partnera pilnej diagnozy przyczyny.

Obciążenie dla płatności automatycznej

Poprawne odebranie identyfikatora usługi (**ClientHash**), sprawia, że Partner jest gotowy do automatycznego obciążania Klienta za towary/usługi zakupione w Serwisie. Proces składa się z transakcji oraz komunikacji ITN.

Poniżej proces automatycznego obciążenia Klienta za usługę/towar (a więc **RecurringAction=MANUAL/AUTO** i rozliczenie transakcji do Partnera).



Komunikat ITN wysyłany po płatności automatycznej jest podobny do tych, otrzymywanych po płatnościach jednorazowych. Rozszerzony jest jedynie o węzeł **RecurringData** oraz (dla płatności kartowej) **CardData**.

Komunikat startu transakcji płatności automatycznej

Aby wykonać automatyczne obciążenie, Serwis Partnera powinien wykonać Przedtransakcję z parametrem **ClientHash**, zgodnym z aktywowaną wcześniej usługą płatności automatycznej (pochodzące z RPAN), z parametrem **RecurringAcceptanceState** o wartości **NOT_APPLICABLE** oraz odpowiednią wartość parametru **RecurringAction**:

- a) **AUTO** - płatność cykliczna (obciążenie bez udziału Klienta),
- b) **MANUAL** - płatność jednym kliknięciem (obciążenie zlecane przez Klienta, zwane OneClick).

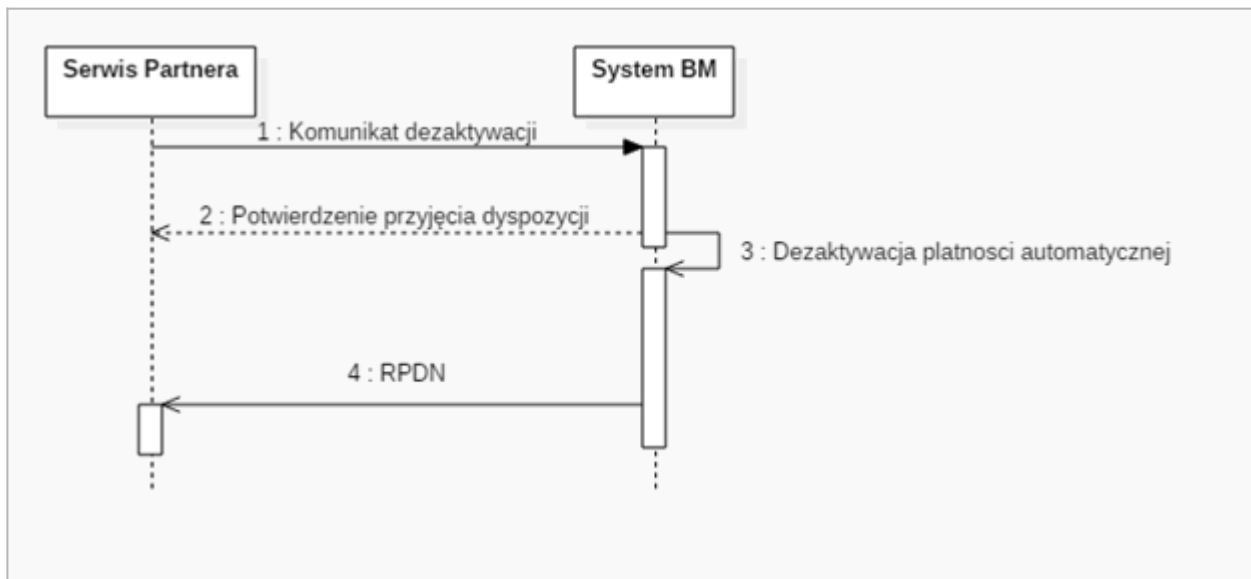
UWAGA: Udział klienta w opcji **MANUAL**, zazwyczaj ogranicza się do wywołania komunikatu (wybranie w Serwisie opcji zapłaty zapamiętaną kartą). W zdecydowanej większości przypadków wymagana jest dodatkowa autoryzacja w banku (w postaci 3DS lub kodu CVC). Wtedy zamiast

obciążenia (i statusu zlecenia w odpowiedzi na przedtransakcję), System zwróci link do kontynuacji – takie jest domyślne zachowanie systemu na środowisku testowym. Aby przetestować scenariusz obciążenia bez potrzeby dodatkowej autoryzacji należy zgłosić potrzebę zmiany konfiguracji Systemu na czas testu.

UWAGA: Opcja niedostępna dla płatności automatycznych BLIK (BLIK OneClick).

Dezaktywacja usługi

Partner może dezaktywować usługę płatności automatycznych w dowolnym momencie. Proces może składać się z komunikatu zlecającego dezaktywację oraz komunikatu RPDN (dedykowanego dla zdarzeń rezygnacji z usługi płatności automatycznej).



Może się również zdarzyć, że rezygnacja z usługi zostanie zainicjowana ze strony AP (np. na wniosek Klienta, banku lub organizacji kartowej). W takiej sytuacji System również dostarczy komunikat RPDN.

Komunikat dezaktywacji płatności automatycznej

Serwis może wyłączyć usługę poprzez dedykowany komunikat. Wszystkie parametry przekazywane są metodą POST (na adres https://{host_bramki}/deactivate_recurring). Protokół rozróżnia wielkość liter zarówno w nazwach jak i wartościach parametrów. Wartości przekazywanych parametrów powinny być kodowane w UTF-8.

Lista parametrów dezaktywacji płatności automatycznej

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	ServiceID	TAK	string{1,10}	Identyfikator Serwisu Partnera, nadawany w trakcie rejestracji usługi, jednoznacznie identyfikuje Serwis Partnera w Systemie płatności online.

kolejność Hash	nazwa	wymagany	typ	opis
2	MessageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łacińskiego (np. na bazie UID), wartość pola musi być unikalna dla Serwisu Partnera.
3	ClientHash	TAK	string{1,64}	Identyfikator płatności automatycznej.
nd.	Hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis.

UWAGA: Element Hash (komunikatu) służy do autentykacji dokumentu. Wartość tego elementu obliczana jest jako wartość funkcji skrótu z łańcucha zawierającego sklejone wartości wszystkich pól dokumentu oraz dołączonego klucza współdzielonego.

Odowiedź

W odpowiedzi na powiadomienie zwracany jest tekst w formacie XML w tej samej sesji HTTP, zawierający potwierdzenie.

Struktura potwierdzenia (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
  <confirmationList>
    <serviceID>ServiceID</serviceID>
    <messageID>MessageID</messageID>
    <recurringConfirmations>
      <recurringConfirmed>
        <clientHash>ClientHash</clientHash>
        <confirmation>Confirmation</confirmation>
        <reason>Reason</reason>
      </recurringConfirmed>
    </recurringConfirmations>
    <hash>Hash</hash>
  </confirmationList>
```

Element confirmation

Element **confirmation** służy do przekazania stanu weryfikacji autentyczności operacji przez Serwis. Wartość elementu wyznaczana jest przez sprawdzenie poprawności wartości parametrów **serviceID** oraz **clientHash** z podanymi w komunikacie RPAN przy rozpoczęciu danej płatności aktywacyjnej, a także weryfikację zgodności wyliczonego skrótu z parametrów komunikatu z wartością przekazaną w polu Hash.

Przewidziano dwie wartości elementu **confirmation**:

a) **CONFIRMED** – wartości parametrów są poprawne oraz parametr Hash są zgodne – operacja autentyczna;

b) **NOTCONFIRMED** – wartości w obu komunikatach są niepoprawne lub niezgodność Hash – operacja nieautentyczna;

UWAGA: Element hash (w odpowiedzi na komunikat) służy do autentykacji odpowiedzi i liczony jest z wartości parametrów odpowiedzi. Wartość tego elementu obliczana jest jako wartość funkcji skrótu z łańcucha zawierającego sklejone wartości wszystkich pól dokumentu (bez znaczników) oraz dołączonego klucza współdzielonego. Opis w części [Bezpieczeństwo transakcji](#).

Powiadomienie o dezaktywacji płatności automatycznej (RPDN)

Po wyłączeniu płatności automatycznej dla danego **ClientHash**, wysyłany jest dedykowany komunikat w postaci dokumentu XML, jest protokołem HTTPS (domyślnie port 443), metodą POST, z parametrem o nazwie **recurring**. Parametr ten zapisany jest mechanizmem kodowania transportowego Base64.

Format dokumentu (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
  <recurringDeactivation>
    <serviceID>ServiceID</serviceID>
    <recurringData>
      <recurringAction>RecurringAction</recurringAction>
      <clientHash>ClientHash</clientHash>
      <deactivationSource>DeactivationSource</deactivationSource>
      <deactivationDate>DeactivationDate</deactivationDate>
    </recurringData>
    <hash>Hash</hash>
  </recurringDeactivation>
```

Wartości elementów: serviceID, clientHash dotyczące każdej z deaktywowanych płatności cyklicznych, są identyczne z wartościami odpowiadających im pól, podanymi w komunikacie RPDN przy rozpoczęciu danej płatności inicjalizacyjnej.

Opis zwracanych parametrów

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceID	TAK	string{1,10}	Identyfikator Serwisu Partnera, nadawany w trakcie rejestracji usługi, jednoznacznie identyfikuje Serwis Partnera w Systemie płatności online.

kolejność Hash	nazwa	wymagany	typ	opis
2	recurringData -> recurringAction	TAK	string{1,100}	Akcja w procesie płatności automatycznych (w tym wypadku wartość DEACTIVATE).
3	recurringData -> clientHash	TAK	string{1,64}	Identyfikator płatności automatycznej.
4	recurringData -> deactivationSource	TAK	string{1,64}	Przyczyna dezaktywacji płatności automatycznej. Opis ten należy traktować informacyjnie, lista jego dozwolonych wartości jest ciągle powiększana i pojawienie się nowych wartości nie może pociągać za sobą braku akceptacji komunikatu RPDN. Poniżej aktualne wartości: - SERVICE : zlecone przez Partnerów - ACQUIRER : zlecone przez AP (np. po otrzymaniu informacji o fraudzie) - BM_PL : zlecone przez Klienta na stronie bills.autopay.eu - PAYBM : wynikająca z wygaśnięcia ważności karty.
5	recurringData -> deactivationDate	TAK	string{14}	Moment wyłączenia płatności automatycznej, przekazywany w formacie YYYYMMDDhhmmss. (Czas CET)
nd.	hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis.

UWAGA: Element hash (komunikatu) służy do autentykacji dokumentu. Wartość tego elementu obliczana jest jako wartość funkcji skrótu z łańcucha zawierającego sklejone wartości wszystkich pól dokumentu oraz dołączonego klucza współdzielonego.

Potwierdzenie otrzymania komunikatu

W odpowiedzi na powiadomienie oczekiwany jest status HTTP 200 (OK) oraz tekst w formacie XML (nie kodowany Base64), zwracany przez Serwis w tej samej sesji HTTP, zawierający potwierdzenie otrzymania komunikatu.

Struktura potwierdzenia (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
  <confirmationList>
    <serviceID>ServiceID</serviceID>
    <recurringConfirmations>
      <recurringConfirmed>
        <clientHash>ClientHash</clientHash>
        <confirmation>Confirmation</confirmation>
      </recurringConfirmed>
    </recurringConfirmations>
    <hash>Hash</hash>
  </confirmationList>
```

Element Confirmation

Element **confirmation** służy do przekazania stanu weryfikacji autentyczności operacji przez Serwis. Wartość elementu wyznaczana jest przez sprawdzenie poprawności wartości parametrów **serviceID** oraz **clientHash** z podanymi w komunikacie RPAN przy rozpoczęciu danej płatności inicjalizacyjnej oraz weryfikację zgodności wyliczonego skrótu z parametrów komunikatu z wartością przekazaną w polu hash.

Przewidziano dwie wartości elementu **confirmation**:

- a) **CONFIRMED** – wartości parametrów są poprawne oraz parametr hash są zgodne – operacja autentyczna
- b) **NOTCONFIRMED** – wartości w obu komunikatach są niepoprawne lub niezgodność hash – operacja nieautentyczna

UWAGA: Element hash (w odpowiedzi na komunikat) służy do autentykacji odpowiedzi i liczony jest z wartości parametrów odpowiedzi. Wartość tego elementu obliczana jest jako wartość funkcji skrótu z łańcucha zawierającego sklejone wartości wszystkich pól dokumentu (bez znaczników) oraz dołączonego klucza współdzielonego. Opis sposobu obliczania skrótu znajduje się w części [Bezpieczeństwo transakcji](#).

W wypadku braku poprawnej odpowiedzi na wysłane powiadomienia, System podejmie kolejne próby jego przekazania po upływie określonego czasu. Serwis powinien wykonywać własną logikę biznesową (np. zatrzymanie płatności automatycznej, mailing itp.), jedynie po pierwszym komunikacie RPAN o danym ClientHash.

WSKAZÓWKA: Zalecamy zapoznanie się również z częściami *Monitoring komunikacji ITN/ISTN/IPN/RPAN/RPDN* i *Schemat ponawiania komunikatów ITN/ISTN/IPN/RPAN/RPDN*.

Weryfikacja tożsamości płatnika

Opis weryfikacji tożsamości płatnika

Dokonanie weryfikacji tożsamości płatnika może odbywać się na zasadzie porównania danych przekazanych do weryfikacji w parametrach startu transakcji oraz danych uzyskanych z wpłaty zarejestrowanej w Systemie. Jeśli w trakcie integracji zostanie uzgodniona opcja weryfikacji danych oraz w starcie transakcji zostaną podane zadeklarowane przez klienta dane do weryfikacji, System dokona sprawdzenia prawdziwości tych danych i jego wynik umieści w komunikacie ITN obok danych nadawcy przelewu. Dla większości transakcji typu PBL, pierwszy komunikat ITN nie będzie zawierać danych adresowych (nazw, adres) klienta. Dane te są uzupełniane w Systemie po zeskanowaniu historii rachunku AP danego kanału płatności. Uzupełnienie tych danych skutkuje wysłaniem kolejnego komunikatu ITN do Partnera, zawierającego już te dane.

Pola w komunikacie startu transakcji powiązane z usługą (opis w części [Rozpoczęcie transakcji z dodatkowymi parametrami](#)):

- **VerificationFName,**
- **VerificationLName,**
- **VerificationStreet,**
- **VerificationStreetHouseNo,**
- **VerificationStreetStaircaseNo,**
- **VerificationStreetPremiseNo,**
- **VerificationPostalCode,**
- **VerificationCity,**
- **VerificationNRB**

Pola w komunikacie ITN powiązane z usługą (opisane w części [Dodatkowe pola w komunikacie ITN/IPN transakcji wejściowej](#)):

- węzeł **customerData**
- **verificationStatus**
- węzeł **verificationStatusReasons**

WSKAZÓWKA: Szczegółowe instrukcje, jak interpretować statusy płatności i weryfikacji w tym procesie zawiera część [Dodatkowe pola w komunikacji ITN/IPN transakcji wejściowej](#).

Rozpoczęcie transakcji z dodatkowymi parametrami

Starty transakcji mogą zostać przeprowadzone z dodatkowymi parametrami rozpisanymi w poniższych punktach.

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	typ	opis
8	Language	string{1,2}	Wybór języka, w jakim będą prezentowane treści w Sytemie. Dopuszczalne wartości to: PL, EN, DE, CS, ES, FR, IT. Użycie innych wartości niż PL powinno być potwierdzone w trakcie integracji i powinno zależeć od faktycznego wyboru języka w Serwisie przez Klienta.
9	CustomerNRB	string{26}	Numer rachunku Klienta, parametr przeznaczony wyłącznie dla Serwisów Partnera generujących dedykowane numery rachunków dla zamówienia lub Klienta (patrz Model rozliczeń transakcji po każdej wpłacie). Dopuszczalne tylko cyfry. Jeśli w trakcie integracji ustalono wykorzystanie rachunków spoza Polski, wtedy pole przesyła IBAN i oczekiwany zakres danych pola zmienia się na: alfanumeryczne znaki alfabetu łacińskiego (min. 15, maks. 32 znaki).
10	SwiftCode	string{8,11}	Kod swift odpowiadający podanemu numerowi rachunku. Dopuszczalne tylko cyfry. Parametr podawany, jeśli w trakcie integracji ustalono wykorzystanie rachunków spoza Polski.

kolejność Hash	nazwa	typ	opis
11	ForeignTransferMode	string{4,5}	<p>System jakim ma zostać wykonany zagraniczny przelew rozliczeniowy: SEPA (Single Euro Payments Area) - możliwy do wykonania przelewu w walucie Euro w obrębie państw członkowskich Unii Europejskiej, jak i innych państwach na terenie Starego Kontynentu, np. Islandii, Liechtensteinu, Norwegii, Szwajcarii, Monako czy Andory, SWIFT - przelewy zagraniczne niemożliwe do wykonania za pomocą SEPA (np. inna waluta niż Euro), wiąże się z wyższymi kosztami wykonania przelewu, niż w przypadku SEPA.</p> <p>Dopuszczalne wartości: SEPA i SWIFT. Parametr podawany, jeśli w trakcie integracji ustalono wykorzystanie rachunków spoza Polski.</p>
12	TaxCountry	string{1,64}	Kraj zamieszkania płatnika.
13	CustomerIP	string{1,15}	Adres IP użytkownika, parametr przeznaczony wyłącznie dla Serwisów Partnera uruchamiających System w tle (patrz Przedtransakcja oraz Zamówienie danych do przelewu w transakcji typu Szybki Przelew).
14	Title	string{1,95}	<p>Tytuł przelewu rozliczającego transakcję, parametr przeznaczony wyłącznie dla Serwisów Partnera rozliczanych przelewem po każdej wpłacie (patrz Model rozliczeń transakcji po każdej wpłacie). W niektórych przypadkach, niezależnych od AP tytuł przelewu rozliczeniowego może zostać samodzielnie zmodyfikowany przez Bank, z którego nastąpiło rozliczenie.</p> <p>Dopuszczalne alfanumeryczne znaki alfabetu łacińskiego oraz znaki z zakresu: ĘęÓóĄąŚśŁłŻżŻżĆćŃń\ \s . - / , ! () \ " , gdzie znak "/" będzie podmieniany na "-" dla transakcji wychodzących.</p>
15	ReceiverName	string{1,35}	<p>Nazwa odbiorcy przelewu rozliczającego transakcję, parametr przeznaczony wyłącznie dla Serwisów Partnera rozliczanych przelewem po każdej wpłacie (patrz Model rozliczeń transakcji po każdej wpłacie).</p> <p>Dopuszczalne alfanumeryczne znaki alfabetu łacińskiego oraz znaki z zakresu: ĘęÓóĄąŚśŁłŻżŻżĆćŃń\ \s . - / , ! () = [] { } ; : ?</p>

kolejność Hash	nazwa	typ	opis
16	Products	string{1,10000}	Informacje o produktach wchodzących w skład transakcji, przekazywany w postaci zakodowanego protokołem transportowym Base64 XMLa. Opis struktury w części Koszyk produktów .
17	CustomerPhone	string{9-15}	Numer telefonu użytkownika. Dopuszczalne tylko cyfry.
18	CustomerPesel	string{11}	Numer PESEL użytkownika. Dopuszczalne tylko cyfry.
20	CustomerNumber	string{1,35}	Numer Klienta w Serwisie.
21	InvoiceNumber	string{1,100}	Numer dokumentu finansowego w Serwisie.
22	CompanyName	string{1,150}	Nazwa firmy do automatycznej weryfikacji wpłacającego, np. Firma Fajna.
23	Nip	string{1,10}	Numer identyfikacyjny NIP weryfikowanej firmy, np. 5851351185. Dopuszczalne tylko cyfry.
24	Regon	string{9,14}	Numer identyfikacyjny REGON weryfikowanej firmy, np. 191781561. Dopuszczalne tylko cyfry.
25	VerificationFName	string{1,32}	Imię podane w Serwisie do automatycznej weryfikacji wpłacającego, np. Jan. Dopuszczalne tylko litery alfabetu polskiego.
26	VerificationLName	string{1,64}	Nazwisko podane w Serwisie do automatycznej weryfikacji wpłacającego, np. Kowalski. Dopuszczalne tylko litery alfabetu polskiego.
27	VerificationStreet	string{1,64}	Ulica podana w Serwisie do automatycznej weryfikacji, np. Długa. Dopuszczalne tylko litery alfabetu polskiego oraz cyfry.
28	VerificationStreetHouseNo	string{1,64}	Numer domu podany w Serwisie do automatycznej weryfikacji wpłacającego. Dopuszczalne tylko litery alfabetu polskiego oraz cyfry.
29	VerificationStreetStaircaseNo	string{1,64}	Numer klatki podany w Serwisie do automatycznej weryfikacji wpłacającego. Dopuszczalne tylko litery alfabetu polskiego oraz cyfry.
30	VerificationStreetPremiseNo	string{1,64}	Numer lokalu podany w Serwisie do automatycznej weryfikacji wpłacającego. Dopuszczalne tylko litery alfabetu polskiego oraz cyfry.

kolejność Hash	nazwa	typ	opis
31	VerificationPostalCode	string{1,64}	Kod pocztowy podany w Serwisie do automatycznej weryfikacji wpłacającego, format XX-XXX, np. 80-180. Dopuszczalne tylko cyfry oraz znak -.
32	VerificationCity	string{1,64}	Miasto podane w Serwisie do automatycznej weryfikacji wpłacającego, np. Warszawa. Dopuszczalne tylko litery alfabetu polskiego oraz cyfry.
33	VerificationNRB	string{1,26}	Numer rachunku bankowego podany w Serwisie do automatycznej weryfikacji wpłacającego, np. 88154010982001554242710005. Dopuszczalne tylko cyfry.
35	RecurringAcceptanceState	string{1,100}	Informacja o akceptacji regulaminu płatności automatycznej określająca, czy Klient zaakceptował regulamin płatności automatycznej, czy należy wymusić jego akceptację po stronie Systemu. Pole wymagane dla płatności automatycznych w modelu WhiteLabel usługi płatniczej świadczonej przez AP na rzecz Klienta (Klient płaci prowizję). Dostępność regulaminu można sprawdzić wywołując metodę legalData. Dozwolone wartości: NOT_APPLICABLE - akceptacja regulaminu niewymagana (płatność jednorazowa lub akcja obciążenia, tj. recurringAction o wartości AUTO lub MANUAL) ACCEPTED - deklaracja akceptacji regulaminu w serwisie kontrahenta (należy podać wraz z RecurringAcceptanceID) PROMPT - na formularzu kartowym jest proponowana zgoda na zapisanie karty, jej zaznaczenie rozpoczyna płatność automatyczną FORCE - w formularzu kartowym jest wymagana zgoda na zapisanie karty, inaczej płatność jest niemożliwa. UWAGA: Dostępność opcji ACCEPTED/PROMPT/FORCE zależy od uzgodnień biznesowych (w szczególności ustalenia miejsca wyświetlania zgody/regulaminu usługi płatności automatycznej).

kolejność Hash	nazwa	typ	opis
36	RecurringAction	string{1,100}	<p>Pole wymagane dla płatności automatycznych, określające możliwe akcje na płatności automatycznej.</p> <p>Dozwolone wartości:</p> <p>INIT_WITH_PAYMENT - aktywacja płatności automatycznej wraz z opłatą za towar/usługę</p> <p>INIT_WITH_REFUND - aktywacja płatności automatycznej, a następnie zwrot wpłaty</p> <p>AUTO - płatność cykliczna (obciążenie bez udziału Klienta)</p> <p>MANUAL - płatność jednym kliknięciem (obciążenie zlecane przez Klienta)</p> <p>UWAGA: Opcja niedostępna dla płatności automatycznych BLIK (BLIK OneClick).</p> <p>DEACTIVATE - dezaktywacja płatności automatycznej</p>
37	ClientHash	string{1,64}	<p>Identyfikator płatności automatycznej. Parametr pozwala w sposób zanonimizowany przypisać Instrument płatniczy (np. Kartę, BLIK) do Klienta. Na jego podstawie Partner może wywoływać kolejne obciążenia w modelu płatności automatycznych.</p>
38	OperatorName	string{1,35}	<p>Nazwa operatora podanego numeru telefonu.</p> <p>Dopuszczalne wartości: Plus, Play, Orange, T-Mobile.</p>
39	ICCID	string{12,19}	<p>Numer karty SIM podanego numeru telefonu.</p> <p>Dozwolone wartości (dopuszczalne tylko cyfry):</p> <p>Dla Plus: 12 lub 13 cyfr</p> <p>Dla Play, Orange, T-Mobile: 19 cyfr</p>
40	AuthorizationCode	string{6}	<p>Kod autoryzacji płatności wprowadzany po stronie Serwisu/Systemu (obecnie obsługiwany w BLIK). Jego zastosowanie powoduje, że nie ma potrzeby przekierowania Klienta na stronę Kanału Płatności. Należy zatem podawać go jedynie poprzez Przedtransakcję. Format zależny od Kanału Płatności. Dla BLIK realizowanego w tle (BLIK 0, ew. BLIK OneClick): 6 cyfr.</p>
41	ScreenType	string{4,6}	<p>Rodzaj widoku formatki autoryzującej płatność. Dopuszczalne wartości: IFRAME - niewspierany FULL.</p>
42	BlikUIDKey	string{1,64}	<p>Klucz Aliasu UID (używane w BLIK). Jest to unikalny identyfikator użytkownika w Serwisie.</p> <p>Dopuszczalne alfanumeryczne znaki alfabetu łacińskiego oraz znaki: _.</p>

kolejność Hash	nazwa	typ	opis
43	BlikUIDLabel	string{1,20}	Etykieta Aliasu UID (używane w BLIK), która będzie prezentowana Klientowi w aplikacji bankowej w celu rozróżniania kont u Partnera. Zaleca się stosowanie loginu, nicka lub adresu mailowego przypisanego do zautoryzowanego konta Klienta. W przypadku możliwości wystąpienia w polu danych osobowych (np. adres mailowy jan.kowalski@poczta.pl) należy wykonać utajnienie danych (poprzez zastąpienie 3 kropkami niektórych znaków, np. ja...ki@po...pl). Dopuszczalne alfanumeryczne znaki alfabetu łacińskiego oraz znaki z zakresu: . : @ - , spacja.
44	BlikAMKey	string{1,64}	Klucz Aliasu aplikacji mobilnej banku (używane w BLIK). Jest to unikalny identyfikator konta w BLIK. Dopuszczalne cyfry.
45	ReturnURL	string{1,1000}	Dynamiczny adres powrotu z płatności zaczynający się od http/https. Dopuszczalne poprawne URL. Może zawierać IP, port, subdomenę, polskie znaki, a także (po domenie) parametry i znaki specjalne: ;\+&;%\$#_!=.
46	TransactionSettlementMode	string{2,10}	Możliwość zmiany sposobu rozliczania transakcji. Brak parametru (kompatybilność wstecz) traktowana, jak przesłanie wartości COMMON. Parametr NONE powoduje potraktowanie transakcji jako zasilenia salda przedpłaconego i brak rozliczenia. Dopuszczalne wartości: COMMON NONE
47	PaymentToken	string{1,100000}	Token używany w portfelach Visa oraz Google Pay umieszczanych bezpośrednio na stronie Partnera (autoryzacja bez przekierowania do Systemu). W tym wypadku Serwis integruje się bezpośrednio z API Visy i/lub Google w celu pobrania uchwytu do karty. Uzyskany token jest przekazywany do Systemu Płatności Online w postaci zakodowanej protokołem transportowym Base64. UWAGA: Parametr jest zbędny, jeśli wybór Kanałów Płatności (oraz logowanie do portfela) odbywa się bezpośrednio na stronie Systemu Płatności Online.
48	DocNumber	string{1,150}	Numer dokumentu finansowego.

kolejność Hash	nazwa	typ	opis
49	RecurringAcceptanceID	string{1,10}	Identyfikator wyświetlanego w Serwisie i akceptowanego przez Klienta regulaminu usługi płatności automatycznej. Pole wymagane dla płatności automatycznych w modelu WhiteLabel usługi płatniczej świadczonej przez AP na rzecz Klienta (Klient płaci prowizję). ID regulaminu odpowiedniego dla wybranego języka (i kanału płatności) należy pobrać za pomocą metody legalData .
50	RecurringAcceptanceTime	string{1,19}	Pole opcjonalne. Moment akceptacji regulaminu przez Klienta, ta wartość będzie weryfikowana przez System z czasem obowiązywania regulaminu o podanym RecurringAcceptanceID . Przykładowa wartość: 2014-10-30 07:54:50. (Czas w CET)
51	DefaultRegulationAcceptanceState	string{1,100}	Informacja o akceptacji regulaminu usługi płatniczej. Pole wymagane w modelu WhiteLabel usługi płatniczej świadczonej przez AP na rzecz Klienta (Klient płaci prowizję). Jego niepodanie może wiązać się z błędem lub wyświetleniem strony przejściowej Systemu z wymaganiem akceptacji regulaminów. Dostępność regulaminu można sprawdzić wywołując metodę legalData . Dozwolone wartości: ACCEPTED - akceptacja regulaminu wykonana w serwisie kontrahenta (należy podać wraz z DefaultRegulationAcceptanceID).
52	DefaultRegulationAcceptanceID	string{1,10}	Identyfikator wyświetlanego w Serwisie i akceptowanego przez Klienta regulaminu usługi płatniczej świadczonej przez AP na rzecz Klienta. Pole wymagane w modelu WhiteLabel usługi płatniczej świadczonej przez AP na rzecz Klienta (Klient płaci prowizję). ID regulaminu odpowiedniego dla wybranego języka (i kanału płatności) należy pobrać za pomocą metody legalData .
53	DefaultRegulationAcceptanceTime	string{1,19}	Pole opcjonalne. Moment akceptacji regulaminu przez Klienta, ta wartość będzie weryfikowana przez System z czasem obowiązywania regulaminu o podanym DefaultRegulationAcceptanceID ; przykładowa wartość: 2014-10-30 07:54:50. (Czas w CET)

kolejność Hash	nazwa	typ	opis
54	WalletType	string{1,32}	Typ portfela płatniczego, określa źródło parametru PaymentToken (jeżeli został przesłany). _Dostępne wartości to: SDK_NATIVE – natywna formatka kartowa (SDK mobilne) WIDGET – widget kartowy (formatka kartowa przed startem transakcji) _
55	RecurringValidityTime	string{10}	Data ważności aktywowanej płatności automatycznej BLIK, format YYYY-MM-DD (przykładowa wartość: 2024-01-30). W przypadku braku parametru, zostanie zaproponowany termin domyślny konfiguracji ustalany w trakcie integracji (zwyczajowo bezterminowy).
56	ServiceURL	string{1,1000}	Parametr określający adres www sklepu, z którego została wystartowana płatność, zaczynający się od http/https. Dopuszczalne poprawne URL. Może zawierać IP, port, subdomenę, polskie znaki, a także (po domenie) parametry i znaki specjalne: ',+&;%\$#_!=
57	BlikPPLabel	string{1,35}	Etykieta płatności powtarzalnej BLIK, wyświetlana w aplikacji mobilnej podczas jej akceptowania. W przypadku braku parametru, zostanie użyta jej wartość domyślna (ustalana w trakcie integracji).
58	ReceiverNameForFront	string{1,35}	Nazwa odbiorcy płatności BLIK, wyświetlana w aplikacji mobilnej podczas jej akceptowania. W przypadku braku parametru, zostanie użyta jej wartość domyślna (zwyczajowo adres URL Serwisu). UWAGA: Usługa musi zostać uzgodnienia z opiekunem biznesowym. Dopuszczalne alfanumeryczne znaki alfabetu łacińskiego, znaki z zakresu: ĘęÓóAąŚśŁłŻżŹźĆćŃń-/,!()=[]{};:~.? oraz spacja.

Koszyk produktów

Opis koszyka produktów

Koszyk produktów przesyłany jest jako parametr (metody POST) o nazwie Products. Jego wartość jest zakodowana protokołem transportowym Base64.

Format przed zakodowaniem (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
  <productList>
    <product>
```



```

        <subAmount>SubAmount1</subAmount>
        <params>Params1</params>
    </product>
</product>
        <subAmount>SubAmount2</subAmount>
        <params>Params2</params>
    </product>
    ...
</product>
        <subAmount>SubAmountN</subAmount>
        <params>ParamsN</params>
    </product>
</productList>

```

Węzeł **productList** musi zawierać przynajmniej 1 element **product**, każdy węzeł **product** musi zawierać po jednym elemencie **subAmount** i **params**.

Element **subAmount** musi zawierać dodatnią kwotę produktu (separatorem dziesiętnym jest kropka, a po niej występują dwie cyfry groszy). Suma kwot kolejnych produktów musi być równa kwocie podanej w parametrze **Amount** (kwocie transakcji).

W przypadku niespełnienia powyższych warunków System zwróci błąd.

Element Params

Element **params** może służyć do przekazywania informacji charakterystycznych dla danego produktu. Nazwy parametrów oraz ich znaczenie podlega każdorazowo uzgodnieniom w formie roboczej podczas integracji.

Przykładowe parametry produktu i ich znaczenie poniżej.

- W tym wypadku węzeł zawiera nazwę produktu:

```

<params>
    <param name="productName" value="Nazwa produktu 1" />
</params>

```

- W tym wypadku węzeł zawiera dwie wartości przypisane do danego produktu, mogące oznaczać przykładowo typ produktu oraz jego nazwę:

```

<params>
    <param name="productType" value="ABCD" />
    <param name="productName" value="Nazwa produktu 1" />
</params>

```

- W przypadku, gdy Serwis posiada saldo w Systemie, oraz planuje wykonywać zwroty do Klienta całości, bądź części kwoty wpłaconej na rzecz wskazanego produktu, zobowiązany jest przekazywać w produkcie jego **unikalny** identyfikator (parametr o nazwie productID o typie string{1,36} (Dopuszczalne alfanumeryczne znaki alfabetu łańciskowego oraz znaki: _ i -)):

```

<params>
    <param name="productID" value="12456" />

```

```
</params>
```

- W przypadku, gdy Partner korzysta z rozszerzonej struktury (wiele punktów rozliczeń), zobowiązany jest przekazywać w każdym produkcie identyfikator punktu rozliczeń (parametr o nazwie idBalancePoint o typie integer{1,10}):

```
<params>  
  <param name="idBalancePoint" value="12456" />  
</params>
```

- W przypadku, gdy Partner korzysta z rozliczeń **MASS_TRANSFER**, tj:
 - każda transakcja rozliczana jest bezpośrednio po wpłacie oraz
 - rozliczenia wykonują się na poziomie produktu/punktu rozliczeń (czyli po wpłacie wykonywane jest tyle przelewów rozliczeniowych, co produktów w koszyku), oraz
 - nie ustalono stałych danych do rozliczeń (lub nie wszystkie dane do rozliczeń są ustalone na sztywno w konfiguracji rozliczeń),

Partner musi przekazać w każdym produkcie brakujące dane do rozliczeń tego produktu. Dostępne parametry stanowiące dane to:

- **customerNRB** - docelowy numer rachunku do rozliczeń.
Format NRB (26 cyfr z sumą kontrolną). Jeśli w trakcie integracji ustalono wykorzystanie rachunków spoza Polski, wtedy pole przenosi IBAN i oczekiwany zakres danych pola zmienia się na: alfanumeryczne znaki alfabetu łacińskiego (min. 15, max. 32 znaki). Podanie wartości w formacie IBAN spowoduje potrzebę podania w produkcie również parametrów swiftCode, foreignTransferMode.
- **title** - tytuł przelewu rozliczającego produkt. W niektórych przypadkach, niezależnych od AP, tytuł przelewu rozliczeniowego może zostać samodzielnie zmodyfikowany przez Bank, z którego nastąpiło rozliczenie.
Dopuszczalne alfanumeryczne znaki alfabetu łacińskiego oraz znaki z zakresu: `ĘęÓóĄąŚśŁłŻżŻżĆćŃń\.-,!()`
- **receiverName** - nazwa odbiorcy przelewu rozliczającego produkt.
Dopuszczalne alfanumeryczne znaki alfabetu łacińskiego oraz znaki z zakresu: `ĘęÓóĄąŚśŁłŻżŻżĆćŃń\.-,!()=[]{ };:;?`

Przykład zastosowania parametrów w produkcie:

```
<params>
```

```
<param name="customerNRB" value="83109010980000000107285707" />
<param name="title" value="Rozliczenie produktu X" />
<param name="receiverName" value="Jan Kowalski" />
</params>
```

- W modelu Marketplace, Partner zobowiązany jest przekazywać w każdym produkcie identyfikator punktu rozliczeń (parametr o nazwie **idBalancePoint** o typie integer{1,10}). Dotyczy to również zasileń salda punktu rozliczeń.

Przykładowe parametry koszyka:

```
<params>
  <param name="idBalancePoint" value="12456" />
  <param name="productName" value="Zasilenie salda dla Autopay" />
</params>
```

Wyświetlanie koszyka produktów na ekranie wyboru Kanału Płatności

Jeżeli w trakcie rozmów dotyczących koszyka produktów ustalono, że jego podsumowanie ma zostać wyświetlone na stronie Systemu (ekran wyboru Kanału Płatności), można określić etykiety każdego użytego w koszyku parametru. System może użyć domyślnej etykiety parametru lub może przyjąć ją w starcie transakcji.

Wartość atrybutu **title** zostanie wyświetlona przed wartością parametru produktu.

Przykład atrybutu title

```
<params>
  <param name="productName" value="Nazwa produktu 1" title="Nazwa"/>
  <param name="productType" value="ABCD" title="Typ"/>
</params>
```

Dodatkowe opcje komunikacji online do Partnera

Dodatkowe pola w komunikacie ITN/IPN transakcji wejściowej

Opis

Natychmiastowe powiadomienia o zmianie statusu transakcji mogą zawierać dodatkowe pola (patrz [Schematy dla Preautoryzacji](#)). Ich występowanie jest kwestią konfiguracyjną, ustalaną w trakcie integracji (domyślnie wysyłane jest tylko węzeł **customerData**).

O tym, czy jest to komunikat ITN, czy IPN decyduje jedynie występowanie węzła **product**.

Pełna lista dodatkowych pól w komunikacie ITN/IPN

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	typ	opis
11	addressIP	string{1,15}	Adres IP Klienta, zarejestrowany przez front Systemu, ew. adres przekazany do Systemu w parametrze CustomerIP, bądź IP z którego nastąpił start transakcji w Systemie.
13	customerNumber	string{1,35}	Numer Klienta w Serwisie.
21	title	string{1,140}	Tytuł wpłaty. W niektórych przypadkach, niezależnych od AP tytuł przelewu może zostać samodzielnie zmodyfikowany przez Bank, w którym nastąpiła wpłata dokonana przez klienta.
22	customerData-> fName	string{1,128}	Imię płatnika.
23	customerData-> lName	string{1,128}	Nazwisko płatnika.
24	customerData-> streetName	string{1,128}	Nazwa ulicy płatnika.
25	customerData-> streetHouseNo	string{1,10}	Numer domu płatnika.
26	customerData-> streetStaircaseNo	string{1,10}	Numer klatki płatnika.
27	customerData-> streetPremiseNo	string{1,10}	Numer lokalu płatnika.
28	customerData-> postalCode	string{1,6}	Kod pocztowy adresu płatnika.
29	customerData-> city	string{1,128}	Miasto płatnika.
30	customerData-> nrb	string{1,26}	Rachunek bankowy płatnika.
31	customerData-> senderData	string{1,600}	Dane płatnika w postaci niepodzielonej.
32	verificationStatus	enum	Element zawierający status weryfikacji płatnika. To enum dopuszczający wartości: PENDING, POSITIVE oraz NEGATIVE.
nd.	verificationStatusReasons	list	Lista zawierająca powody negatywnej lub oczekującej weryfikacji. Powodów może być wiele.

kolejność Hash	nazwa	typ	opis
33	verificationStatus	enum	<p>Szczegółowy powód w przypadku negatywnej lub oczekującej weryfikacji. Dozwolone wartości dla negatywnej weryfikacji:</p> <ul style="list-style-type: none"> - NAME - nie zgadza się imię lub nazwisko - NRB - nie zgadza się numer rachunku - TITLE - nie zgadza się tytuł - STREET - nie zgadza się nazwa ulicy - HOUSE_NUMBER - nie zgadza się numer domu - STAIRCASE - nie zgadza się numer klatki schodowej - PREMISE_NUMBER - nie zgadza się numer lokalu - POSTAL_CODE - nie zgadza się kod pocztowy - CITY - nie zgadza się miasto - BLACKLISTED - rachunek, z którego została wykonana wpłata znajduje się na czarnej liście - SHOP_FORMAL_REQUIREMENTS - weryfikowany serwis nie spełnił warunków formalnych <p>Dozwolone wartości dla oczekującej weryfikacji:</p> <ul style="list-style-type: none"> - NEED_FEEDBACK - trwa oczekiwanie na spełnienie przez serwis warunków formalnych. UWAGA: Do liczenia wartości Hash pobierane są wartości kolejnych węzłów: verificationStatusReasons, verificationStatusReason.
60	startAmount	amount	<p>Kwota transakcji podana w Linku Płatności (nie uwzględnia ew. kwoty prowizji naliczonej Klientowi). Suma prowizji Klienta i startAmount znajduje się w polu amount, ponieważ jest to wynikowa wartość transakcji). Jako separator dziesiętny używana jest kropka - '.' Format: 0.00; maksymalna długość: 14 cyfr przed kropką i 2 po kropce.</p>
70	recurringData-> recurringAction	string{1,100}	<p>Akcja w procesie płatności automatycznej (znaczenie i dozwolone wartości opisano w części Definicje).</p>
71	recurringData-> clientHash	string{1,64}	<p>Identyfikator płatności automatycznej generowany przez AP i przekazywany do Partnera po skutecznej aktywacji płatności automatycznej.</p>

kolejność Hash	nazwa	typ	opis
72	recurringData-> expirationDate	string{14}	Moment wygaśnięcia ważności płatności automatycznej, przekazywany w formacie YYYYMMDDhhmmss. (Czas CET)
73	cardData-> index	string{1,64}	Index karty (jeśli użyto karty). Index identyfikuje kartę o danej dacie ważności (zmiana daty lub numeru karty powoduje zmianę wartości tego parametru).
74	cardData-> validityYear	string{4}	Ważność karty w formacie YYYY (jeśli użyto karty).
75	cardData-> validityMonth	string{4}	Ważność karty w formacie mm (jeśli użyto karty).
76	cardData-> issuer	string{1,64}	Typ karty (jeśli użyto karty). Możliwe wartości: - VISA - MASTERCARD - MAESTRO - AMERICAN EXPRESS (obecnie nie wspierane) - DISCOVER (obecnie nie wspierane) - DINERS (obecnie nie wspierane) - UNCATEGORIZED (nierozpoznany wystawca)
77	cardData-> bin	string{6}	Pierwsze 6 cyfr numeru karty (jeśli użyto karty). Przekazywane, jeśli nie przekazywany jest parametr cardData-> mask.
78	cardData-> mask	string{4}	Ostatnie 4 cyfry numeru karty (jeśli użyto karty). Przekazywane, jeśli nie przekazywany jest parametr cardData-> bin.
90	product-> subAmount	amount	Kwota produktu jako separator dziesiętny używana jest kropka - '.' Format: 0.00; maksymalna długość: 14 cyfr przed kropką i 2 po kropce. Węzeł dostępny tylko w komunikatach IPN.
91	product-> params	list	Kolejne parametry produktu zgodne z formatem koszyka w starcie transakcji. Węzeł dostępny tylko w komunikatach IPN.

UWAGA: Do liczenia wartości Hash pobierane są atrybuty **value** kolejnych węzłów **product.params**.

Przykład komunikatu ITN/IPN z dodatkowymi parametrami (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<transactionList>
  <serviceID>ServiceID</serviceID>
  <transactions>
    <transaction>
      <orderID>OrderID</orderID>
      <remoteID>RemoteID</remoteID>
      <amount>999999.99</amount>
      <currency>PLN</currency>
      <gatewayID>GatewayID</gatewayID>
      <paymentDate>YYYYMMDDhhmmss</paymentDate>
      <paymentStatus>PaymentStatus</paymentStatus>
      <paymentStatusDetails>PaymentStatusDetails</paymentStatusDetails>
      <addressIP>127.0.0.1</addressIP>
      <customerNumber>1111111</customerNumber>
      <title>title</title>
      <customerData>
        <fName>fName</fName>
        <lName>lName</lName>
        <streetName>streetName</streetName>
        <streetHouseNo>streetHouseNo</streetHouseNo>
        <streetStaircaseNo>streetStaircaseNo</streetStaircaseNo>
        <streetPremiseNo>streetPremiseNo</streetPremiseNo>
        <postalCode>postalCode</postalCode>
        <city>city</city>
        <nrb>nrb</nrb>
        <senderData>senderData</senderData>
      </customerData>
      <verificationStatus>verificationStatus</verificationStatus>
      <verificationStatusReasons>
        <verificationStatusReason>reason1</verificationStatusReason>
        <verificationStatusReason>reason2</verificationStatusReason>
        <verificationStatusReason>reason3</verificationStatusReason>
      </verificationStatusReasons>
      <startAmount>999998.99</startAmount>
      <recurringData>
        <recurringAction>RecurringAction</recurringAction>
        <clientHash>ClientHash</clientHash>
        <expirationDate>YYYYMMDDhhmmss</expirationDate>
      </recurringData>
      <cardData>
        <index>Index</index>
        <validityYear>ValidityYear</validityYear>
        <validityMonth>ValidityMonth</validityMonth>
        <issuer>Issuer</issuer>
        <bin>BIN</bin>
      </cardData>
      <product>
        <subAmount>SubAmount</subAmount>
        <params>
          <param name="idBalancePoint" value="idBalancePoint"/>
          <param name="invoiceNumber" value="invoiceNumber"/>
          <param name="customerNumber" value="customerNumber"/>
          <param name="subAmount" value="SubAmount"/>
        </params>
      </product>
    </transaction>
  </transactions>
  <hash>Hash</hash>

```

</transactionList>

Szczegółowy opis zmiany statusu weryfikacji - dla transakcji zakończonej poprawnie (wynik pozytywny lub negatywny)

Status Płatności (paymentStatus)	Status Weryfikacji (verificationStatus)	Szczegóły Weryfikacji (verificationStatusReasons)	Szczegóły
PENDING	PENDING	Puste	Klient wybrał metodę płatności.
SUCCESS	PENDING	Puste	Transakcja została opłacona, System oczekuje na pozyskanie danych wpłacającego z rachunku.
SUCCESS	PENDING	NEED_FEEDBACK	Autopay oczekuje na spełnienie warunków formalnych przez Partnera.
SUCCESS	POSITIVE	Puste	Weryfikacja przebiegła pozytywnie.
SUCCESS	NEGATIVE	Lista powodów dla NEGATIVE	Weryfikacja negatywna.

Szczegółowy opis zmiany statusu weryfikacji - dla transakcji nie zakończonej poprawnie

Status Płatności (paymentStatus)	Status Weryfikacji (verificationStatus)	Szczegóły Weryfikacji (verificationStatusReasons)	Szczegóły
PENDING	PENDING	Puste	Klient wybrał metodę płatności.
FAILURE	PENDING	Puste	Transakcja nie została zakończona poprawnie. Status weryfikacji nie zostanie dostarczony.

Szczegółowe statusy transakcji

Komunikat ITN dla transakcji wejściowej zawiera oprócz statusu płatności (pole **paymentStatus**)

szczegółowy opis tego statusu (pole **paymentStatusDetails**). Opis ten należy traktować informacyjnie, lista jego dozwolonych wartości jest ciągle powiększana i pojawienie się nowych wartości nie może pociągać za sobą brak akceptacji komunikatu ITN.

Wartości statusów transakcji - Statusy ogólne (niezależne od kanału płatności)

Wartość pola	Znaczenie pola
AUTHORIZED	transakcja zautoryzowana przez Kanał Płatności
ACCEPTED	transakcja zatwierdzona przez Call Center (np. w wyniku pozytywnie rozpatrzonej reklamacji)
REJECTED	transakcja przerwana przez Kanał Płatności (bank/agenta rozliczeniowego)
REJECTED_BY_USER	transakcja przerwana przez Klienta
INCORRECT_AMOUNT	wpłacono kwotę różną od kwoty podanej przy starcie transakcji
EXPIRED	transakcja przeterminowana
CANCELLED	transakcja anulowana przez Serwis Partnera lub Call Center (np. na prośbę Klienta). Nie jest możliwe wystartowanie nowej ani kontynuowanie wcześniej wystartowanej transakcji o tym samym OrderID
RECURSION_INACTIVE	błąd aktywności płatności cyklicznej
ANOTHER_ERROR	wystąpił inny błąd przy przetwarzaniu transakcji

Wartości statusów transakcji - Statusy kartowe

Wartość pola	Znaczenie pola	Opcjonalny kod błędu organizacji kartowej
CONNECTION_ERROR	błąd z połączeniem do banku wystawcy karty płatniczej	✓
CARD_LIMIT_EXCEEDED	błąd limitów na karcie płatniczej	✓
SECURITY_ERROR	błąd bezpieczeństwa (np. nieprawidłowy cvv)	✓
DO_NOT_HONOR	odmowa autoryzacji w banku; sugerowany kontakt klienta z wystawcą karty	✓
THREEDS_NEGATIVE	transakcja nieudana w systemie 3DS	✓
CARD_EXPIRED	karta nieważna	✓
INCORRECT_CARD_NUMBER	nieprawidłowy numer karty	✓
FRAUD_SUSPECT	podejrzenie fraudu (np. zagubiona karta itp.)	✓
STOP_RECURRING	rekurencja niemożliwa z powodu anulowania dyspozycji klienta	✓
VOID	transakcja porzucona lub błąd komunikacyjny	✗

Wartość pola	Znaczenie pola	Opcjonalny kod błędu organizacji kartowej
UNCLASSIFIED	pozostałe błędy	✓

Wartości statusów transakcji - Statusy specyficzne dla transakcji BLIK

Wartość pola	Znaczenie pola
INSUFFICIENT_FUNDS	Brak środków. Zalecane wyświetlenie Klientowi komunikatu o treści: <i>Płatność nieudana - odmowa banku. Sprawdź powód odmowy w aplikacji bankowej.</i> <i>Jeśli powodem jest przekroczenie limitu, możesz go podwyższyć kontaktując się z bankiem.</i>
LIMIT_EXCEEDED	Błąd limitów (np. kwotowych). Zalecane wyświetlenie Klientowi komunikatu wyświetlenie informacji o treści: <i>Płatność nieudana - odmowa banku. Sprawdź powód odmowy w aplikacji bankowej..</i> <i>Jeśli powodem jest przekroczenie limitu, możesz go podwyższyć kontaktując się z bankiem.</i>
BAD_PIN	podano nieprawidłowy PIN podczas potwierdzania transakcji
ISSUER_DECLINED, USER_DECLINED, SEC_DECLINED	transakcja przerwana przez Klienta
TIMEOUT i AM_TIMEOUT	timeout w komunikacji z aplikacją mobilną banku
USER_TIMEOUT	timeout oczekiwania na potwierdzenie transakcji przez Klienta

Natychmiastowe powiadomienia o zmianie statusu produktu (IPN)

W przypadku Partnera korzystającego z [rozszerzonej struktury \(Koszyka produktów\)](#) z wieloma punktami rozliczeń), System udostępnia niezależne powiadomienie o zmianach statusów każdego z produktów. Taka usługa ma sens, jeśli poszczególne punkty rozliczeń powinny otrzymywać własne powiadomienia. W tym przypadku konfiguracja IPN (adres do powiadomień, umieszczane w komunikacie pola itp.) przechowywana jest właśnie na poziomie konfiguracji punktu rozliczeń. Struktura IPN jest podobna do ITN (rozszerzona jedynie o węzeł **product** podobny do opisanego w w rozdziale [Dodatkowe pola w komunikacie ITN/IPN transakcji wejściowej](#), uzupełniony jedynie o powtórzenie kwoty **subAmount** w params). Przykład IPN w podrozdziale [Dodatkowe pola w komunikacie ITN/IPN transakcji wejściowej](#)).

Natychmiastowe powiadomienia o zmianie statusu transakcji rozliczeniowej (ISTN)

Istnieje możliwość dostarczania komunikatów o wszystkich wypłatach (rozliczeniach, wypłatach z salda i zwrotach) wykonywanych przez System w ramach usługi płatności. Ponieważ usługa nie jest domyślnie uruchomiona, zapotrzebowanie na nią, wraz z adresem do wysyłki ISTN, musi być przez Partnera zgłoszone w trakcie ustalania wymagań.

W przypadku skutecznego uruchomienia komunikacji ISTN, System niezwłocznie przekazuje powiadomienia o fakcie zlecenia transakcji rozliczeniowej (ew. wypłatach/zwrotach) oraz zmianie jej statusu. Potwierdzenia przesyłane są, na ustalony w trakcie dodawania konfiguracji Serwisu Partnera, adres na serwerze Serwisu Partnera:

```
https://sklep_nazwa/odbior_informacji_o_rozliczeniu
```

Powiadomienie to polega na wysłaniu przez System dokumentu XML zawierającego nowe statusy transakcji. Dokument wysyłany jest protokołem HTTPS (domyślnie port 443). Dokument przesyłany jest metodą POST, jako parametr HTTP o nazwie transactions. Parametr ten zapisany jest mechanizmem kodowania transportowego Base64.

Format dokumentu (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
  <transactionList>
    <serviceID>ServiceID</serviceID>
    <transactions>
      <transaction>
        <isRefund>true/false</isRefund>
        <productID>ProductID</productID>
        <orderID>OrderID</orderID>
        <orderOutID>OrderOutID</orderOutID>
        <remoteID>RemoteID</remoteID>
        <remoteOutID>RemoteOutID</remoteOutID>
        <amount>999999.99</amount>
        <currency>PLN</currency>
        <transferDate>YYYYMMDDhhmmss</transferDate>
        <transferStatus>TransferStatus</transferStatus>
      <transferStatusDetails>TransferStatusDetails</transferStatusDetails>
        <title>Title</title>
        <receiverBank>ReceiverBank</receiverBank>
        <receiverNRB>ReceiverNRB</receiverNRB>
        <receiverName>ReceiverName</receiverName>
        <receiverAddress>ReceiverAddress</receiverAddress>
        <senderBank>SenderBank</senderBank>
        <senderNRB>SenderNRB</senderNRB>
      </transaction>
    </transactions>
    <hash>Hash</hash>
  </transactionList>
```

Zwracane parametry

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceID	NIE	string{1,10}	Identyfikator Serwisu Partnera, nadawany w trakcie rejestracji usługi, jednoznacznie identyfikuje Serwis Partnera w Systemie płatności online.

kolejność Hash	nazwa	wymagany	typ	opis
2	isRefund	NIE	Boolean	Informacja, czy ISTN dotyczy zwrotu transakcji (true), czy normalnego rozliczenia (false).
3	productID	NIE	string{1,36}	Identyfikator rozliczanego produktu z koszyka produktów transakcji wejściowej, wartość pola musi być unikalna dla Serwisu Partnera.
4	orderID	NIE	string{1,32}	Identyfikator transakcji wejściowej o długości do 32 znaków alfanumerycznych alfabetu łańciskiego, wartość pola musi być unikalna dla Serwisu Partnera.
5	orderOutID	NIE	string{1,32}	Identyfikator transakcji wyjściowej o długości do 32 znaków alfanumerycznych alfabetu łańciskiego. Pole może być nadawane przez Serwis (w przypadku zlecenia rozliczenia) lub przez System płatności online.
6	remoteID	NIE	string{1,20}	Alfanumeryczny identyfikator transakcji wejściowej nadany przez System płatności online (podany, jeśli do rozliczenia dowiązana jest jedna wpłata).
7	remoteOutID	NIE	string{1,20}	Alfanumeryczny identyfikator transakcji rozliczeniowej nadany przez System płatności online.
8	amount	TAK	amount	Kwota transakcji. Jako separator dziesiętny używana jest kropka - '.' Format: 0.00; maksymalna długość: 14 cyfr przed kropką i 2 po kropce.
9	currency	TAK	string{1,3}	Waluta transakcji.
40	transferDate	NIE	string{14}	Moment zautoryzowania transakcji, przekazywany w formacie YYYYMMDDhhmmss. (Czas CET). Występuje jedynie dla transferStatus=SUCCESS.
41	transferStatus	TAK	enum	Status autoryzacji transakcji rozliczeniowej. Przyjmuje następujące wartości: - PENDING - przelew oczekuje na wykonanie - SUCCESS - przelew zlecono do banku - FAILURE - nie można wykonać przelewu, np. błędny numer rachunku
42	transferStatusDetails	NIE	enum	Szczegółowy status transakcji, wartość może być ignorowana przez Serwis Partnera. Przyjmuje poniższe wartości (lista może zostać rozszerzona): - AUTHORIZED - transakcja przekazana do realizacji w banku - CONFIRMED - transakcja potwierdzona w banku (fizycznie wysłane pieniądze) - CANCELLED - transakcja anulowana przez Serwis Partnera lub Call Center (np. na prośbę Serwisu) - ANOTHER_ERROR - wystąpił inny błąd przy przetwarzaniu transakcji
43	title	NIE	string{1,140}	Tytuł przelewu rozliczającego transakcję. W niektórych przypadkach, niezależnych od AP tytuł przelewu rozliczeniowego może zostać samodzielnie zmodyfikowany przez Bank, z którego nastąpiło rozliczenie. Dopuszczalne alfanumeryczne znaki alfabetu łańciskiego oraz znaki z zakresu: ĘęÓóĄąŚśłłŻżŹźĆćŃń\ \s . - / , ! @ # % ^ \ * () \ _ = + [\] { } ; : ? , gdzie znak "/" będzie podmieniany na "-" dla transakcji wychodzących.
44	receiverBank	NIE	string{1,64}	Nazwa banku, do którego System wykonał przelew.
45	receiverNRB	NIE	string{26}	Numer rachunku bankowego odbiorcy przelewu.
46	receiverName	NIE	string{1,140}	Nazwa odbiorcy przelewu. Dopuszczalne alfanumeryczne znaki alfabetu łańciskiego oraz znaki z zakresu: ĘęÓóĄąŚśłłŻżŹźĆćŃń\ \s . - / , ! @ # % ^ \ * () \ _ = + [\] { } ; : ?
47	receiverAddress	NIE	string{1,140}	Adres odbiorcy przelewu. Dopuszczalne alfanumeryczne znaki alfabetu łańciskiego oraz znaki z zakresu: ĘęÓóĄąŚśłłŻżŹźĆćŃń\ \s . - / , ! @ # % ^ \ * () \ _ = + [\] { } ; : ?
48	senderBank	NIE	string{1,64}	Nazwa banku, za pomocą którego System wykonał przelew.

kolejność Hash	nazwa	wymagany	typ	opis
49	senderNRB	NIE	string{26}	Numer rachunku bankowego nadawcy przelewu.
nd.	hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis.

Odpowiedź na powiadomienie

W odpowiedzi na powiadomienie oczekiwany jest tekst w formacie XML (nie kodowany Base64), zwracany przez Serwis Partnera w tej samej sesji HTTP, zawierający potwierdzenie otrzymania statusu transakcji.

Struktura potwierdzenia (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
  <confirmationList>
    <serviceID>ServiceID</serviceID>
    <transactionsConfirmations>
      <transactionConfirmed>
        <remoteOutID>RemoteOutID</remoteOutID>
        <confirmation>Confirmation</confirmation>
      </transactionConfirmed>
    </transactionsConfirmations>
    <hash>Hash</hash>
  </confirmationList>
```

Element **confirmation** służy do przekazania stanu weryfikacji autentyczności transakcji przez Serwis Partnera. Wartość elementu wyznaczana jest przez sprawdzenie poprawności wartości parametru serviceID, a także weryfikację zgodności wyliczonego skrótu z wartością przekazaną w polu hash.

Przewidziano dwie wartości tego elementu:

- a) **CONFIRMED** – parametr hash jest zgodny – transakcja autentyczna;
- b) **NOTCONFIRMED** – parametr hash jest niezgodny – transakcja nieautentyczna;

W wypadku braku poprawnej odpowiedzi na wysłane powiadomienia, System podejmie kolejne próby przekazania nowego statusu po upływie określonego czasu. Serwis Partnera powinien wykonywać własną logikę biznesową, jedynie po pierwszym komunikacie o danym statusie płatności.

WSKAZÓWKA: Warto zapoznać się ze [Schematem ponawiania komunikatów ITN/ISTN/IPN/RPAN/RPDN](#).

Szczegółowy opis zachowania i zmiany statusów rozliczenia (transferStatus)

W podstawowym modelu, System dostarczy jedynie status **SUCCESS**, możliwe jest jednak dokładniejsze powiadamianie. Opcja pełna powinna być zgłoszona podczas integracji i wiąże się z poniższym schematem przejść statusów.

Zlecenie transakcji rozliczeniowej powoduje wysłanie statusu **PENDING**. Później system dostarczy **SUCCESS** lub **FAILURE**. Dla transakcji, dla której wystąpił status **SUCCESS**, nie powinna już nastąpić zmiana statusu na **FAILURE**. Może jednakże nastąpić zmiana statusu szczegółowego (kolejne komunikaty o zmianie statusu szczegółowego są jedynie informacyjne i nie powinny pociągać za sobą ponownego wykonywania żadnej logiki biznesowej).

W szczególnych przypadkach (np. błąd w banku) transakcja pierwotnie potwierdzona, może zostać przekazana do ponownego wykonania, a więc zmienić swój status na **PENDING** i ponownie na **SUCCESS**.

Innym szczególnym przypadkiem może być status **FAILURE** (np. po błędzie wewnętrznym Systemu), następnie zastąpiony statusem **SUCCESS**.

Usługi dodatkowe

Odpytywanie o listę aktualnie dostępnych Kanałów Płatności

Opis

Aby zbudować w Serwisie widok wyboru metody płatności, System umożliwia zdalne odpytanie o aktualną listę kanałów płatności. W tym celu należy wywołać metodę **gatewayList** (https://host_bramki/gatewayList/v2) z odpowiednimi parametrami (w formacie JSON). Wszystkie parametry przekazywane są za pomocą technologii REST. Protokół rozróżnia wielkość liter zarówno w nazwach jak i wartościach parametrów. Wartości przekazywanych parametrów powinny być kodowane w UTF-8.

Lista dostępnych parametrów

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	ServiceID	TAK	integer	Identyfikator Serwisu Partnera.
2	MessageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łacińskiego (np. na bazie UID). Wartość pola musi być unikalna dla Serwisu Partnera.
3	Currencies	TAK	string{0,1000}	Lista walut, których lista dostępnych kanałów ma być zwrócona. Lista powinna być minimum jednoelementowa. Dopuszczalne jedynie wartości: PLN, EUR, GBP oraz USD.
nd.	Hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis.

Przykładowy komunikat

```
{
    "ServiceID": 47498,
    "MessageID": "11111111111111111111111111111111",
    "Hash": "aeaa52d56d9f0f4906e9affc7719911584c0eae2c88178ffd1e20771db0fe07",
    "Currencies": "PLN, EUR"
}
```

Odpowiedź na żądanie

W odpowiedzi na żądanie zwracana jest (w tej samej sesji HTTP) lista, zawierająca kolejne Kanały płatności (**gatewayID**), ich nazwę, typ, opis, adres logotypu oraz informację o stanie oraz aktualności statusu (będzie on odświeżany co kilka minut).

Poniżej opis tych pól:

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	result	TAK	string{1,5}	Status odpowiedzi. Dopuszczalne wartości: - OK - ERROR
2	errorStatus	TAK	string{1,100}	Status błędu, wypełniany w przypadku błędu (w przeciwnym wypadku null).
3	description	TAK	string{1,500}	Opis błędu, wypełniany w przypadku błędu (w przeciwnym wypadku null).
4	serviceID	TAK	string{1,10}	Identyfikator Serwisu Partnera; pochodzi z żądania metody.
5	messageID	TAK	string{32}	Identyfikator komunikatu pochodzący z żądania metody.
	gatewayList	NIE	list	Lista zawierająca kolejne kanały płatności (pusta w przypadku braku skonfigurowanych kanałów płatności).
6	gatewayID	TAK	integer{1,5}	Identyfikator Kanału Płatności, za pomocą, którego Klient może uregulować płatność.
7	gatewayName	TAK	string{1,200}	Nazwa Kanału Płatności, którą można wyświetlić na liście dostępnych banków.

kolejność Hash	nazwa	wymagany	typ	opis
8	gatewayType	NIE	string{1,30}	<p>Typ, służący przede wszystkim do grupowania Kanałów Płatności na ich liście.</p> <p>Parametr przyjmuje aktualnie poniższe wartości (ale lista ta może zostać rozszerzona):</p> <ul style="list-style-type: none"> - PBL – najliczniejsza grupa kanałów, w których autoryzacja płatności odbywa się po automatycznym przekierowaniu i uzupełnieniu danych do przelewu w bankowości elektronicznej - Szybki Przelew – uzupełnienie powyższej grupy o banki, niewspierające automatycznego uzupełniania danych do przelewu po zalogowaniu do bankowości elektronicznej - BLIK – szybkie płatności z aplikacji mobilnej - Karta płatnicza – płatność z podaniem danych kartowych (na bezpiecznej formatce kartowej AP, na natywnej formatce w aplikacji mobilnej, lub za pomocą Widget Autopay). - Portfel elektroniczny – płatność z użyciem instrumentów typu Visa, Google Pay, Apple Pay etc. - Płatność automatyczna – płatność z użyciem zapisanej w Systemie karty - Raty online – płatności z użyciem produktów ratalnych
9	bankName	NIE	string{1,32}	Nazwa banku.
10	iconURL	NIE	string{1,100}	Adres, z którego można pobrać logotyp Kanału Płatności.
11	state	TAK	string{1,64}	<p>Informacja o stanie dostępności kanału.</p> <p>Przyjmuje wartości:</p> <ul style="list-style-type: none"> - OK – kanał dostępny - TEMPORARY_DISABLED – kanał chwilowo niedostępny (np. z powodu prac po stronie banku) - DISABLED – kanał niedostępny (usługa zawieszona na dłuższy okres)
12	stateDate	NIE	string{1,19}	Moment ostatniej aktualizacji statusu Kanału Płatności; przykładowa wartość: 2014-10-30 07:54:50. (Czas CET)

kolejność Hash	nazwa	wymagany	typ	opis
13	gatewayDescription	NIE	string{1,1000}	Opcjonalne pole opisujące kanał płatności. Można wyświetlić po jego zaznaczeniu.
14	inBalanceAllowed	NIE	boolean	Informacja czy kanał może być użyty (po uzgodnieniach biznesowych) do zasilania salda przedpłaconego (start transakcji z użyciem parametru TransactionSettlementMode=NONE).
15	minValidityTime	NIE	integer	Minimalny czas ważności transakcji w minutach. Pojawia się dla kanałów gdzie ustalenie statusu płatności trwa dłużej niż zwykle.
	currencyList	TAK	lista	Lista zawierająca waluty dostępne dla kanału płatności, wraz z ograniczeniami kwot.
16	currency	TAK	string{3}	Waluta, którą można opłacić tym kanałem. W przypadku dostępności dla danego kanału płatności w wielu walutach, lista będzie zawierać więcej niż jeden element. Dopuszczalne jedynie wartości: PLN, EUR, GBP oraz USD. Do liczenia wartości Hash pobierane są wartości kolejnych węzłów currencyList .
17	minAmount	NIE	amount	Minimalna kwota transakcji, którą można opłacić tym kanałem. Jako separator dziesiętny używana jest kropka - '.' Format: 0.00; maksymalna długość: 14 cyfr przed kropką i 2 po kropce. Pole występuje tylko dla niektórych kanałów, wartość jest wyrażona w walucie pola currency . Do liczenia wartości Hash pobierane są wartości kolejnych węzłów currencyList .
18	maxAmount	NIE	amount	Maksymalna kwota transakcji, którą można opłacić tym kanałem. Jako separator dziesiętny używana jest kropka - '.' Format: 0.00; maksymalna długość: 14 cyfr przed kropką i 2 po kropce. Pole występuje tylko dla niektórych kanałów, wartość jest wyrażona w walucie pola currency . Do liczenia wartości Hash pobierane są wartości kolejnych węzłów currencyList .


```

    },
    "hash" :
    "d3ffb125bd1effeb27f46af7fcbada36fb1b9df0ff81438db98ecd06f502c8b9" ,
  }

```

UWAGA: Wynik odpytania metody powinno się co odświeżać kilka minut (zbyt częste wołanie **gatewayList** zwiększy obciążenie obu systemów, nie wnosząc wiele korzyści). W przypadku braku lub nieprawidłowej odpowiedzi, należy wyświetlić ostatnią znaną i poprawną konfigurację Kanałów Płatności. Jest to drugi powód do przechowywania tymczasowej kopii **gatewayList** w Serwisie Partnera. Jako nieprawidłową odpowiedź należy traktować pustą odpowiedź, timeout, nieprawidłowy Hash, bądź pustą listę węzłów **gateway**.

Odpytywanie o listę aktualnie dostępnych zgód formalnych

Opis

Opis integracji umożliwiający używanie listy płatności osadzonej w serwisie (lub aplikacji mobilnej), bez kroków przejściowych. W niektórych przypadkach zamiast kroku przejściowego, standardowe zachowanie systemu przewiduje blokadę startu transakcji.

Należy wyświetlić odpowiednie treści formalne (a więc klauzule informacyjne oraz ew. regulaminy) już w momencie wyboru formy płatności, a następnie przekazać do Systemu Płatności Online potwierdzenie ich wyświetlenia oraz ew. akceptacji (w postaci identyfikatorów).

System umożliwia zdalne odpytanie o aktualną listę obowiązków i powiązanych treści formalnych. W tym celu należy wywołać metodę **legalData** (https://host_bramki/legalData) z odpowiednimi parametrami (w formacie JSON).

WSKAZÓWKA: Wszystkie parametry przekazywane są za pomocą technologii REST. Protokół rozróżnia wielkość liter zarówno w nazwach jak i wartościach parametrów. Wartości przekazywanych parametrów powinny być kodowane w UTF-8.

Lista dostępnych parametrów

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	ServiceID	TAK	integer	Identyfikator Serwisu Partnera.
2	MessageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łacińskiego (np. na bazie UID). Wartość pola musi być unikalna dla Serwisu Partnera.

kolejność Hash	nazwa	wymagany	typ	opis
3	description	TAK	string{1,500}	Opis błędu, wypełniany w przypadku błędu. W przeciwnym wypadku null.
4	serviceID	TAK	string{1,10}	Identyfikator Serwisu Partnera; pochodzi z żądania metody.
5	messageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łacińskiego (np. na bazie UID). Pochodzi z żądania metody.
6	gatewayID	TAK	integer{1,5}	Identyfikator Kanału Płatności, za pomocą którego Klient może uregulować płatność.
7	language	TAK	string{2}	Język, w jakim System zwraca treści (klauzule i regulaminy).
8	serviceModel	TAK	string{1,20}	Pole oznaczające model, w którym pracuje serwis, na potrzeby ew. przyszłych wytycznych w oparciu o te wartości (aktualnie o wartościach: MERCHANT, PAYER). W tym momencie powinno być ignorowane.
nd.	regulationList	TAK	list	Lista zawierająca treści formalne dostępne dla kanału płatności.
9	regulationID	TAK	integer{1,10}	Identyfikator treści formalnej, który (w przypadku jego akceptacji przez Klienta) powinien być przekazany w parametrze startowym DefaultRegulationAcceptanceID , lub RecurringAcceptanceID (odpowiednio dla typu DEFAULT i RECURRING). Sposób akceptacji określają pola showCheckbox i isCheckboxRequired . UWAGA: Ta wartość może się powtarzać dla wywołań o różnych GatewayID , gdyż regulaminy są przyporządkowane raczej grupie kanałów płatności, a nie pojedynczym kanałom.

kolejność Hash	nazwa	wymagany	typ	opis
10	type	TAK	string{1,64}	<p>Typ obowiązku formalnego. Przewidziane wartości:</p> <ul style="list-style-type: none"> - DEFAULT – klauzula (lub klauzule) oraz regulamin płatności w modelu usługi świadczonej przez AP na rzecz Klienta - RECURRING – klauzule (lub klauzule) oraz regulamin płatności automatycznej. Wartość dostępna tylko, jeśli skonfigurowana jest usługa płatności automatycznej - PSD2 – klauzula dedykowana kanałom typu PSD2 (w tej chwili wartość nie jest używana) - RODO – klauzula informacyjna dotycząca przetwarzania danych osobowych - PRIVACY – klauzula informacyjna dotycząca polityki prywatności
11	url	NIE	string{1,500}	<p>Adres do pliku z regulaminem (do samodzielnego osadzenia w Serwisie). Standardowo, jeśli tak stanowi obowiązek formalny, powinien być częścią jednej z jego klauzul, tj. pola inputLabel.</p> <p>WSKAZÓWKA: <i>Pojawia się w przypadku wystąpienia dokumentu powiązanego ze zgodą.</i></p>
nd.	labelList	TAK	list	<p>Lista zawierająca klauzule dostępne dla danego obowiązku formalnego. Obowiązek ten może wymagać wyświetlenia jednej lub więcej treści.</p>
12	labelID	TAK	integer{1,10}	<p>Identyfikator klauzuli, przekazywane na potrzeby diagnostyczne (może być przez Partnera ignorowany).</p>
13	inputLabel	TAK	string{1,500}	<p>Treść klauzuli do wyświetlenia w Serwisie w powiązaniu z odpowiednim regulationID. W niektórych przypadkach może zawierać link do regulaminu.</p>


```

<a class="privacy-policy" href="https://{host_bramki}/polityka-prywatnosci.pdf"
target="_blank">Politykę prywatności</a></li><li>\r\nChcę aby usługa została
zrealizowana niezwłocznie, a w przypadku odstąpienia od umowy, wiem, że nie otrzymam
zwrotu poniesionych kosztów za usługi zrealizowane na moje żądanie do chwili odstąpienia
od umowy\r\n</li></ul>",
        "placement": "ABOVE_BUTTON",
        "showCheckbox": true,
        "checkboxRequired": true
    }
    ],
    {
        "regulationID": 1,
        "type": "PRIVACY",
        "labelList": [
            {
                "labelID": 1,
                "inputLabel": "Autopay korzysta z plików
cookie. Pozostając na tej stronie, wyrażasz zgodę na korzystanie z plików cookie zgodnie
z <a class="privacy-policy" href=?r="https://{host_bramki}/polityka-prywatnosci.pdf"
target="_blank">Polityką prywatności Autopay S.A. </a> Możesz samodzielnie zarządzać
cookies zmieniając odpowiednio ustawienia swojej przeglądarki lub oprogramowania
urządzenia."
            }
        ],
        "placement": "BOTTOM_OF_PAGE",
        "showCheckbox": false,
        "checkboxRequired": false
    }
    ],
    ],
    "hash":
"61789013d932e2bc728d6206f7e9222b93e3176f7f07f6aa8cce1ccd65afaf0d",
    "result": "OK",
    "errorStatus": null,
    "description": null
}

```

Opis obsługi odpowiedzi

Ponieważ wymogi formalne dotyczące treści klauzul, ich rozmieszczenie oraz sposób akceptacji zależą od stosowanego kanału płatniczego, metoda ta powinna być wywoływana każdorazowo po jego wyborze (stąd obowiązkowy parametr **GatewayID**).

Odpowiednie treści i zachowanie powinny być dynamicznie dostosowywane do odpowiedzi z Systemu (np. powinien pojawić się wymagany checkbox z klauzulą informacyjną oraz linkiem do regulaminu). Oczywiście, aby aplikacja działała szybko, mile widziane jest używanie cache do zapamiętywania odpowiedzi niedawno wykonanych wywołań (np. na 15 minut).

Wyświetlona (i ew. zatwierdzona) w momencie przejścia do płatności zgoda, powinna być potwierdzona w Systemie poprzez dołączenie do komunikatu startu transakcji w parametrze startowym jej identyfikatora (a więc odpowiednią wartością **regulationID**).

W zależności od wartości pola **type** regulaminu:

- dla wyświetlanej/akceptowanej klauzuli o **type=DEFAULT**:

- a. do parametru **DefaultRegulationAcceptanceID** powinna trafić jej wartość **regulationID**;
- b. do parametru **DefaultRegulationAcceptanceState** powinna trafić wartość **ACCEPTED** oraz
- c. do parametru **DefaultRegulationAcceptanceTime** powinna trafić wartość odpowiadająca chwili akceptacji zgody poprzez zaznaczenie checkboxa oraz kliknięcie przycisku „Rozpocznij płatność”

- dla wyświetlanej/akceptowanej klauzuli o **type=RECURRING**:

- a. do parametru **RecurringAcceptanceID** powinna trafić jej wartość **regulationID**;
- b. do parametru **RecurringAcceptanceState** powinna trafić wartość **ACCEPTED** oraz
- c. do parametru **RecurringAcceptanceTime** powinna trafić wartość odpowiadająca chwili akceptacji zgody poprzez zaznaczenie checkboxa oraz kliknięcie przycisku „Rozpocznij płatność”

UWAGA: Pola (np. **serviceModel**, **url**, **labelID**) i wartości pól (np. **PSD2**, **RODO**, **PRIVACY**) metody **legalData** nie są wymagane do obsługi, ale należy przewidzieć możliwość ich występowania w odpowiedzi na żądanie.

Odpytanie o stan salda

Opis

Dla wszystkich serwisów możliwe jest odpytanie o aktualny stan salda. W tym celu należy wywołać metodę **balanceGet** https://{host_bramki}/webapi/balanceGet z odpowiednimi parametrami. Wszystkie parametry przekazywane są metodą POST (Content-Type: application/x-www-form-urlencoded). Protokół rozróżnia wielkość liter zarówno w nazwach jak i wartościach parametrów. Wartości przekazywanych parametrów powinny być kodowane w UTF-8.

Lista dostępnych parametrów dla aktualnego stanu salda

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	ServiceID	TAK	string{1,10}	Identyfikator Serwisu Partnera.
1	BalancePointID	TAK	string{1,10}	Identyfikator Punktu Rozliczeń. UWAGA: Wymagane jedno z pól ServiceID lub BalancePointID . Podanie obu spowoduje zatrzymanie przetwarzania żądania oraz błąd http.

kolejność Hash	nazwa	wymagany	typ	opis
2	MessageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łacińskiego (np. na bazie UID). Wartość pola musi być unikalna dla Serwisu Partnera.
3	PlenipotentiaryID	NIE	string{8,8}	Identyfikator pełnomocnika. Jeżeli jest obecny to do obliczania Hash jest używany klucz współdzielony pełnomocnika, a nie główny klucz serwisu/punktu rozliczeń. Wpływa też na Hash w odpowiedzi na ten komunikat. WAŻNE! <i>Użycie tego pola wymaga specjalnych uzgodnień biznesowych.</i>
nd.	Hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Stosowany jest klucz współdzielony przypisany do zastosowanego identyfikatora konfiguracji (Serwisu, bądź Punktu Rozliczeń). Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis.

Odpowiedź na żądanie aktualnego salda

W odpowiedzi na żądanie zwracany jest (w tej samej sesji HTTP) tekst w formacie XML, zawierający potwierdzenie przyjęcia operacji do realizacji lub opis błędu.

Struktura potwierdzenia (XML)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <balanceGet>
    <serviceID>ServiceID</serviceID>
    <messageID>MessageID</messageID>
    <balance>Balance</balance>
    <currency>Currency</currency>
    <hash>Hash</hash>
  </balanceGet>
```

lub

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <balanceGet>
    <balancePointID>BalancePointID</balancePointID>
    <messageID>MessageID</messageID>
    <balance>Balance</balance>
```

```

<currency>Currency</currency>
<hash>Hash</hash>
</balanceGet>

```

Lista pól odpowiedzi

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceID	TAK	string{1,10}	Identyfikator Serwisu Partnera. Pochodzi z żądania metody.
1	balancePointID	TAK	string{1,10}	Identyfikator Punktu Rozliczeń. Pochodzi z żądania metody. UWAGA: Zwrócone zostanie jedno z pól ServiceID lub BalancePointID.
2	messageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łacińskiego (np. na bazie UID). Pochodzi z żądania metody.
3	balance	TAK	amount	Wartość salda; jako separator dziesiętny używana jest kropka - '.' Format: 0.00.
4	currency	TAK	string{1,3}	Waluta salda. Dopuszczalne jedynie wartości: PLN, EUR, GBP oraz USD.
nd.	hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Stosowany jest klucz współdzielony przypisany do zastosowanego identyfikatora konfiguracji (Serwisu, bądź Punktu Rozliczeń). Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis Partnera.

Zasilanie salda

Opis

Dla serwisów posiadających saldo w Systemie, możliwe jest wykonanie operacji zasilenia salda na poczet przyszłych zwrotów. W tym celu należy wykonać start transakcji z parametrem **TransactionSettlementMode** o wartości **NONE** oraz **Amount** wskazującym **kwotę zasilenia**.

Skorzystanie z **Przedtransakcji** pozwoli na proste dostarczenie gotowej transakcji do płatnika (np. podając w CustomerEmail adres księgowości Partnera).

UWAGA: Usługa musi zostać uzgodniona z opiekunem biznesowym. W modelu Marketplace konieczne będzie ponadto zbudowanie koszyka produktów wskazującego jaki Punkt Rozliczeń (BalancePointID) ma zostać zasilony.

Wypłaty z salda

Opis

Dla serwisów posiadających saldo w Systemie, możliwe jest wykonanie operacji wypłaty całości bądź części salda na zdefiniowany rachunek do rozliczeń. W tym celu należy wywołać metodę *balancePayoff* (https://{host_bramki}/settlementapi/balancePayoff) z odpowiednimi parametrami.

Wszystkie parametry przekazywane są metodą POST (Content-Type: application/x-www-form-urlencoded). Protokół rozróżnia wielkość liter zarówno w nazwach jak i wartościach parametrów. Wartości przekazywanych parametrów powinny być kodowane w UTF-8.

Lista dostępnych parametrów dla wypłat z salda

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

UWAGA: Wymagane jedno z pól **ServiceID** lub **BalancePointID**.
Podanie obu spowoduje zatrzymanie przetwarzania żądania oraz błąd http.

kolejność Hash	nazwa	wymagany	typ	opis
1	ServiceID	TAK	string{1,10}	Identyfikator Serwisu Partnera.
1	BalancePointID	TAK	string{1,10}	Identyfikator Punktu Rozliczeń.
2	MessageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łacińskiego (np. na bazie UID), wartość pola musi być unikalna i wskazywać konkretne zlecenie wypłaty w Serwisie Partnera. Weryfikacja unikalności po stronie Systemu pozwala na ponawianie MessageID w przypadku problemów z komunikacją (ponowienie tej wartości skutkować będzie potwierdzeniem zlecenia, bez ponownego wykonania w Systemie).

kolejność Hash	nazwa	wymagany	typ	opis
3	Amount	NIE	amount	Kwota wypłaty z salda (nie może być większa niż aktualne saldo serwisu); niepodanie tego parametru skutkuje wypłatą całości środków zgromadzonych na saldzie; jako separator dziesiętny używana jest kropka - '.' Format: 0.00.
4	Currency	NIE	string{1,3}	Waluta wypłaty. Domyślną walutą jest PLN (użycie innej waluty musi być uzgodnione w trakcie integracji). W ramach ServiceID obsługiwana jest jedna waluta. Dopuszczalne jedynie wartości: PLN, EUR, GBP oraz USD.
5	CustomerNRB	NIE	string{26}	Numer rachunku, na który ma być wykonana wypłata z salda. Domyślnie konfiguracja wypłat nie zezwala na definiowanie tej wartości w żądaniu metody balancePayoff . Należy takie zapotrzebowanie zgłosić w trakcie integracji. UWAGA: W niektórych modelach użycie tego pola może następować jedynie w przypadku żądań z listy zaufanych IP oraz zastosowaniu jednego z 3 dodatkowych elementów: - zabezpieczenie komunikacji certyfikatem klienckim lub - zabezpieczenie komunikacji tunelem IPsec lub - użycie wartości parametru CustomerNRB z listy zaufanych rachunków Niespełnienie ich kończy się błędem CUSTOMER_NRB_NOT_AUTHORIZED . Administratorzy panelu Systemu (https://portal.autopay.eu/admin) mogą samodzielnie aktualizować listy zaufanych IP i NRB w zakładce Kontrola dostępu konfiguracji serwisu. Dopuszczalne tylko cyfry. Jeśli w trakcie integracji ustalono wykorzystanie rachunków spoza Polski, wtedy pole przesyła IBAN i oczekiwany zakres danych pola zmienia się na: alfanumeryczne znaki alfabetu łacińskiego (min. 15, max. 32 znaki).

kolejność Hash	nazwa	wymagany	typ	opis
6	SwiftCode	NIE	string{8,11}	Kod swift odpowiadający podanemu numerowi rachunku. Dopuszczalne tylko cyfry. Parametr podawany, jeśli w trakcie integracji ustalono wykorzystanie rachunków spoza Polski.
7	ForeignTransferMode	NIE	string{4,5}	System jakim ma zostać wykonany zagraniczny przelew rozliczeniowy: - SEPA (Single Euro Payments Area) - możliwy do wykonania przelewu w walucie Euro w obrębie państw członkowskich Unii Europejskiej, jak i innych państwach na terenie Starego Kontynentu, np. Islandii, Liechtensteinu, Norwegii, Szwajcarii, Monako czy Andory, - SWIFT – przelewy zagraniczne niemożliwe do wykonania za pomocą SEPA (np. inna waluta niż Euro. Opcja ta wiąże się z wyższymi kosztami wykonania przelewu niż w przypadku SEPA. Dopuszczalne wartości: SEPA i SWIFT . Parametr podawany, jeśli w trakcie integracji ustalono wykorzystanie rachunków spoza Polski.
8	ReceiverName	NIE	string{35}	Nazwa odbiorcy wypłaty z salda. Domyślnie konfiguracja wypłat nie zezwala na definiowanie tej wartości w żądaniu metody balancePayoff . Należy takie zapotrzebowanie zgłosić w trakcie integracji. Dopuszczalne alfanumeryczne znaki alfabetu łacińskiego oraz znaki z zakresu: ĘęÓóĄąŚśłłŻżŻżĆćŃń\\$.-./!()=[]{}::?;

kolejność Hash	nazwa	wymagany	typ	opis
9	Title	NIE	string{32}	<p>Tytuł wypłaty z salda. Domyślnie konfiguracja wypłat nie zezwala na definiowanie tej wartości w żądaniu metody balancePayoff. Należy takie zapotrzebowanie zgłosić w trakcie integracji.</p> <p>W niektórych przypadkach, niezależnych od AP, ten tytuł może zostać samodzielnie zmodyfikowany przez Bank.</p> <p>Dopuszczalne alfanumeryczne znaki alfabetu łacińskiego oraz znaki z zakresu: <i>ĘęÓóAąŚśŁłŻżŻżĆćŃń</i> s.-./!()", gdzie znak "/" będzie podmieniany na "-" dla transakcji wychodzących.</p>
10	RemoteRefID	NIE	string{1,20}	<p>Alfanumeryczny identyfikator transakcji wejściowej nadany przez System oraz przekazywany do Partnera w komunikacie ITN transakcji wejściowej. Wartość w tym komunikacie służy wskazaniu instrumentu płatniczego (karty, rachunku etc.), który ma zostać użyty do wykonania wypłaty.</p>
11	InvoiceNumber	NIE	string{1,100}	<p>Numer dokumentu finansowego w Serwisie. W tym komunikacie wartość służy wskazaniu faktury korygującej powiązanej z wypłatą.</p>
12	PlenipotentiaryID	NIE	string{8,8}	<p>Identyfikator pełnomocnika. Jeżeli jest obecny to do obliczania Hash jest używany klucz współdzielony pełnomocnika, a nie główny klucz serwisu/punktu rozliczeń. Wpływa też na Hash w odpowiedzi na ten komunikat. WAŻNE! Użycie tego pola wymaga specjalnych uzgodnień biznesowych.</p>
nd.	Hash	TAK	string{1,128}	<p>Stosowany jest klucz współdzielony przypisany do zastosowanego identyfikatora konfiguracji (Serwisu lub Punktu Rozliczeń). Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji.</p> <p>Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis.</p>

Odpowiedź na żądanie

Po otrzymaniu żądania wypłaty z salda System wykonuje wstępną weryfikację pod kątem przesłanych w komunikacie pól i ich wartości oraz zapisuje zlecenie do wykonania. W odpowiedzi na żądanie zwracany jest (w tej samej sesji HTTP) tekst w formacie XML, zawierający potwierdzenie zapisania zlecenia w kolejce do wykonania lub opis błędu (struktura komunikatu błędu opisana w części [Komunikaty błędu](#)).

UWAGA: Potwierdzenie przyjęcia zlecenia nie jest równoznaczne z faktycznym jego wykonaniem. Wypłata z salda może być realizowana do 30 minut od momentu wysyłki żądania wypłaty i nie zawsze może się zakończyć sukcesem. W przypadku problemów napotkanych podczas procesowania wypłaty z salda (np. niewystarczające środki na saldzie) następnego dnia roboczego wysyłany jest raport zawierający informację o wypłatach z salda, które nie doszły do skutku.

Struktura potwierdzenia (XML)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <balancePayoff>
    <serviceID>ServiceID</serviceID>
    <messageID>MessageID</messageID>
    <hash>Hash</hash>
  </balancePayoff>
```

lub

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <balancePayoff>
    <balancePointID>BalancePointID</balancePointID>
    <messageID>MessageID</messageID>
    <hash>Hash</hash>
  </balancePayoff>
```

Opis pól

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceID	TAK	string{1,10}	Identyfikator Serwisu Partnera. Pochodzi z żądania metody.
1	balancePointID	TAK	string{1,10}	Identyfikator Punktu Rozliczeń. Pochodzi z żądania metody. UWAGA: Zwrócone zostanie jedno z pól ServiceID lub BalancePointID .

kolejność Hash	nazwa	wymagany	typ	opis
2	messageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łacińskiego (np. na bazie UID). Pochodzi z żądania metody.
nd.	hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis Partnera.

UWAGA: Za błąd uznać można wszystkie odpowiedzi inne niż założona (tj. z niepoprawnymi polami, w szczególności pustym lub niepoprawnym **hash**). W przypadku, gdy System zautoryzuje nadawcę komunikatu zwrotu, jednak nie uda się wykonać operacji, odpowiedź będzie zgodna ze schematem opisanym w części [Komunikaty błędów](#). W przypadku błędu komunikacji lub braku wystarczającego salda (np. ON_DEMAND_ERROR) można ponowić zlecenie. W przypadku blokady salda (BALANCE_DISABLED) lub nieaktywnej konfiguracji (PARTNER_DISABLED) nie należy ponawiać zlecenia.

WAŻNE! W przypadku błędu zwróconego w odpowiedzi na żądanie wypłaty z salda można ponowić zlecenie, należy jednak pamiętać o tym, że każde zlecenie wypłaty nie kończące się błędem może zostać wykonane. W przypadku problemów z połączeniem, przekroczenia maksymalnego czasu oczekiwania na odpowiedź, żądanie może zostać ponowione z tym samym MessageID bez obawy o zduplikowanie zlecenia. W przypadku blokady salda (BALANCE_DISABLED) lub nieaktywnej konfiguracji (PARTNER_DISABLED) nie należy ponawiać zlecenia. 3 błędy pod rząd (bez względu na przyczynę) powodują blokadę usługi wypłat na żądanie dla określonego IP nadawcy komunikatu na 10 minut - wywołania w tym czasie dla tego IP będą kończyć się błędem TEMPORARY_DISABLED.

Zwroty transakcji

Opis

Dla serwisów posiadających saldo w Systemie, możliwe jest wykonanie operacji zwrotu do Klienta całości bądź części kwoty wpłaconej na rzecz wskazanej transakcji. Skuteczny zwrot całości transakcji można wykonać jeden raz (w przypadku ponownej próby zlecenia zwrotu tej samej transakcji, System zwraca odpowiednio opisany błąd). Zwroty części kwoty transakcji można na niej wykonywać wiele razy, o ile ich suma nie przekroczy kwoty wpłaty.

Aby wykonać zwrot transakcji, należy wywołać metodę **transactionRefund** (https://{host_bramki}/settlementapi/transactionRefund) z odpowiednimi parametrami. Wszystkie parametry przekazywane są metodą POST (Content-Type: application/x-www-form-urlencoded).

Protokół rozróżnia wielkość liter zarówno w nazwach jak i wartościach parametrów. Wartości przekazywanych parametrów powinny być kodowane w UTF-8.

Lista dostępnych parametrów

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	ServiceID	TAK	string{1,10}	Identyfikator Serwisu Partnera.
2	MessageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łańciskiego (np. na bazie UID), wartość pola musi być unikalna i wskazywać konkretne zlecenie wypłaty w Serwisie Partnera. Weryfikacja unikalności po stronie Systemu pozwala na ponawianie MessageID w przypadku problemów z komunikacją (ponowienie tej wartości skutkować będzie potwierdzeniem zlecenia, bez ponownego wykonania w Systemie).
3	RemotelD	TAK	string{1,20}	Alfanumeryczny identyfikator zwracanej transakcji wejściowej nadany przez System oraz przekazywany do Partnera w komunikacie ITN transakcji wejściowej.
4	Amount	NIE	amount	Kwota wypłaty z salda (nie może być większa niż aktualne saldo serwisu); niepodanie tego parametru skutkuje wypłatą całości środków zgromadzonych na saldzie; jako separator dziesiętny używana jest kropka - '.' Format: 0.00. W modelu Marketplace, pole musi być puste (zwrot całkowity). W przeciwnym wypadku nie byłoby możliwości wskazania punktu/ów rozliczeń, do obciążenia za taką operację. Przy zwrocie całkowitym saldo jest potrącane zgodnie z sumą kwot produktów danego punktu rozliczeń.
5	Currency	NIE	string{1,3}	Waluta wypłaty. Domyślną walutą jest PLN (użycie innej waluty musi być uzgodnione w trakcie integracji). W ramach ServiceID obsługiwana jest jedna waluta. Dopuszczalne jedynie wartości: PLN, EUR, GBP oraz USD.
nd.	Hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis Partnera.

W odpowiedzi na żądanie zwracany jest (w tej samej sesji HTTP) tekst w formacie XML, zawierający potwierdzenie wykonania operacji lub opis błędu (opis poniżej).

Struktura potwierdzenia (XML)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <transactionRefund>
    <serviceID>ServiceID</serviceID>
    <messageID>MessageID</messageID>
    <hash>Hash</hash>
  </transactionRefund>
```

Opis pól

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceID	TAK	string{1,10}	Identyfikator Serwisu Partnera. Pochodzi z żądania metody.
2	messageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łacińskiego (np. na bazie UID). Pochodzi z żądania metody.
nd.	hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis Partnera.

Opcja wykonywania zwrotów do opłaconych transakcji możliwa jest do 12 miesięcy wstecz, licząc od daty rozpoczęcia transakcji. Wyjątek stanowią płatności BLIK, które ze względu na ograniczenia czasowe po stronie dostawcy Kanału płatności, można zwracać do 6 miesięcy wstecz.

Powyższe terminy dotyczą realizowania zwrotów poprzez panel administracyjny oraz **transactionRefund**, np. za pośrednictwem własnych narzędzi administracyjnych. Po ich przekroczeniu zostanie zwrócony błąd (**TRANSACTION_TOO_OLD_TO_REFUND**). Schemat działania jest analogiczny jak w przypadku wypłat z salda. To znaczy, że System przyjmuje zlecenie i asynchronicznie przetwarza je w ciągu maksymalnie 30 minut, a w przypadku niepowodzenia informacja o operacjach zakończonych błędem jest wysyłana w raporcie następnego dnia roboczego.

W przypadku problemów z połączeniem, przekroczenia maksymalnego czasu oczekiwania na odpowiedź, żądanie może zostać ponowione z tym samym MessageID bez obawy o zduplikowanie zlecenia. Za błąd uznać można wszystkie odpowiedzi inne niż założona (tj. z niepoprawnymi polami, w szczególności pustym lub niepoprawnym **hash**). W przypadku, gdy System zautoryzuje nadawcę komunikatu zwrotu, jednak nie uda się wykonać operacji, w odpowiedzi zwrócony zostanie komunikat błędu.

Zwroty produktu

Opis

Dla serwisów posiadających saldo w Systemie oraz podających w koszyku produktów parametr **productID**, możliwe jest wykonanie operacji zwrotu do Klienta całości, bądź części kwoty wpłaconej na rzecz wskazanego produktu. Skuteczny zwrot całości kwoty produktu można wykonać jeden raz (w przypadku ponownej próby zlecenia zwrotu tego samego produktu, System zwraca odpowiednio opisany błąd). Zwroty części kwoty produktu można na nim wykonywać wiele razy, o ile ich suma nie przekroczy kwoty wpłaconej na rzecz produktu.

Aby wykonać zwrot produktu, należy wywołać metodę **productRefund** (https://{host_bramki}/settlementapi/productRefund) z odpowiednimi parametrami. Wszystkie parametry przekazywane są metodą POST (Content-Type: application/x-www-form-urlencoded). Protokół rozróżnia wielkość liter zarówno w nazwach jak i wartościach parametrów. Wartości przekazywanych parametrów powinny być kodowane w UTF-8.

Lista parametrów

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	ServiceID	TAK	string{1,10}	Identyfikator Serwisu Partnera.
2	MessageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łańciskiego (np. na bazie UID), wartość pola musi być unikalna i wskazywać konkretne zlecenie wypłaty w Serwisie Partnera. Weryfikacja unikalności po stronie Systemu pozwala na ponawianie MessageID w przypadku problemów z komunikacją (ponowienie tej wartości skutkować będzie potwierdzeniem zlecenia, bez ponownego wykonania w Systemie).
3	RemotelD	TAK	string{1,20}	Alfanumeryczny identyfikator zwracanej transakcji wejściowej nadany przez System oraz przekazywany do Partnera w komunikacie ITN transakcji wejściowej.
4	ProductID	TAK	string{1,36}	Identyfikator zwracanego produktu.
5	Amount	NIE	amount	Kwota zwrotu (nie może być większa niż kwota produktu oraz aktualne saldo serwisu + ew. kwota prowizji za zwrot). Niepodanie tego parametru skutkuje zwrotem do Klienta całości środków wpłaconych na rzecz zwracanego produktu; jako separator dziesiętny używana jest kropka - '.' Format: 0.00.

kolejność Hash	nazwa	wymagany	typ	opis
6	Currency	NIE	string{1,3}	Waluta wypłaty. Domyślną walutą jest PLN (użycie innej waluty musi być uzgodnione w trakcie integracji). W ramach ServiceID obsługiwana jest jedna waluta. Dopuszczalne jedynie wartości: PLN, EUR, GBP oraz USD.
nd.	Hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis Partnera.

Odpowiedź na żądanie

W odpowiedzi na żądanie zwracany jest (w tej samej sesji HTTP) tekst w formacie XML, zawierający potwierdzenie wykonania operacji lub opis błędu (opis poniżej).

Struktura potwierdzenia (XML)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <productRefund>
    <serviceID>ServiceID</serviceID>
    <messageID>MessageID</messageID>
    <hash>Hash</hash>
  </productRefund>
```

Opis pól

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceID	TAK	string{1,10}	Identyfikator Serwisu Partnera. Pochodzi z żądania metody.
2	messageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łacińskiego (np. na bazie UID). Pochodzi z żądania metody.
nd.	hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis Partnera.

Opcja wykonywania zwrotów do opłaconych transakcji możliwa jest do 12 miesięcy wstecz, licząc od daty rozpoczęcia transakcji. Wyjątek stanowią płatności BLIK, które ze względu na ograniczenia czasowe po stronie dostawcy Kanału płatności, można zwracać do 6 miesięcy wstecz.

Powyższe terminy dotyczą realizowania zwrotów poprzez panel administracyjny oraz **productRefund**, np. za pośrednictwem własnych narzędzi administracyjnych. Po ich przekroczeniu zostanie zwrócony błąd (**TRANSACTION_TOO_OLD_TO_REFUND**). Schemat działania jest analogiczny jak w przypadku wypłat z salda. To znaczy, że System przyjmuje zlecenie i asynchronicznie przetwarza je w ciągu maksymalnie 30 minut, a w przypadku niepowodzenia informacja o operacjach zakończonych błędem jest wysyłana w raporcie następnego dnia roboczego.

W przypadku problemów z połączeniem, przekroczenia maksymalnego czasu oczekiwania na odpowiedź, żądanie może zostać ponowione z tym samym MessageID bez obawy o zduplikowanie zlecenia. Za błąd uznać można wszystkie odpowiedzi inne niż założona (tj. z niepoprawnymi polami, w szczególności pustym lub niepoprawnym **hash**). W przypadku, gdy System zautoryzuje nadawcę komunikatu zwrotu, jednak nie uda się wykonać operacji, w odpowiedzi zwrócony zostanie komunikat błędu (Zob. [Komunikaty błędów](#)).

Odpytanie o status zwrotu lub wypłaty z salda

Opis

Po wykonaniu zwrotu lub wypłaty z salda mamy możliwość weryfikacji w jakim statusie jest transakcja rozliczeniowa oraz jaki został jej nadany identyfikator przez System płatności online. W tym celu należy wywołać metodę *outDetails* (https://{host_bramki}/settlementapi/outDetails) z odpowiednimi parametrami.

Wszystkie parametry przekazywane są metodą POST (Content-Type: application/x-www-form-urlencoded). Protokół rozróżnia wielkość liter zarówno w nazwach jak i wartościach parametrów. Wartości przekazywanych parametrów powinny być kodowane w UTF-8.

Lista dostępnych parametrów dla wypłat z salda

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

UWAGA: Wymagane jedno z pól **ServiceID** lub **BalancePointID**.

Podanie obu spowoduje zatrzymanie przetwarzania żądania oraz błąd http.

kolejność Hash	nazwa	wymagany	typ	opis
1	ServiceID	TAK	string{1,10}	Identyfikator Serwisu Partnera.
1	BalancePointID	TAK	string{1,10}	Identyfikator Punktu Rozliczeń.
2	MessageID	TAK	string{32}	Należy podać ten sam identyfikator komunikatu, który był wysłany poprzednio przy zleceniu zwrotu lub wypłaty z salda.

kolejność Hash	nazwa	wymagany	typ	opis
3	Method	TAK	enum	Operacja dla której powstała transakcja rozliczeniowa: BALANCE_PAYOFF - wypłata z salda TRANSACTION_REFUND - zwrot transakcji PRODUCT_REFUND - zwrot produktu
nd.	Hash	TAK	string{1,128}	Stosowany jest klucz współdzielony przypisany do zastosowanego identyfikatora konfiguracji (Serwisu lub Punktu Rozliczeń). Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis.

Odpowiedź na żądanie

Struktura potwierdzenia (XML)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <outDetails>
    <serviceID>ServiceID</serviceID>
    <messageID>MessageID</messageID>
    <status>Status</status>
    <remoteOutId>RemoteOutId</remoteOutId>
    <hash>Hash</hash>
  </outDetails>
```

lub

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <outDetails>
    <balancePointID>BalancePointID</balancePointID>
    <messageID>MessageID</messageID>
    <status>Status</status>
    <remoteOutId>RemoteOutId</remoteOutId>
    <hash>Hash</hash>
  </outDetails>
```

Opis pól

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceID	TAK	string{1,10}	Identyfikator Serwisu Partnera. Pochodzi z żądania metody.
1	balancePointID	TAK	string{1,10}	Identyfikator Punktu Rozliczeń. Pochodzi z żądania metody. UWAGA: Zwrócone zostanie jedno z pól ServiceID lub BalancePointID.
2	messageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łacińskiego (np. na bazie UID). Pochodzi z żądania metody.
3	status	TAK	enum	Status przetwarzania: NEW - Oczekuje w kolejce na przetworzenie. PROCESSING - W trakcie procesowania ERROR - Przetwarzanie zakończyło się niepowodzeniem np. nie było środków na saldzie DONE - Przetwarzanie zakończyło się sukcesem.
4	remoteOutId	NIE	string{1,20}	Alfanumeryczny identyfikator transakcji rozliczeniowej nadany przez System płatności online.
nd.	hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis Partnera.

Strona podsumowania transakcji

Opis

System umożliwia wyświetlanie Klientowi podsumowania transakcji. W tym celu Partner może budować linki uruchamiające z odpowiednimi parametrami metodę **confirmation** (https://{host_bramki}/confirmation/payment). Wszystkie parametry przekazywane są metodą GET. Protokół rozróżnia wielkość liter zarówno w nazwach jak i wartościach parametrów. Wartości przekazywanych parametrów powinny być kodowane w UTF-8.

Lista parametrów dla metody podsumowania transakcji

Przekierowanie Klienta z poprawnymi parametrami spowoduje wyświetlenie podsumowania transakcji (o treści zależnej od jej stanu) lub informacji o jej braku (jeśli System nie jej nie odnajdzie).

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	ServiceID	TAK	string{1,10}	Identyfikator Serwisu Partnera.
2	OrderID	TAK	string{32}	Identyfikator transakcji nadany w Serwisie Partnera i przekazany w starcie transakcji.
nd.	Hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis.

UWAGA: Usługa musi zostać aktywowana po uzgodnieniu z opiekunem biznesowym. Istnieje możliwość zmiany treści komunikatów lub dostosowanie szaty graficznej (podlegają one każdorazowo ustaleniom w formie roboczej podczas integracji).

Odpytanie o status transakcji

Opis

Dla wszystkich serwisów możliwe jest odpytanie o status transakcji. W tym celu należy wywołać metodę **transactionStatus** (https://{host_bramki}/webapi/transactionStatus) z odpowiednimi parametrami.

Wszystkie parametry przekazywane są metodą POST (Content-Type: application/x-www-form-urlencoded). Protokół rozróżnia wielkość liter zarówno w nazwach jak i wartościach parametrów. Wartości przekazywanych parametrów powinny być kodowane w UTF-8.

Lista parametrów dostępnych dla statusu transakcji

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	ServiceID	TAK	string{1,10}	Identyfikator Serwisu Partnera.
2	OrderID	TAK	string{32}	Identyfikator transakcji nadany w Serwisie Partnera i przekazany w starcie transakcji.
nd.	Hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis Partnera.

Nagłówek HTTP dla zapytania o status transakcji

Do poprawnego odpytania należy, wraz z przekazywanymi parametrami, przesłać zdefiniowany nagłówek HTTP o odpowiedniej treści. Dołączony nagłówek powinien nosić nazwę 'BmHeader' i posiadać następującą wartość 'pay-bm', w całości powinien prezentować się następująco 'BmHeader: pay-bm'. W przypadku poprawnego komunikatu zwracany jest (w tej samej sesji HTTP) tekst w formacie XML, zawierający wszystkie transakcje o wskazanym OrderID wraz z podstawowymi o nich informacjami.

WSKAZÓWKA: Sytuacja taka może mieć miejsce np. w przypadku, gdy Klient zmieni Kanał Płatności, wywoła ponownie ten sam start transakcji z historii przeglądarki itp. System umożliwia blokowanie takich przypadków, jednak opcja nie jest zalecana (nie byłoby możliwe opłacenie porzuconej transakcji).

WAŻNE! W przypadku gdy odpytanie dotyczy orderID, które występuje w ponad 50 transakcjach danego serwisu, zwracana jest odpowiedź (XML) z kodem błędu 403.

Struktura odpowiedzi w przypadku osiągniętego limitu transakcji z tym samym orderID:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<transaction>
<reason>LIMIT_REQUESTED_TRANSACTIONS_WITH_THE_SAME_ORDER_ID_AND_SERVICE_ID_EXCEEDED</reason>
  <description>Transaction limit 50 with the same order id {{ORDER_ID}} and service id {{SERVICE_ID}} exceeded. Requested count {{TRANSACTION_AMOUNT}}
  </description>
</transaction>
```

Lista pól dla zapytania o status transakcji

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceID	TAK	string{1,10}	Identyfikator Serwisu Partnera, nadawany w trakcie rejestracji usługi, jednoznacznie identyfikuje Serwis Partnera w Systemie płatności online.
2	orderID	TAK	string{1,32}	Identyfikator transakcji nadany w Serwisie Partnera i przekazany w starcie transakcji.
3	remotelID	TAK	string{1,20}	Alfanumeryczny identyfikator transakcji nadany przez System płatności online.

kolejność Hash	nazwa	wymagany	typ	opis
5	amount	TAK	amount	Kwota transakcji jako separator dziesiętny używana jest kropka - '.' Format: 0.00; maksymalna długość: 14 cyfr przed kropką i 2 po kropce.
6	currency	TAK	string{1,3}	Waluta transakcji.
7	gatewayID	NIE	string{1,5}	Identyfikator Kanału Płatności, za pomocą, którego Klient uregulował płatność.
8	paymentDate	TAK	string{14}	Moment zautoryzowania transakcji, przekazywany w formacie YYYYMMDDhhmmss. (Czas CET)
9	paymentStatus	TAK	enum	Status autoryzacji transakcji, przyjmuje wartości (opis zmian statusów dalej): PENDING - transakcja rozpoczęta. SUCCESS - poprawna autoryzacja transakcji, Serwis Partnera otrzyma środki za transakcje - można wydać towar/usługę. FAILURE - transakcja nie została zakończona poprawnie.
10	paymentStatusDetails	NIE	string{64}	Szczegółowy status transakcji, wartość może być ignorowana przez Serwis Partnera. WSKAZÓWKA: Szczegółowy opis w części Szczegółowe statusy transakcji.
nd.	hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis.

UWAGA: Ponieważ metoda może zwrócić wiele transakcji, do Hash pobierane są kolejne transakcje (zgodnie z kolejnością występowania transakcji w odpowiedzi). W ramach danej transakcji zastosowanie ma kolejność zgodna z numerem obok pola (z wyłączeniem parametru ServiceID, poziom wyżej).

Przykład łańcucha funkcji skrótu

```

Hash = funkcja(<serviceID> + „|” +
<orderID1> + „|” + <remoteID1> + „|” + <amount1> + „|” + <currency1> + „|” +
<gatewayID1> + „|” + <paymentDate1> + „|” + <paymentStatus1> + „|” +
<paymentStatusDetails1> + „|” +
<orderID2> + „|” + <remoteID2> + „|” + <amount2> + „|” + <currency2> + „|” +
<gatewayID2> + „|” + <paymentDate2> + „|” + <paymentStatus2> + „|” +
<paymentStatusDetails2> + ...

```

Obsługa odpowiedzi zapytania o status transakcji - propozycja obsługi wielu transakcji w odpowiedzi

Warunki	Znaczenie	Proponowany komunikat dla Klienta
Dokładnie jedna transakcja o statusie paymentstatus=SUCCESS	Poprawnie opłacona transakcja.	System poprawnie zarejestrował płatność.
Więcej niż jedna transakcja o statusie paymentstatus=SUCCESS	Wielokrotnie opłacona transakcja.	System zarejestrował więcej niż jedną płatność.
Istnieje RemoteID o statusie paymentstatus=PENDING i nie istnieje o statusie paymentstatus=SUCCESS	Transakcja oczekuje na opłacenie.	System oczekuje na płatność.
Istnieje przynajmniej jedna transakcja, ale nie ma innych statusów niż paymentstatus=FAILURE	Transakcja anulowana.	System zarejestrował rezygnację z płatności lub brak autoryzacji płatności.
Brak transakcji lub inny błąd	Nieudana próba znalezienia transakcji.	Transakcja nie została odnaleziona.

Anulowanie nieopłaconej transakcji

Opis anulowania nieopłaconej transakcji

Dla wszystkich serwisów możliwe jest anulowanie rozpoczętej, ale nieopłaconej transakcji poprzez wywołanie metody *transactionCancel* (https://{host_bramki}/webapi/transactionCancel) z odpowiednimi parametrami. Wszystkie parametry przekazywane są metodą POST (Content-Type: application/x-www-form-urlencoded).

Protokół rozróżnia wielkość liter zarówno w nazwach jak i wartościach parametrów. Wartości przekazywanych parametrów powinny być kodowane w UTF-8.

Lista parametrów dostępnych dla anulowania nieopłaconej transakcji

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	ServiceID	TAK	string{1,10}	Identyfikator Serwisu Partnera.

kolejność Hash	nazwa	wymagany	typ	opis
2	MessageID	TAK	string{32}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łacińskiego (np. na bazie UID). Wartość pola musi być unikalna dla Serwisu Partnera.
3	RemoteID	NIE	string{1,20}	Alfanumeryczny identyfikator transakcji nadany przez System oraz przekazywany do Partnera w komunikacie ITN transakcji wejściowej. Jego podanie spowoduje anulowanie tylko jednej transakcji o wskazanym RemoteID , jeśli oczekuje na wpłatę (status PENDING).
4	OrderID	NIE	string{32}	Identyfikator transakcji nadany w Serwisie Partnera i przekazany w starcie transakcji. Jego podanie (brak RemoteID) spowoduje anulowanie wszystkich transakcji oczekujących na wpłatę (status PENDING) o wskazanym OrderID (oraz ServiceID). UWAGA: Wymagane jedno z pól OrderID lub RemoteID. Podanie obu spowoduje zatrzymanie przetwarzania żądania oraz błąd http.
nd.	Hash	TAK	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis.

Nagłówek dla anulowania nieopłaconej transakcji

Do poprawnego odpytania należy, wraz z przekazywanymi parametrami, przesłać zdefiniowany nagłówek HTTP o odpowiedniej treści. Dołączony nagłówek powinien nosić nazwę 'BmHeader' i posiadać następującą wartość 'pay-bm'. W całości powinien prezentować się następująco:

```
'BmHeader: pay-bm'
```

W przypadku poprawnego komunikatu zwracany jest (w tej samej sesji HTTP) tekst w formacie XML, zawierający potwierdzenie wykonania operacji lub opis błędu.

Struktura potwierdzenia (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
  <transaction>
    <serviceID>ServiceID</serviceID>
    <messageID>MessageID</messageID>
    <confirmation>ConfStatus</confirmation>
    <reason>Reason</reason>
    <hash>Hash</hash>
```

</transaction>

Lista parametrów dla anulowania nieopłaconej transakcji

WAŻNE! Kolejność atrybutów do wyliczenia Hash musi być zgodna z ich numeracją.

kolejność Hash	nazwa	wymagany	typ	opis
1	serviceID	NIE	string{1,32}	Identyfikator Serwisu Partnera. Pochodzi z żądania metody. Wymagany dla confirmation=CONFIRMED
2	messageID	NIE	string{1,20}	Pseudolosowy identyfikator komunikatu o długości 32 znaków alfanumerycznych alfabetu łańskiego (np. na bazie UID). Pochodzi z żądania metody. Wymagany dla confirmation=CONFIRMED
3	confirmation	TAK	string{1,100}	Status potwierdzenia przyjęcia zlecenia. Może przyjmować dwie wartości: - CONFIRMED - operacja powiodła się - NOTCONFIRMED - operacja nie powiodła się
4	reason	NIE	string{1,1000}	Wyjaśnienie szczegółów przetwarzania żądania.
nd.	hash	NIE	string{1,128}	Wartość funkcji skrótu dla komunikatu obliczona zgodnie z opisem w części Bezpieczeństwo transakcji . Obowiązkowa weryfikacja zgodności wyliczonego skrótu przez Serwis. Wymagany dla confirmation=CONFIRMED

Odpowiedzi na żądania anulowania transakcji

Jeśli komunikat jest poprawny składniowo, System zwróci jedną z poniższych par opisujących wynik przetwarzania. Poza jej interpretacją, zaleca się kontrolnie odpytać o status transakcji (metoda **transactionStatus**).

Należy pamiętać, że po skutecznym anulowaniu przynajmniej jednej transakcji, nie jest możliwe wystartowanie nowej, ani kontynuowanie wcześniej wystartowanej transakcji o tym samym **OrderID**.

confirmation	reason	szczegóły
CONFIRMED	CANCELED_FULLY	Dla wskazanego OrderID: wszystkie transakcje oczekujące na wpłaty zostały anulowane. Dla wskazanego RemotelD: transakcja została anulowana.

cofnfirmation	reason	szczegóły
CONFIRMED	CANCELED_PARTIALLY	Dla wskazanego OrderID: anulowano przynajmniej jedną transakcję, jednak wystąpiły transakcje, których nie można było anulować (np. były już opłacone). Dla wskazanego RemoteID: taka odpowiedź nie występuje.
NOTCONFIRMED	INCORRECT_PAYMENT_STATUS	Znaleziono przynajmniej jedną wskazaną transakcję, jednak nie udało się żadnej anulować (np. nie było transakcji oczekującej na wpłatę).
NOTCONFIRMED	TRANSACTION_NOT_FOUND	Nie znaleziono wskazanej transakcji.
NOTCONFIRMED	OTHER_ERROR	Wystąpił inny błąd przy przetwarzaniu żądania.

Komunikaty błędu

Opis komunikatów błędu

Wszystkie komunikaty błędów będą zwracane w postaci dokumentu XML, zawierającego kod błędu, jego nazwę oraz opis. Ze względu na dużą zmienność możliwych błędów, nie jest utrzymywana pełna dokumentacja błędów.

WSKAZÓWKA: Pole **description**, dokładnie opisuje każdy z błędów (pola **statusCode** i **name** mogą być ignorowane).

Przykładowy błąd (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
  <error>
    <statusCode>55</statusCode>
    <name>BALANCE_ERROR</name>
    <description>Wrong services balance! Should be 100 but is
40</description>
  </error>
```

Schematy obsługi transakcji i rozliczeń

W tej części przedstawione są modele, scenariusze zdarzeń i przepływu informacji.

Rozszerzona struktura serwisów i punktów rozliczeń

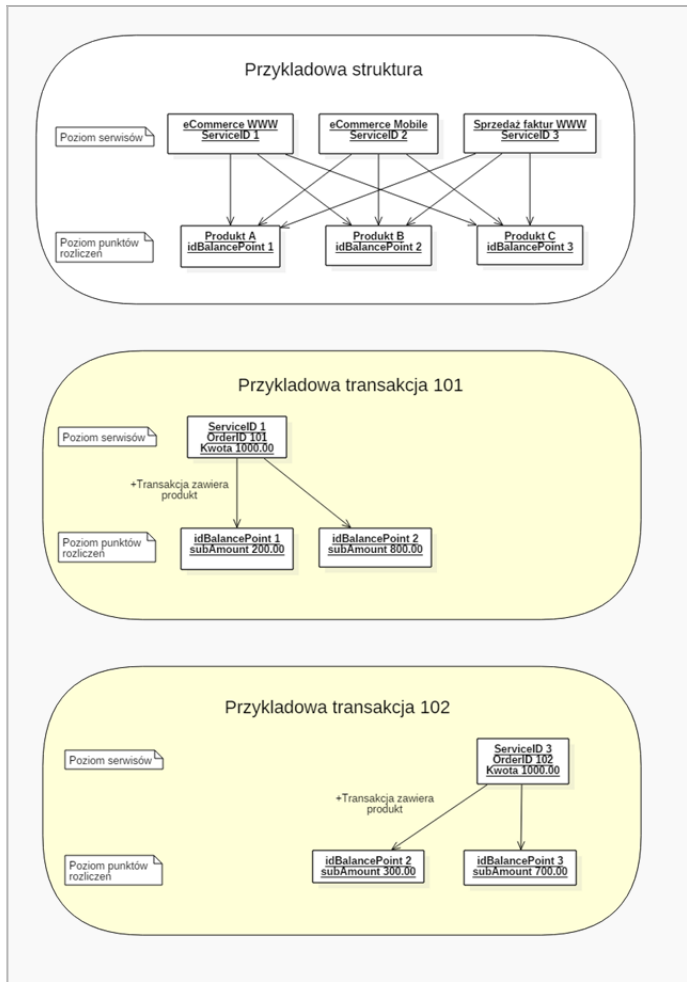
Struktura Partnera składa się z przynajmniej 1 serwisu (identyfikowane za pomocą identyfikatora **ServiceID**) oraz dowolnej liczny punktów rozliczeń (identyfikowane za pomocą identyfikatora **idBalancePoint**).

- Serwisami są zazwyczaj źródła Linków Płatności (strona WWW, aplikacja mobilna, maile itp). Serwisy też rozdzielają ruch dotyczący różnych branż (płatności za faktury, zakupy e-Commerce itp.). Ponieważ transakcja identyfikowana jest przez parę **OrderID** i **ServiceID**, można powiedzieć, że „serwis odpowiada poziomowi transakcji”.
- Punkty rozliczeń definiuje się, jeśli istnieje potrzeba rozróżnienia w jakiś sposób składowych płatności (np. poprzez wskazanie ich w raportach lub niezależne rozliczenie). Ponieważ produkt transakcji identyfikowany jest przez przynależący mu punkt rozliczeń (**idBalancePoint**), można powiedzieć, że „punkt rozliczeń odpowiada poziomowi produktu”.

Koszyk produktów (a więc również punkty rozliczeń) może nie występować w strukturze opisującej Partnera. Przyczyna dodania do struktury punktów rozliczeń wpływa na decyzję o modelu rozliczeniowym:

- potrzeba rozróżnienia składowych wpłaty na liście transakcji (w raportach) niekoniecznie musi pociągać za sobą osobnego rozliczania każdego z produktów lub punktów rozliczeń; w tej sytuacji zazwyczaj w zupełności wystarczają modele rozliczeniowe na poziomie serwisu (zbiorcze lub po każdej wpłacie)
- potrzeba rozdzielenia rozliczeń składowych wpłaty, powoduje zastosowanie modelu rozliczeniowego na poziomie punktu rozliczeń (zbiorcze lub po każdej wpłacie).

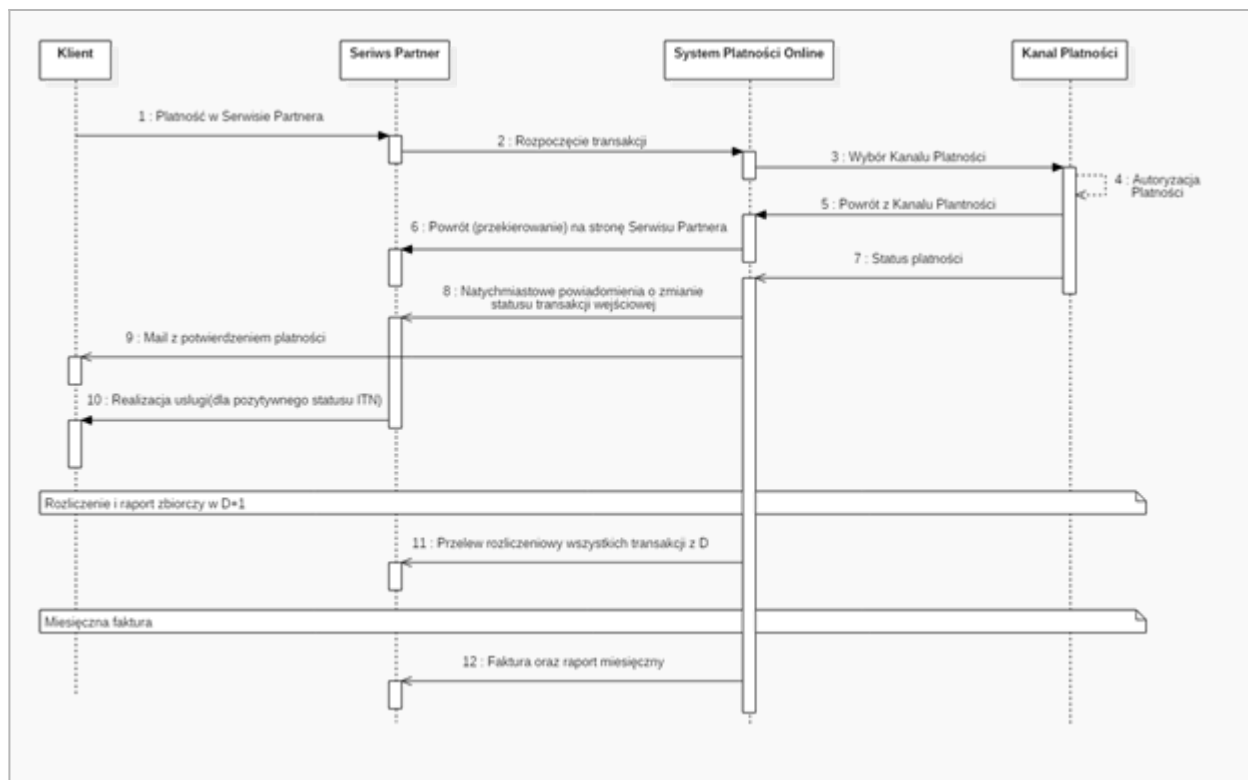
Poniżej ilustracja przykładowej struktury (bez wskazywania konkretnego modelu rozliczeniowego)



Modele rozliczeń

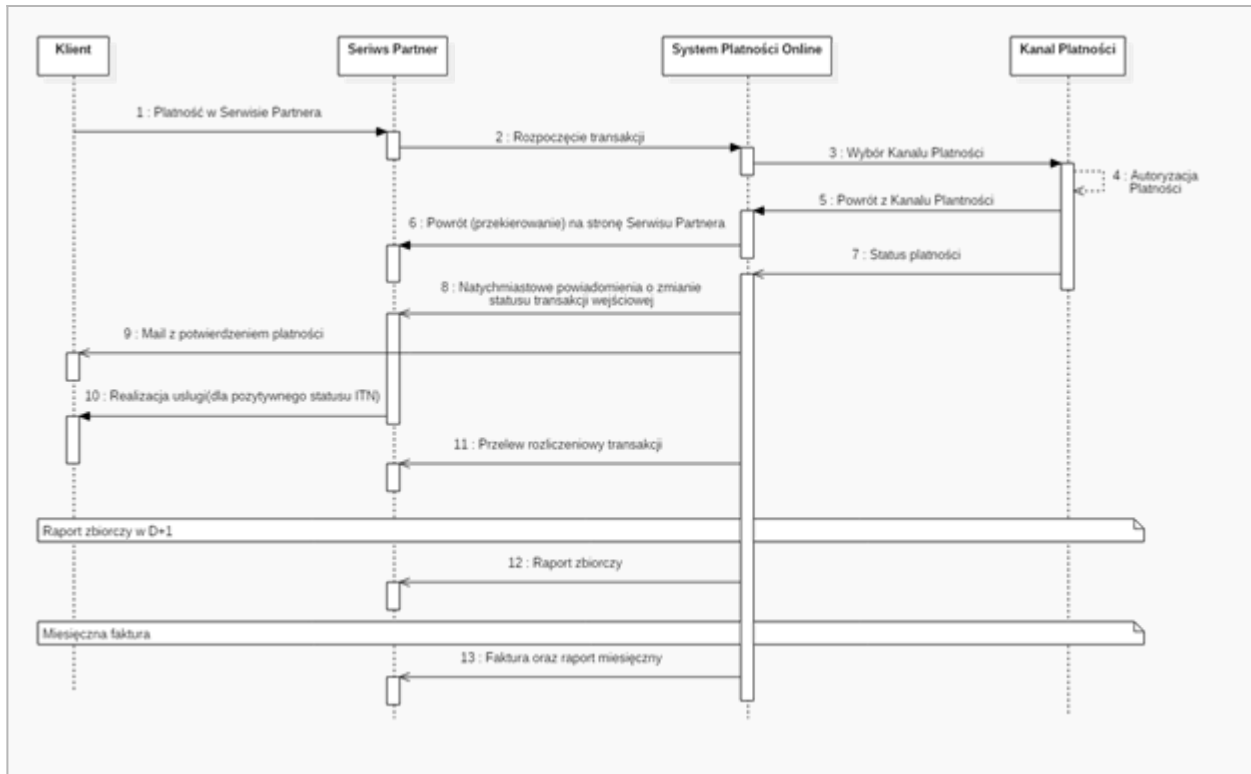
Model rozliczeń zbiorczych transakcji (model domyślny)

Rozliczenia zbiorcze następują następnego dnia roboczego (D+1).



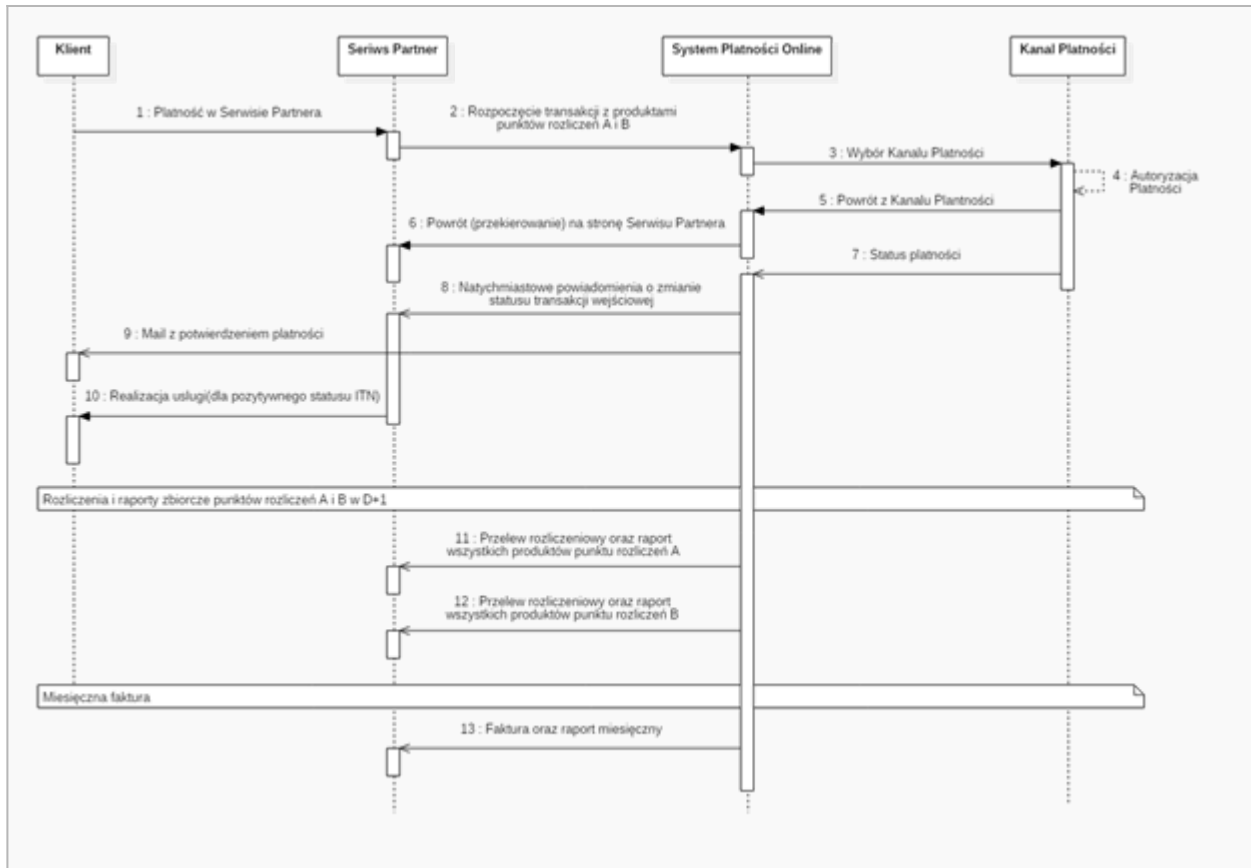
Model rozliczeń transakcji po każdej wpłacie

Rozliczenia po każdej wpłacie wykonywane mogą być niezwłocznie po otrzymaniu wpłaty od Klienta na wskazane w parametrach Linka płatności dane transakcji (opcje: Konto odbiorcy, Tytuł przelewu rozliczeniowego, Nazwa odbiorcy przelewu rozliczeniowego).



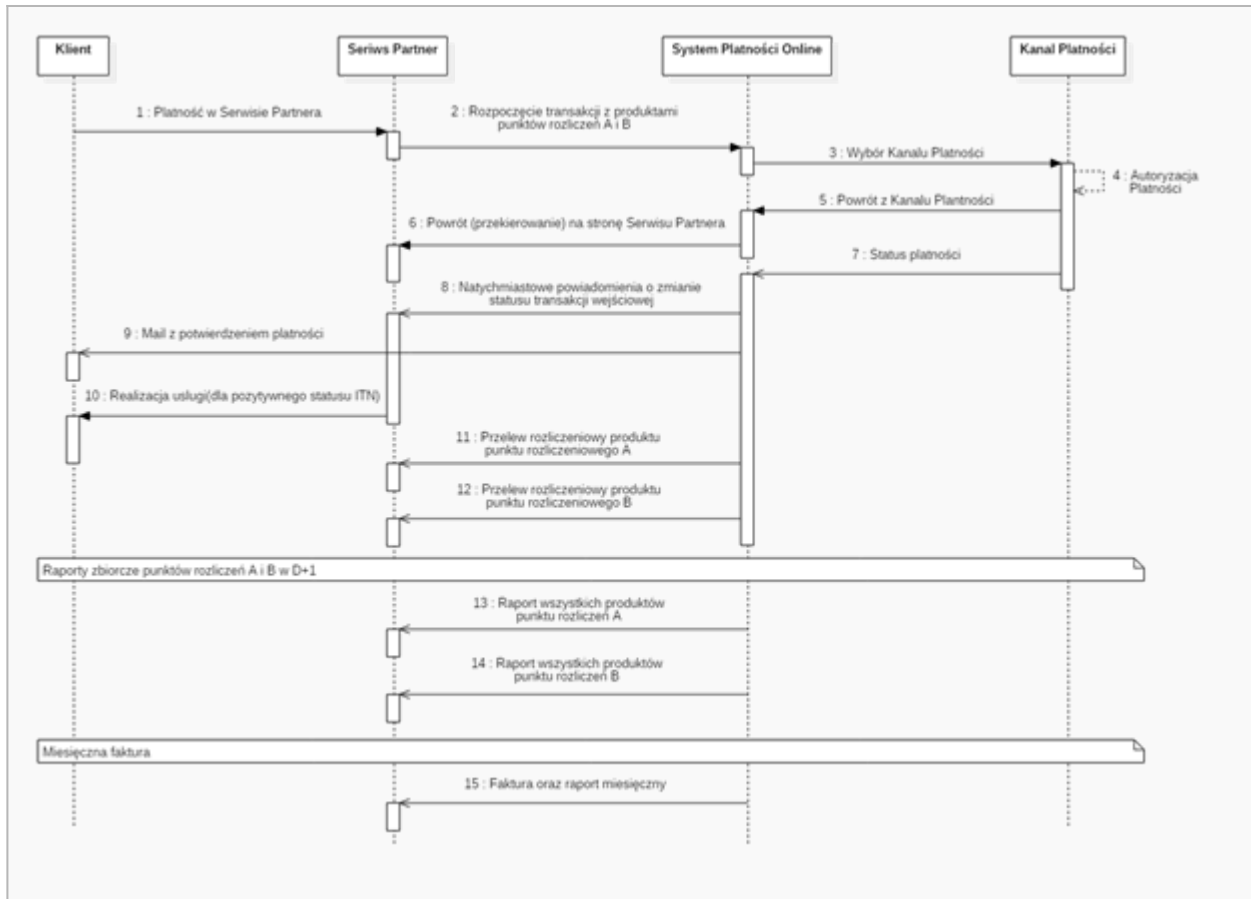
Model rozliczeń zbiorczych produktów

Rozliczenia zbiorcze następują następnego dnia roboczego (D+1).



Model rozliczeń produktów po każdej wpłacie

Rozliczenia po każdej wpłacie wykonywane mogą być niezwłocznie po otrzymaniu wpłaty od Klienta na wskazane w parametrach Linka płatności dane produktu (opcje: Konto odbiorcy, Tytuł przelewu rozliczeniowego, Nazwa odbiorcy przelewu rozliczeniowego).



Model rozliczeń transakcji na żądanie

Rozliczenia mogą być zlecane przez Partnera poprzez wywołanie metody: **transactionSettlement**.

