Dokumentacja usługi weryfikacji - ID HUB

Data wygenerowania: 2026-10-29

Specyfikacja integracji 4
Definicje 4
Narzędzie ID HUB 4
Sposób działania usługi ID HUB 5
Procedura integracji ID HUB 6
Integracja dla nowego Partnera 6
Integracja dla Partnera z wdrożonym „Przelewem weryfikacyjnym" 6
Środowiska 7
Środowisko testowe 7
Środowisko produkcyjne 7
Modele integracji 7
Paywall 7
Whitelabel 7
Tryby pracy ID HUB 8
Pobieranie danych klienta (DATA HARVEST) 8
Weryfikacje (PERSONAL VERIFICATION) 9
Cięcie danych 9
Porównywanie danych 10
Dane osobowe 11
Dane, które Komponenty są w stanie zweryfikować 11
Dane firmowe 12
Dane firmowe możliwe do zweryfikowania przez system 12
Komponenty weryfikujące 12
Komponent weryfikujący AIS 12
Przykładowe typy raportów Komponentu AIS 12
Dostęp do raportów 13
Komponent weryfikujący 1PLN (Przelew Weryfikacyjny) 14
Sposoby realizacji przelewu 14
Komponent weryfikujący PHOTO 15
Obsługiwane dokumenty 15
Raporty zwracane w obiekcie wynikowym dla pozytywnej weryfikacji 15
Komponent weryfikujący MOBYWATEL 17
Implementacja usługi w systemie Partnera 17
Metody interfejsu programistycznego 17
BankList 17
BankList – Parametry wejściowe w URL 17
BankList – Obiekt Bank 18
BankList – Przykład żądania i odpowiedzi 18
Initiate 19
Inicjowanie weryfikacji w komponencie mObywatel 19
Initiate – Parametry wejściowe 19
Inicjowanie weryfikacji w modelu whitelabel bez podania ID banku 20
Initiate with iban – Parametry wejściowe 20
Initiate – Słownik kluczy mapy „params" dla wartości pola „type": PERSONAL_VERIFICATION 21
Initiate – Słownik kluczy mapy „params" dla wartości pola „type": COMPANY_VERIFICATION 22
Initiate – Słownik kluczy mapy „params" dla wartości pola „type": DATA_HARVEST 22
Initiate – Parametry wyjściowe 22
Initiate – przykład żądania i odpowiedzi 23
Initiate - przykład żądania dla weryfikacji firmowej (1 PLN) 23

Inicjowanie dwuetapowej weryfikacji w komponentach 1 PLN i AIS w PKO BP 23
Krok 1. Inicjacja weryfikacji 1 PLN 24
Krok 2. Inicjacja weryfikacji AIS 24
Initiate – przykład żądania i odpowiedzi w przypadku dwuetapowej weryfikacji w PKO BP 25
Start 26
Metoda Start dla komponentu mObywatel 26
Metoda Start i Refresh dla komponentu mObywatel - przykład żądania (dla metody Refresh) i odpowiedzi (dla metod Start i Refresh) 26
Result 28
Result – Parametry wejściowe 28
Result – Parametry wejściowe 28
Result – Parametry wyjściowe 28
Result - Przykład żądania i odpowiedzi 30
Health-Check 35
Health-Check komponentów PHOTO i MOBYWATEL 35
Przykładowa odpowiedź dla Partnera z aktywnymi komponentami: 35
Przykładowa odpowiedź dla Partnera, który nie posiada skonfigurowanych komponentów 35
Przykładowa odpowiedź dla Partnera, który nie istnieje 35
BankList-notification (PUSH) 36
Result-notification (PUSH) 36
Powrót klienta z komponentu do systemu Partnera 37
Ręczna zmiana wyniku weryfikacji 37
Kształt odpowiedzi metody Result z włączoną możliwością ręcznej zmiany wyniku weryfikacji 38
Kształt odpowiedzi metody Result po przeprowadzeniu ręcznej zmiany wyniku weryfikacji 38
Potwierdzenia wykonanych weryfikacji 38
Rekomendacje deweloperskie 39
Bezpieczeństwo 39
Sieć 39
Dane osobowe i raporty 39
Uwierzytelnianie 40
Przykład sposobu liczenia sumy kontrolnej (JAVA) 40
Przykład sposobu liczenia sumy kontrolnej (PHP) 41
Przykład sposobu liczenia sumy kontrolnej (C#) 41
BasicAuth 42
NONE 42

Specyfikacja integracji

Wersja dokumentu - 2.0.4

Definicje

ID HUB (Usługa, System) – opisywane narzędzie. Zorganizowany zbiór mechanizmów służących do oceny przez Partnera prawdziwości danych Klienta. Weryfikacja może obejmować dane osobowe, zgodność wizerunku z dokumentem czy zdolność kredytową.

Klient – użytkownik docelowy, osoba korzystająca z usługi, dokonująca uwierzytelnienia swoich danych.

Partner – kontrahent, podmiot integrujący się z ID HUB (administrator strony/platformy technologicznej, z której korzysta Klient).

Paywall – widok w interfejsie, na którym prezentowana jest Klientowi lista dostępnych banków/kanałów płatności, którymi może zrealizować usługę.

Komponent – element składowy usługi dostarczający mechanizm weryfikacji w określonym dla Partnera zakresie (np. weryfikacja imienia i nazwiska na podstawie danych z rachunku bankowego).

Raport dzienny / Raport miesięczny – wysyłany do Partnera dokument z danymi dotyczącymi ilości transakcji wykonanych w określonym czasie.

Weryfikacja – proces technologiczny polegający na przyjęciu przez System danych Klienta oraz przekazaniu informacji zwrotnej określonej przez dany Komponent (np. foto-weryfikacja zdjęć, dane rachunku bankowego).

Wynik weryfikacji – dokument wytworzony przez Komponent weryfikacyjny. Może zawierać status lub dane dodatkowe. Jego treść jest specyficzna dla danego Komponentu.

Narzędzie ID HUB

ID HUB to narzędzie umożliwiające zbadanie przez Partnera wiarygodności swojego Klienta lub pobrania danych osobowych Klienta i danych o kliencie z zewnętrznych źródeł (dokumenty, rachunki bankowe, bazy gospodarcze, aplikacja mObywatel).

Weryfikacja i pozyskanie danych mogą być wykonane przez różne komponenty, każdy z nich o odmiennej charakterystyce.

Możliwości narzędzia to:

1. Usługa dostępu do informacji o rachunku (AIS) - mechanizm skanowania rachunków bankowych przy pomocy interfejsów PSD2
2. Przelew weryfikacyjny - realizacja przelewu na jedną złotówkę i weryfikacja danych transakcji
3. Fotoweryfikacja - OCR dowodu osobistego oraz weryfikacja biometryczna klienta
4. mObywatel - pobieranie danych z aplikacji mObywatel.

ID HUB unifikuje wszystkie komponenty w jeden interfejs i standaryzuje sposób przeprowadzenia Klienta przez proces – niezależnie od tego, do jakiego rodzaju procedury weryfikacji/dostarczenia danych Partner zobowiązuje swoich Klientów.

Sposób działania usługi ID HUB

Klient w systemie Partnera wykonuje akcję wymagającą zweryfikowania podanych przez niego danych. Partner wysyła do Systemu żądanie utworzenia nowej Weryfikacji. ID HUB, na bazie konfiguracji konta Partnera, analizuje jakiego rodzaju Weryfikacja powinna być wykonana. Jeśli System dysponuje Komponentem, który ma możliwość zweryfikowania określonych przez Partnera parametrów, kontaktuje się z nim i buduje unikalny adres URL, na który należy przekierować Klienta.

Adres URL jest zwracany do systemu Partnera razem z identyfikatorem nadanym nowej Weryfikacji. System Partnera w odpowiednim momencie przekierowuje Klienta na otrzymany adres. Klient przechodzi do witryny w domenie Autopay lub bezpośrednio do banku i postępuje według kroków wyznaczanych przez wybrany Komponent weryfikacyjny. Po wykonaniu wszystkich akcji, jakich zażądał Komponent, Klient jest przekierowywany z powrotem do systemu Partnera. W tym momencie Wynik weryfikacji powinien być już dostępny dla systemu Partnera.

Wynik zakończonej weryfikacji Partner pobiera, wysyłając żądanie na ustalony, opisany w dalszej części dokumentu adres. W odpowiedzi System zwróci obliczony wynik weryfikacji albo informację, że ten jest w trakcie przygotowywania. Do samego wyniku, poszczególne komponenty mogą dołączyć dodatkowe dane – tzw. raporty. Treści raportów są specyficzne dla wybranych komponentów.

Jest możliwość, aby to System aktywnie powiadamiał Partnera o przygotowanym wyniku weryfikacji. Jest to tzw. PUSH. Wymaga on dostarczenia przez system Partnera adresu URL, który HUB będzie wywoływał, dostarczając notyfikację o możliwości pobrania gotowego wyniku.

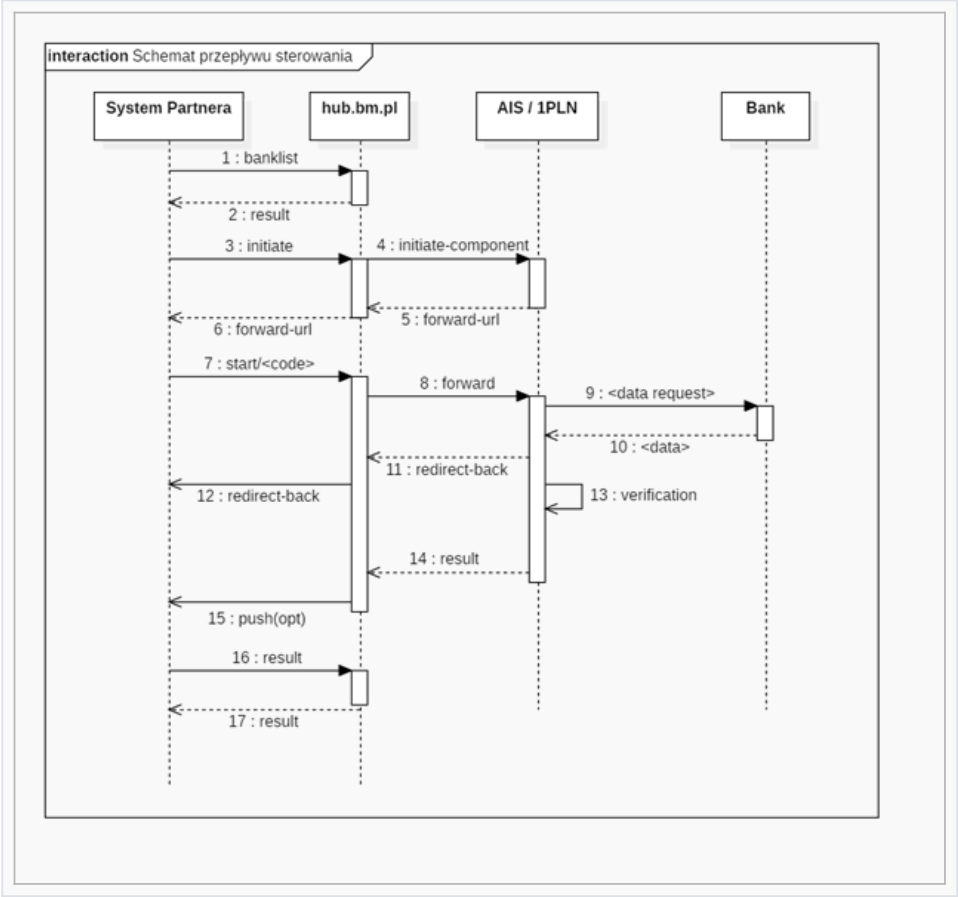


Diagram sekwencji

Procedura integracji ID HUB

Integracja dla nowego Partnera

Integracja z Systemem weryfikacji składa się z następujących kroków:

1. Ustalenie z Opiekunem Biznesowym parametrów działania systemu. W kroku tym Partner określa swoje potrzeby, definiuje wymagania względem ID HUB-a. Wybierane są niezbędne komponenty oraz parametry ich używania. Rezultatem tego kroku jest wypełniony dokument karty wdrożeniowej.

2. Na podstawie wypełnionej karty, w środowisku testowym tworzone jest konto.

3. Partner może skorzystać z aplikacji: https://id-hub-accept.bm.pl/test/start .

WSKAZÓWKA: Na stronie znajduje się wiele pól, z których nie wszystkie są wymagane do przeprowadzenia testu. Wartości wymagane w formularzu zależą od potrzeb i zamówienia złożonego w karcie wdrożeniowej oraz ostatecznie ustalonej konfiguracji.

4. Przygotowanie przez Partnera obsługi endpointów (w technologii REST+JSON) wystawionych w API Systemu:

- banklist – pobranie listy dostępnych banków z informacją o obsługujących je komponentach (krok opcjonalny)
- initiate – uruchomienie procesu weryfikacji
- start – przekierowanie klienta do komponentu
- result – pobranie wyniku weryfikacji

5. Przygotowanie przez Partnera adresów powrotu, tzw. "landing page", na którą należy przekierować klienta po wyjściu z systemu, gdzie powinien on oczekiwać na wynik weryfikacji.

6. Przygotowanie adresu do obsługi powiadomień o gotowym do pobrania wyniku weryfikacji. [OPCJONALNIE]

7. Przygotowanie adresu do obsługi powiadomień o zmianie w liście aktywnych banków. [OPCJONALNIE]

Integracja dla Partnera z wdrożonym „Przelewem weryfikacyjnym"

Partner może być już zintegrowany z usługą wystawianą przez Autopay - Przelew weryfikacyjny.

Połączenie się z ID HUB-em nie wyklucza współistnienia Przelewu weryfikacyjnego jako równoległego mechanizmu weryfikacji.

Jeśli jednak Partner życzyłby sobie, aby Przelew weryfikacyjny został, zgodnie z koncepcją ID HUBa przykryty jednym interfejsem i był używany jako komponent jednej usługi, nie da się niestety dokonać migracji bezobsługowej i prace techniczne w aplikacji Partnera będą konieczne.

Konfiguracja biznesowa (sposoby realizacji zwrotów, rozliczenia prowizji itp.) jak i część konfiguracji technologicznej (rodzaje zwracanych przez przelew weryfikacyjny podsumowań, raportów, uruchamiane podsystemy) pozostaną bez zmian, przeniesione zostaną do integracji w nowym modelu.

Odmienny będzie start weryfikacji, przekierowanie klienta i sposób odebrania ostatecznego wyniku.

Środowiska

Do użytku Partnera przygotowane są dwa środowiska: testowe i produkcyjne.

> **UWAGA:** Łącząc się do Aplikacji pod podanymi wyżej adresami rekomendujemy używanie protokołu TLS w wersji przynajmniej 1.2. Obsługa wersji niższych zostanie w niedługim czasie wycofana.

Środowisko testowe

- Aplikacja: https://id-hub-accept.bm.pl/api
- API Swagger: https://id-hub-accept.bm.pl/swagger/
- Aplikacja testująca: https://id-hub-accept.bm.pl/test/start

Środowisko produkcyjne

- Aplikacja: https://id-hub.bm.pl/api
- Aplikacja testująca: https://id-hub.bm.pl/test/start

Modele integracji

Partner może zintegrować się z ID HUB korzystając z modelu Paywall lub Whitelabel.

Paywall

1. Lista banków dostępna jest po stronie Autopay
2. Partner inicjuje weryfikację
3. Przekierowanie Klienta do frontendu Autopay
4. Klient realizuje proces
5. Klient powraca do systemu Partnera

W tym modelu wygląd może być spersonalizowany, zgodnie z wytyczymi Partnera. Oznacza to możliwość dodania logotypu, zmiany kolorów, czcionek, loaderów, itp.

Whitelabel

WSKAZÓWKA: Cechą modelu Whitelabel jest większa kontrola Partnera nad działaniami Klienta oraz minimalizacja czasu przebywania Klienta poza domeną Partnera (który ogranicza się tylko do przekierowań przez strony Autopay i pobytu w banku).

1. Paywall dostępny jest po stronie Partnera (na podstawie listy banków, pobieranej metodą BankList)
2. Partner inicjuje weryfikację
3. Przekierowanie Klienta wprost do banku
4. Klient realizuje proces
5. Klient powraca do systemu Partnera

W tym sposobie integracji pojawia się nowa metoda API – BankList. Partner wywołuje tę metodę i w odpowiedzi otrzymuje listę skonfigurowanych dla niego dostępów bankowych.

Dostępne są dwa rodzaje dostępów: 1PLN i AIS. W ramach 1PLN mamy dodatkowo podział na dostępy: PayByLink, PIS (zgodnie z PSD2) oraz Samodzielny przelew. Na etapie integracji Partner ma możliwość zdania się na zespół Autopay w doborze banków i przypisanych do nich komponentów, ma też możliwość ustalić listę banków skrojoną pod precyzyjne zapotrzebowanie. Pobrana lista banków powinna zostać wyświetlona klientowi celem wybrania dostępnego mu kanału weryfikacji.

-> wskazówka -> Lista dostępnych banków możliwych do skonfigurowania w ID HUB znajduje się na podstronie.

Na Partnera spada obowiązek odebrania od klienta stosownych zgód. Rodzaje zgód i ich treści ustalane są z zespołem Autopay przed rozpoczęciem prac integracyjnych.

Gdy klient wybierze bank, w którym ma rachunek, przechodzimy do inicjacji weryfikacji za pomocą metody opisanej w głównej części dokumentacji, uzupełnionej o pole z identyfikatorem banku. Partner otrzymuje adres przekierowania klienta.

Klient po dostaniu się na widok swojego banku autoryzuje systemom Autopay dostęp do swoich rachunków lub realizuje przelew. Uwaga, w przypadku komponentu AIS i skanowaniu historii dłuższej niż 180 dni, klient może być przekierowany do domeny Autopay celem wykonania dodatkowej autoryzacji SCA.

Klient jest przekierowywany z powrotem, wprost do serwisu Partnera z chwilowym „przeskokiem" przez domenę Autopay.

Pobranie raportu realizowane jest standardowym trybem, przy użyciu metody API - Result.

Tryby pracy ID HUB

Pobieranie danych klienta (DATA HARVEST)

Jest to tryb, który nie przeprowadza żadnych weryfikacji. Służy pobraniu danych Klienta z zewnętrznych źródeł. Partner ma wówczas możliwość dokonania weryfikacji danych Klienta własną metodą.

W trybie pobierania danych system przekierowuje Klienta do wybranego komponentu. Tam Klient autoryzuje dostęp Usługi do jego danych albo wprowadza je samodzielnie (Fotoweryfikacja).

ID HUB pobiera dane dostępne w danym komponencie i przekazuje je do Partnera w formie „surowej" lub wzbogacone o dodatkowe raporty (statystyczne, kategoryzujące rodzaje przelewów itp.).

Tryb pobierania danych dostępny jest w każdym oferowanym komponencie.

Weryfikacje (PERSONAL VERIFICATION)

Wszystkie weryfikacje dokonywane przez system polegają na porównaniu danych deklarowanych przez Klienta, z danymi jakie otrzymamy z wybranego komponentu.

Pozyskiwane z komponentów dane klientów są skomplikowane (pod względem braku ustandaryzowanej struktury, nieprzewidywalnej kolejności elementów, różnorodności adresów, imion i nazwisk), nie gwarantujemy 100% poprawnej klasyfikacji danych, a w konsekwencji 100% poprawności wyniku porównania danych.

Żeby współczynnik skuteczności cięcia i porównywania danych mieć jak na najwyższym poziomie posługujemy się szeregiem instrumentów i algorytmów, które mają w tym pomóc.

Cięcie danych

Dane adresowe, jakie komponenty Systemu zbierają ze swoich źródeł przybierają różne formaty.

Przykłady zbieranych danych:

```
IZABELA ZIELIŃSKA Warszawska 39/14, 58-400 Kamienna Góra
KOWALSKI MARCIN ul. OSIEK 990, 63-920 OSIEK
WRÓBLEWSKI MARCIN JERZY CEYNOWY 136/15 77-100 BYTÓW
ORGANEK MARTA I ORGANEK WANDA NADWIŚLAŃSKA 82/4 03-349 WARSZAWA
GOSPODARSTWO ROLNE KAMIL MARECZEK BODZIEJOWICE 7B 42-446 IRZĄDZE
JĘDRZEJ NOREK JADWIGA JASKÓŁA-NOREK BRZEŹNICKA 1C32-700 BOCHNIA PL
NIKODEM ARLETA JANA III SOBIESKIEGO 2/6 21-500 BIAŁA PODLASKA
JANUSZ-STOLARCZYK JANINA KOSZARSKO 1 22-335 ŻÓŁKIEW KA
```

Narzędzie do cięcia danych musi w liniach takich jak powyższe (i innych) rozpoznać:

1. Imię
2. Nazwisko
3. Ulicę
4. Numer domu
5. Numer klatki schodowej
6. Numer lokalu
7. Kod pocztowy
8. Miejscowość

Nieprzewidywalny i niedeterministyczny charakter danych agregowanych z zewnętrznych źródeł wymusza na Systemie posiłkowanie się specjalnymi technikami, aby podział linii na konkretne porcje danych był jak najbliższy ideału.

Do technik tych zaliczamy:

1. Stosowanie słowników imion i nazwisk z rejestru Pesel

2. Weryfikowanie adresów przy użyciu baz adresowych i kodów pocztowych

Przyjęty sposób działania może generować nieoczekiwane rezultaty (i w efekcie nieudane procesy weryfikacyjne) dla użytkowników legitymujących się danymi adresowymi spoza obszaru Polski oraz imieniem i nazwiskiem spoza bazy Pesel (analogicznie, gdy w danej wejściowej jest nazwa firmy).

Mimo przyłożenia najwyższych starań, aby otrzymane dane były poprawnie pocięte i labelizowane, należy być przygotowanym na ewentualność, że pewne weryfikacje są skazane na nieudany rezultat. Wynika to z braku wpływu na jakość danych, jakie ID-HUB otrzymuje z zewnętrznych źródeł. Dane te są czasem obciążone wadami uniemożliwiającymi poprawne przetworzenie (np. imię, nazwisko i adres z tytułu przelewu mogą być pozbawione znaków spacji tworząc tzw. zlepek słów).

Porównywanie danych

System umożliwia zdefiniowanie zakresu pól z danym, które mają podlegać porównaniu, oraz zasad, którym ma podlegać mechanizm porównujący. Lista pól wynika z zakresu danych możliwych do uzyskania z danych przelewu weryfikacyjnego:

- imię
- nazwisko
- ulica
- numer domu
- klatka
- numer mieszkania
- kod pocztowy
- miasto
- tytuł
- NRB

Parametry działania trybu porównywania danych:

1. Tolerancja współwłasności

Imię i nazwisko są porównywane na szczególnych zasadach, umożliwiających obsługę danych z rachunków współdzielonych. Rachunek współdzielony to taki, z którego otrzymujemy dane dwóch osób. Jeśli dane współwłaściciela lub pełnomocnika nie znajdą się w danych nadawcy przelewu, rachunek taki jest uznawany za indywidualny, należący do jednej osoby. Możliwe ustawienia: a) Dane Klienta mogą pochodzić z rachunku wspólnego b) Dane Klienta mogą pochodzić z rachunku wspólnego, ale muszą się znajdować na pierwszym miejscu w danych z przelewu c) Dane Klienta nie mogą pochodzić z rachunku wspólnego

2. Tolerancja nadmiarowych danych w polach: imię, nazwisko, ulica. Możliwe ustawienia: a) Obustronna tolerancja nadmiarowych danych

-> przykłady:

imię z formularza	imię z banku	wynik porównania
Krystyna	Krystyna Maria	pozytywny
Krystyna Maria	Krystyna	pozytywny

b) Tolerancja danych nadmiarowych w formularzu

-> przykłady:

imię z formularza	imię z banku	wynik porównania
Krystyna	Krystyna Maria	negatywny
Krystyna Maria	Krystyna	pozytywny

c) Tolerancja danych nadmiarowych na rachunku bankowym

-> przykłady:

imię z formularza	imię z banku	wynik porównania
Krystyna	Krystyna Maria	pozytywny
Krystyna Maria	Krystyna	negatywny

3. Wrażliwość na znaki diakrytyczne

Domyślnie znaki diakrytyczne są istotne w trakcie porównywania danych, jednak można skonfigurować tryb tak, by je ignorował.

Dane osobowe

Tryb weryfikacji danych osobowych polega na weryfikacji danych zadeklarowanych przez Klienta, danymi, które System pobiera z m.in. systemów bankowych.

Dane osobowe System weryfikuje za pomocą Komponentów: AIS, 1PLN, PHOTO, MOBYWATEL.

Dane, które Komponenty są w stanie zweryfikować

Dane	AIS	1PLN	PHOTO	MOBYWATEL
Imię	✔	✔	✔	✔
Nazwisko	✔	✔	✔	✔
Nazwa firmy	✔	✔	✖	✖
Ulica	✔	✔	✔	✔
Numer domu	✔	✔	✔	✔
Numer klatki schodowej	✔	✔	✔	✔
Numer mieszkania	✔	✔	✔	✔
Kod pocztowy	✔	✔	✔	✔
Miasto	✔	✔	✔	✔
Numer rachunku	✔	✔	✖	✖
Numer dowodu osobistego	✖	✖	✔	✔

Dane	AIS	1PLN	PHOTO	MOBYWATEL
Miejscowość urodzenia	✖	✖	✔	✔
Data ważności dokumentu	✖	✖	✔	✔
Numer PESEL	✖	✖	✔	✔

Dane firmowe

Tryb weryfikacji danych firmowych polega na podaniu w fazie inicjacji numeru NIP albo Regon. W tak zainicjowanym procesie użytkownik jest przekierowywany do strony logowania swojego banku. Po zatwierdzeniu przelewu, klient wraca do systemu Partnera. W tym czasie System pobiera dane z rachunku klienta oraz z bazy Głównego Urzędu Statystycznego. Dane z rachunku poddawane są porównaniu z danymi z GUS. Zgodność pozyskanej nazwy nadawcy przelewu z nazwą w GUS skutkuje pozytywnym rezultatem weryfikacji. W przeciwnym razie nadawany jest status negatywny.

Tryb weryfikacji danych firmowych dotyczy tylko rachunków bankowych dostępnych w bankowościach internetowych w kontekście "Retail".

Dane firmowe możliwe do zweryfikowania przez system

| Rodzaj danej | 1PLN | | --- | --- | --- | | Numer rachunku | ✔ | | Nazwa firmy | ✔ | | Numer regon | ✔ | | NIP | ✔ |

Komponenty weryfikujące

Komponent weryfikujący AIS

Komponent nazywany umownie "AIS" to podsystem ID HUB, który działa w oparciu o analizę historii rachunku bankowego Klienta.

Klient po wybraniu swojego banku, na liście banków w serwisie Partnera albo Autopay (zależnie trybu integracji), jest przekierowywany na stronę logowania do swojego konta. Tam Klient dokonuje uwierzytelnienia i autoryzacji Systemu do pobrania historii transakcji. Klient powraca do serwisu, w którym zaczął proces i tam oczekuje, aż Komponent zakończy pobieranie i analizę danych.

Komponent AIS, oprócz weryfikacji danych klienta, ma możliwość generowania raportów i podsumowań.

Przykładowe typy raportów Komponentu AIS

Raport	Opis
Dane osobowe	Raport zawierający następujące dane osobowe: 1) Imię i nazwisko 2) Adres meldunkowy

Raport	Opis
Dane o rachunkach	Raport zawierający listę rachunków, jakie Klient udostępnił do skanowania. Zawiera: 1) Dane właściciela 2) Numer i rodzaj konta 3) Saldo
Zagregowane dane finansowe	Raport zawierający podsumowanie aktywności na rachunku Klienta w zadanym okresie.Transakcje grupowane są według następujących reguł: 1) Numer konta 2) Przychód/rozchód 3) Kategoria transakcji (podatki, zakupy, kredyty, itd.) Do komórek wyznaczonych przez powyższe reguły sumujemy kwoty przelewów. Raport wzbogacamy o informację o dacie pierwszego przelewu.
Dane surowe	Raport zawierający listę transakcji z rachunku Klienta w ustalonym formacie (np. CSV, JSON).
Dane o rachunkach	Lista rachunków bankowych ze szczegółowym opisem.
Dane o rozszerzonym zakresie (tzw. "lambda")	Raport zawierający zestaw obliczonych różnorakich (definiowanych przez wymagania Partnera) parametrów. Mogą to być m.in. mediany/średnie/liczby/min/max kwot spełniających określone warunki (np. rodzaj kategorii), liczba dni jakie upłynęły od określonego zdarzenia na rachunku, itp.

Pliki do pobrania, w formacie JSON, ze schematem danych i przykładowymi wartościami znajdują się pod adresem: https://developers.autopay.pl/weryfikacje/ais/raporty.

Możliwe jest także stworzenie raportów pod indywidualne potrzeby Partnera. W celu uruchomienia nowego rodzaju raportu należy skontaktować się z opiekunem biznesowym.

Dostęp do raportów

Raporty generowane przez komponent AIS nie są integralną częścią obiektu z wynikiem weryfikacji. Są one dostępne pod adresem URL, który zostanie do wyniku weryfikacji załączony. Przyczyną odseparowania wyniku weryfikacji od powiązanych z nim raportów jest rozmiar danych, jakie raport może wygenerować.

Dostęp do treści raportów wymaga autoryzacji metodą BasicAuth. Login i hasło zostaną dostarczone Partnerowi w formularzu wdrożeniowym lub na etapie integracji.

W **Przykład pełnego raportu generowanego przez komponent AIS** znajduje się przykładowy obiekt z wybranymi raportami.

W **Przykład drzewa kategorii do budowania zagregowanego raportu finansowego** znajduje się przykładowa lista-drzewo kategorii, które jest używane przy obliczaniu raportu finansowego. Istnieje możliwość indywidualnego zaprojektowania drzewa kategorii wedle potrzeb Partnera.

Komponent weryfikujący 1PLN (Przelew Weryfikacyjny)

Komponent 1PLN (Przelew Weryfikacyjny) to mechanizm weryfikacji Klienta, w którym użytkownik dokonuje przelewu ustalonej kwoty na rachunek bankowy Autopay.

Po otrzymaniu przelewu, Komponent sprawdza, czy przekazane z banku informacje o nadawcy są zgodne z danymi otrzymanymi w momencie inicjacji weryfikacji. Obliczony wynik przekazywany jest Partnerowi, który podejmuje dalsze kroki w procesie biznesowym Klienta.

Personalizacja konfiguracji dostępna jest dla następujących rozszerzeń:

Rozszerzenie	Opis
Predefiniowany tytuł przelewu	Każdy przelew wykonywany przez Klienta będzie w swoim tytule zawierał ustalony opis, na przykład: „Potwierdzenie zawarcia umowy z XYZ"
Inna kwota przelewu	Domyślnie użytkownik dokonuje przelewu kwoty 1 zł. Istnieje jednak możliwość konfiguracji komponentu tak, aby użytkownik dokonywał przelewu innej, ustalonej kwoty.
Zwracanie danych otrzymanych w tytule przelewu	Do obiektu z wynikiem weryfikacji dołączona zostanie struktura z danymi zawierającymi szczegóły przelewu zrealizowanego przez Klienta: firstNameFromTransfer – imię lastNameFromTransfer – nazwisko streetFromTransfer – ulica streetHouseNumberFromTransfer – numer domu streetFlatNumberFromTransfer – numer mieszkania streetStaircaseNumberFromTransfer – numer klatki cityFromTransfer – ulica postCodeFromTransfer – kod pocztowy bankAccountNumberFromTransfer – numer rachunku unseparatedDataFromTransfer – dane przed klasyfikacją companyNameFromTransfer – nazwa firmy pobrana z bazy GUS podczas weryfikacji danych z przelewu (dotyczy trybu weryfikacji firmowych)

Sposoby realizacji przelewu

Przelew weryfikacyjny wykonywany w Bramce Płatniczej Autopay może być zrealizowany za pośrednictwem czterech sposobów.

Dostępne kanały płatności:

Pay By Link - przelew realizowany jest za pośrednictwem formularza wygenerowanego w bankowości elektronicznej wybranego banku. Cały proces zamyka się w kilkadziesiąt sekund, do kilku minut.

Samodzielny przelew - przelew realizowany jest podobnie w metodzie Pay By Link, z tą różnicą, że dane do przelewu użytkownik musi wprowadzić w swojej bankowości ręcznie, na podstawie przedstawionej w Bramce Płatniczej formatki.

PSD2-PIS - mechanizm podobny do metody Pay By Link, realizowany przy użyciu bankowych interfejsów PSD2.

Mam konto w innym banku - w sytuacji, gdy powyższe kanały płatności są trwale lub przejściowo niedostępne, przelew może być zrealizowany klasyczną metodą przelewu Elixir. Czas realizacji weryfikacji w tym przypadku może zająć do dwóch dni roboczych.

Weryfikacje realizowane przez komponent 1PLN są weryfikacjami, które mogą trwać dłużej. Czynnikami wpływającymi na ten czas są mechanizmy bankowe dokonujące transferu pieniędzy wraz z procedurami zabezpieczającymi ten proces.

W szczególnych przypadkach czas realizacji przelewu może zająć do siedmiu dni (dotyczy tygodni z następującymi po sobie dniami świątecznymi). Z tej przyczyny czas ważności weryfikacji i oczekiwania na przelew ustawiamy standardowo na 7 dni. Ten parametr może być przyczyną nienadchodzących do systemu Partnera powiadomień o zakończonej weryfikacji nawet przez kilka dni.

Komponent weryfikujący PHOTO

Komponent PHOTO jest mechanizmem, w którym System pozyskuje dane osobowe klienta wprost z dokumentu tożsamości (dowodu osobistego lub paszportu) Klienta. Prócz pozyskania danych z dokumentu, komponent umożliwia także ocenę zgodności parametrów biometrycznych klienta przy urządzeniu, z którego korzysta, z wizerunkiem znajdującym na przedstawionym dokumencie tożsamości.

Klient może zweryfikować swoją tożsamość przy pomocy jednego urządzenia mobilnego albo przy pomocy dwóch urządzeń – komputera typu desktop, w którym realizuje część procesu odpowiedzialną za zgłoszenie danych do weryfikacji i telefonu, w którym przebiega proces pozyskania i przekazania zdjęć, po czym wraca do procesu realizowanego w komputerze.

Obsługiwane dokumenty

Dowód osobisty: polski Paszport: polski, ukraiński, białoruski, gruziński

Raporty zwracane w obiekcie wynikowym dla pozytywnej weryfikacji

Raport	Opis
Dane z OCR (Zakres danych jest ustalany w fazie konfiguracji)	Raport zawiera dane tekstowe pozyskane z dokumentu tożsamości, wybrane spośród poniższego zakresu: idDocumentTypeFromOcr: "IDENTITY_CARD" idDocumentIssueCountryFromOcr: "POL" idDocumentIssueStateFromOcr: "" idDocumentExpiryDateFromOcr: "2021-06-30" idDocumentIssuingDateFromOcr: "2011-06-30" idDocumentNumberFromOcr: "ATX012345" peselFromOcr: "74121524371" genderFromOcr: "M" streetFromOcr: "Powstańców Warszawy" streetHouseNumberFromOcr: "6" streetStircaseNumberFromOcr: "" streetFlaNumberFromOcr: "" postCodeFromOcr: "81-718"," cityFromOcr: "Sopot" fullAddressFromOcr: "Powstańców Warszawy 6, 81-718 Sopot" dateOfBirthFromOcr: "1974-12-15" placeOfBirthFromOcr: "ORNETA" firstNameFromOcr: "SZYMON" secondNameFromOcr: "JAN" thirdNameFromOcr: "PAWEŁ" maidenNameFromOcr: "ROGALIK" lastNameFromOcr: "ROGALIK" fullNameFromOcr: "SZYMON JAN PAWEŁ ROGALIK"
Status procesu fotoweryfikacji	Są to trzy wskaźniki: documentStatus – określa status przetwarzania dowodu tożsamości bioStatus – określa status przetwarzania parametrów biometrycznych overallStatus – określa status całego procesu i zgodności dokumentu z parametrami biometrycznymi Możliwe wartości jakie przyjmują statusy to: VERIFIED – status pozytywny NOT_PERFORMED – weryfikacja w ogóle się nie odbyła SUSPICIOUS – weryfikacja została przeprowadzona, ale pojawiły się niezgodności w trakcie sprawdzania autentyczności dokumentu W sytuacji wystąpienia problemów z weryfikacją Klienta, do odpowiedzi dołączane jest pole "verificationProblems", którego wartością są kodowy oznaczające zlokalizowane problemy, np. "expiry_date" oznaczające problem z datą ważności dokumentu
Wynik porównania danych (result)	System porównuje wybrane przez Partnera dane z dokumentu z danymi pozykanymi w inicjacji weryfikacji w ID HUB. Wynik może przyjąć następujące wartości: POSITIVE - dane są zgodne dane są różne Poziom zgodności danych jest określany na podstawie konfiguracji trybu porównania, opisanego w części https://developers.autopay.pl/weryfikacje/dokumentacja#por%C3%B3wnywanie-danych

ID HUB porównuje dane otrzymane z komponentu PHOTO z danymi zadeklarowanymi przez Klienta. Dane, które mają być poddane weryfikacji, powinny znaleźć się w parametrach wejściowych (szczegóły opisuje metoda Initiate).

Komponent weryfikujący MOBYWATEL

Komponent "MOBYWATEL" umożliwia pobranie danych z Systemu mObywatel, dostarczanego przez Ministerstwo Cyfryzacji.

Aby móc korzystać z komponentu mObywatel, niezbędne jest nawiązanie współpracy z Ministerstwem w celu uzyskania odpowiedniego certyfikatu. Złożenie wniosku jest możliwe na stronie https://wspolpraca.mobywatel.gov.pl/ Certyfikat jest przechowywany w systemie Autopay i jest wykorzystywany w imieniu i na rzecz jego właściciela. Proces weryfikacji od strony Klienta składa się z kilku prostych kroków. Klient może przekazać swoje dane, skanując kod QR w aplikacji mObywatel na swoim telefonie. Kod QR jest zwracany w odpowiedzi na inicjację weryfikacji mObywatel i wyświetlany na stronie Partnera. Po pobraniu danych, mogą one podlegać dodatkowemu porównaniu, a po przetworzeniu są wysyłane jako wynik, analogicznie jak dane z pozostałych komponentów.

Zestawy danych możliwych do pobrania z Systemu mObywatel są określone w dokumentacji udostępnianej przez Ministerstwo Cyfryzacji, oraz ograniczone nadanymi w certyfikacie uprawnieniami.

Implementacja usługi w systemie Partnera

Komponent funkcjonuje wyłącznie w modelu whitelabel. Aby Klient mógł skorzystać z usługi, należy zaimplementować funkcjonalność wyświetlania danych otrzymywanych z systemu ID HUB w wersji desktop oraz mobile. Każdorazowo, w odpowiedzi na inicjację weryfikacji, ID HUB zwróci kod QR oraz identyfikator składający się ze znaków możliwych do skopiowania przez Klienta. W zależności od urządzenia Klienta, Partner powinien wyświetlić odpowiednią daną. Kod ma określony termin ważności i domyślnie jest to 60 sekund. Po tym czasie kod wygasa i jest możliwość jego odświeżenia. W takim przypadku Klient powinien mieć możliwość wygenerowania nowego kodu QR, poprzez kliknięcie odpowiedniego przycisku, po którego użyciu Partner wywoła metodę odświeżenia kodu.

Metody interfejsu programistycznego

BankList

GET /api/bank/v1.1/list/{partnerUuid}

Służy do pobrania listy banków powiązanych z partnerem. Odpowiedź zawiera listę banków z informacją, jakie komponenty umożliwiają przeprowadzenie subskrybowanej usługi (weryfikacja/pobranie historii transakcji) oraz aktualny status dostępności banku wraz z datą ostatniej zmiany statusu.

BankList – Parametry wejściowe w URL

ID	nazwa	typ	wymagany	opis
n/d	partnerUuid	uuid	tak	identyfikator Partnera

Parametry wyjściowe

ID	nazwa	typ	wymagany	opis
5	banks	map<Integer, Bank>	tak	Lista banków powiązanych z profilem Partnera. Klucze mapy są identyfikatorami banku. Opis obiektu Bank w tabeli poniżej.
25	status	Przyjmuje wartości: OK, ERROR	nie	
30	description	string	nie	dodatkowy komentarz związany z akcją - komunikat informacyjny, jeśli status=OK, komunikat błędu, jeśli status=ERROR

BankList – Obiekt Bank

ID	nazwa	typ	wymagany	opis
5	name	string	tak	Nazwa banku
10	bic	string	nie	Kod BIC/SWIFT banku
15	iconUrl	string	nie	Adres URL do ikony logotypu banku
20	additionalConsentsRequired	boolean	nie	Flaga określająca konieczność wyświetlenia dodatkowych zgód dla klienta (dla modelu white label).
25	component	string	tak	Nazwa komponentu, jaki zostanie wybrany w procesie. W szczególnych przypadkach klient zostanie przekierowany do innego komponentu. Np. w przypadku nagłej niedostępności wstępnie wybranego komponentu w trakcie realizowanego procesu.

BankList – Przykład żądania i odpowiedzi

Request: GET https://id-hub-accept.bm.pl/api/bank/v1.1/list/c455(...)6d89

Response:


```
{
  "status": "OK",
  "description": null,
  "hash": null,
  "banks": {
    "1": {
      "name": "Mbank",
      "bic": "BREXPLPWMBK",
      "iconUrl": "https://platnosci.bm.pl/pomoc/grafika/1800.png",
      "additionalConsentsRequired": true,
      "components": "AIS"
      }
    },
    "24": {
```

```
        "name": "Test Mock Bank",
        "bic": "BMMOCKBANK",
        "iconUrl": null,
        "additionalConsentsRequired": true,
        "components": "1PLN"
      }
    }
  }
```

Initiate

POST /api/verification/v1.0/initiate

Służy do zainicjowania procesu weryfikacji w systemie. Na podstawie odebranych parametrów wejściowych i dokonanych ustaleń projektowych system wybiera odpowiedni Komponent weryfikacyjny i przygotowuje adres do przekierowania Klienta.

Inicjowanie weryfikacji w komponencie mObywatel

POST /verification/v1.0/initiate POST /verification/v1.1/initiate/mobywatel

Metoda initiate może być wywołana za pomocą dwóch endpointów. Endpoint w wersji 1.0 wymaga podania parametru component (MOBYWATEL). Endpoint /verification/v1.1/initiate/mobywatel domyślnie ustawia ten parametr. Jeśli przy użyciu endpointa v1.1 zostanie dodatkowo uzupełnione pole component, zostanie ono zignorowane.

Initiate – Parametry wejściowe

nazwa	typ i zakres danych	wymagany	opis
verificationId	string (^[a-zA-Z0-9-_]{1,64}+$)	nie	identyfikator weryfikacji nadany przez Partnera
email	string (zakres zgodny z EmailValidator z Apache Commons ver. 1.6)	tak – dla 1PLN, nie - dla PHOTO; tak – dla AIS, nie- dla MOBYWATEL	adres email klienta
type	enum: PERSONAL_VERIFICATION i COMPANY_VERIFICATION i DATA_HARVEST	tak	rodzaj weryfikacji do wykonania w Systemie
params	map<string, string>	tak	mapa parametrów wejściowych do weryfikacji przez Komponent danego typu. Lista parametrów jest określona przez typ weryfikacji (patrz pole "type")
partnerUuid	uuid	tak	tekstowy identyfikator Partnera

nazwa	typ i zakres danych	wymagany	opis
bankId	integer (dozwolone wartości to klucze mapy, zwracanej metodą BankList)	nie	id wybranego banku - wymagany w modelu „White Label” w komponentach AIS i 1PLN
component	enum: AIS, 1PLN, PHOTO, MOBYWATEL	nie	wybrany przez Partnera rodzaj komponentu do zainicjowania (gdy jest ich więcej niż jeden i Partner nie chce zdać sięna dobór automatyczny)
test	enum: true, false	nie	znacznik informujący o testowej operacji

Inicjowanie weryfikacji w modelu whitelabel bez podania ID banku

POST /api/verification/v1.0/initiate-with-iban

Istnieje moliwość zainicjowania weryfikacji w modelu whitelabel w komponencie 1 PLN bez podania ID banku, ale zamiast tego z podaniem IBANu Klienta. Na podstawie 4. lub 8. cyfr prefiksu, zostanie przypisany odpowiedni bank, do którego system wygeneruje URL przekierowania. W przypadku nieznanego prefiksu, Klient zostanie przekierowany do opcji "Mam konto w innym banku", czyli do strony w domenie autopay.eu, na której znajdą się dane do przelewu samodzielnego.

Jeśli numer rachunku ma być dodatkowo porównywany, w trybie PERSONAL_VERIFICATION, powinien zostać umieszczony w odrębnym polu “BankAccountNumber” w sekcji “params” w formacie 26 cyfr, bez prefiksu PL.

Initiate with iban – Parametry wejściowe

nazwa	typ i zakres danych	wymagany	opis
verificationId	string (^[a-zA-Z0-9-_]{1,64}+$)	nie	identyfikator weryfikacji nadany przez Partnera
email	string (zakres zgodny z EmailValidator z Apache Commons ver. 1.6)	tak – dla 1PLN, nie - dla PHOTO; tak – dla AIS, nie- dla MOBYWATEL	adres email klienta
type	enum: PERSONAL_VERIFICATION i COMPANY_VERIFICATION i DATA_HARVEST	tak	rodzaj weryfikacji do wykonania w Systemie
params	map<string, string>	tak	mapa parametrów wejściowych do weryfikacji przez Komponent danego typu. Lista parametrów jest określona przez typ weryfikacji (patrz pole "type")

nazwa	typ i zakres danych	wymagany	opis
partnerUuid	uuid	tak	tekstowy identyfikator Partnera
iban	string (PL + 26cyfrowy numer rachunku bankowego)	tak	IBAN z polskim prefiksem (PL). Prefiksy innych krajów nie są dozwolone.
component	enum: AIS, 1PLN, PHOTO, MOBYWATEL	nie	wybrany przez Partnera rodzaj komponentu do zainicjowania (gdy jest ich więcej niż jeden i Partner nie chce zdać sięna dobór automatyczny)

Initiate – Słownik kluczy mapy „params" dla wartości pola „type": PERSONAL_VERIFICATION

Wymagalność parametrów ustalana jest indywidualnie w procesie zakładania konta, zależy od zakresu przeprowadzanej weryfikacji oraz dodatkowych funkcjonalności, jakie procesowi mają towrzyszyć (np. zabezpieczanie dostępu do raportów w plikach PDF hasłem wysyłanym pod numer telefonu).

parametr	dopuszczalne wartości	wyjątki
firstName (Imię)	^[\p{L}\s]{1,32}+$	
lastName (Nazwisko)	^[\p{L}-'.\s]{1,64}+$	
pesel	\d{11}	Dla komponentu PHOTO walidacja jest uproszczona - nie może być pustej wartości
residenceAddressStreet (Ulica)	^[A-Za-z0-9ĘęÓóĄąŚśŁłŻżŹźĆćŃń\s-.]{1,64}+$	
residenceAddressHouseNumber (Numer domu)	^[A-Za-z0-9ĘęÓóĄąŚśŁłŻżŹźĆćŃń\s-./\]{1,10}+$	
residenceAddressStaircaseNumber (Numer klatki)	^[A-Za-z0-9ĘęÓóĄąŚśŁłŻżŹźĆćŃń\s-./\]{1,10}+$	
residenceAddressFlatNumber (Numer mieszkania)	^[A-Za-z0-9ĘęÓóĄąŚśŁłŻżŹźĆćŃń\s-./\]{1,10}+$	
residenceAddressPostalCode (Kod pocztowy)	^[0-9]{2}-[0-9]{3}$	
residenceAddressCity (Miasto)	^[A-Za-z0-9ĘęÓóĄąŚśŁłŻżŹźĆćŃń\s-.()]{1,64}+$	
phoneNumber (Numer telefonu)	^((\+|00)?((?!00)\d{2}))?\d{9}$	

parametr	dopuszczalne wartości	wyjątki
bankAccountNumber (Numer rachunku)	^([0-9]{26})$	
idDocumentNumber (Numer dokumentu tożsamości)	^[A-Z]{3}\d{6}$	Dla komponentu PHOTO walidacja jest uproszczona - nie może być pustej wartości
idDocumentExpiryDate (Data ważności dokumentu tożsamości)	Data z przyszłości w formacie YYYY-MM-DD	

Initiate – Słownik kluczy mapy „params" dla wartości pola „type": COMPANY_VERIFICATION

Wymagalność parametrów ustalana jest indywidualnie w procesie zakładania konta, zależy od zakresu przeprowadzanej weryfikacji oraz dodatkowych funkcjonalności, jakie procesowi mają towrzyszyć (np. zabezpieczanie dostępu do raportów w plikach PDF hasłem wysyłanym pod numer telefonu).

parametr	dopuszczalne wartości
companyName (Nazwa firmy)	^.{1,150}+$
nip (Numer NIP)	^\d{10}$
regon (Numer REGON)	^(\d{9}|\d{14})$
bankAccountNumber (Numer rachunku)	^([0-9]{26})$

Initiate – Słownik kluczy mapy „params" dla wartości pola „type": DATA_HARVEST

parametr	dopuszczalne wartości
phoneNumber (Numer telefonu)	^((\+|00)?((?!00)\d{2}))?\d{9}$

Pole phoneNumber jest wymagane w komponencie AIS w sytuacji gdy Partner nie dokonuje weryfikacji adresów email swoich użytkowników. Pole nie jest wymagane dla komponentu MOBYWATEL.

Initiate – Parametry wyjściowe

nazwa	typ	wymagany	opis
redirectUrl	string	tak	Adres URL, na który należy przekierować klienta
orderUuid	uuid	tak	Identyfikator weryfikacji nadany przez System
status	enum	nie	Status odpowiedzi. Przyjmuje wartości: OK i ERROR
description	string	nie	Dodatkowy komentarz związany z akcją. Komunikat informacyjny dla status=OK, komunikat błędu dla status=ERROR

Initiate – przykład żądania i odpowiedzi

Request:

```
{
  "partnerUuid": "cc955e86-…65d2",
  "type": "PERSONAL_VERIFICATION",
  "email": "jan@example.com",
  "params": {
    "firstName": "Jan",
    "lastName": "Niezbędny",
    "residenceAddressStreet": "Ciemna",
    "residenceAddressHouseNumber": "1",
    "residenceAddressPostalCode": "89-999",
    "residenceAddressCity": "Grodkowo",
    "bankAccountNumber": "72249000052663617643733450"
  }
}
```

Response:

```
{
  "status": "OK",
  "description": null,
  "hash": null,
  "redirectUrl": "https://id-hub-accept.bm.pl/api/verification/v1.0/start/U5F9VMY8WV",
  "orderUuid": "2ed3575a-37a6-487a-a993-b753b6e4e607"
}
```

Initiate - przykład żądania dla weryfikacji firmowej (1 PLN)

```
{
  "partnerUuid": "0aaa4440-1g1g-345t-8877-g7jj5781d1d1d",
  "email": "test@autopay.pl",
  "component": "1PLN",
  "type": "COMPANY_VERIFICATION",
  "params": {
    "companyName": "Autopay",
    "nip": "5851351185",
    "regon": "191781561",
  },
  "verificationId": "test_autopay"
}
```

Inicjowanie dwuetapowej weryfikacji w komponentach 1 PLN i AIS w PKO BP

Aby pozyskać dane z banku PKO BP w komponencie AIS, w pierwszej kolejności należy przeprowadzić weryfikację w komponencie 1 PLN w tym banku. Dwuetapowość procesu w tym przypadku wynika z braku pełnych danych osobowych, udostępnianych przez PKO BP w ramach usługi AIS. Brakujące dane

można pozyskać z komponentu 1 PLN, a łącznikiem pomiędzy obydwoma zestawami danych będzie numer rachunku bankowego. Powiązanie danych ułatwi nadanie w inicjacji tego samego VerificationID dla obu procesów weryfikacyjnych. Proces może być wykorzystany w modelu whitelabel.

Kolejność kroków w procesie dwuetapowym w PKO BP:

1. Inicjacja weryfikacji w komponencie 1 PLN z podaniem imienia i nazwiska Klienta.
2. Wykonanie przelewu przez Klienta.
3. Porównanie imienia i nazwiska z inicjacji oraz z rachunku bankowego.
4. Zwrócenie wyniku porównania wraz z pozyskanymi danymi. W przypadku pozytywnego wyniku, powinien być zainicjowany drugi etap (weryfikacja AIS).
5. Inicjacja weryfikacji w komponwncie AIS z podaniem NRB Klienta, pozyskanego w pierwszym etapie (1 PLN).
6. Pobranie danych transakcyjnych oraz NRB z banku.
7. Porównanie NRB z inicjacji z NRB pozyskanym z rachunku metodą AIS. W przypadku otrzymania kilku rónych NRB w ramach jednego procesu, sprawdzane jest podobieństwo któregokolwiek udostępnionego NRB z NRB podanym w inicjacji.
8. Zwrócenie wyniku porównania wraz z pozyskanymi danymi.
9. Partner posiada dwa zestawy danych, które może ze sobą powiązać na podstawie nadanego VerificationID oraz NRB.

Krok 1. Inicjacja weryfikacji 1 PLN

nazwa	typ i zakres danych	wymagany	opis
verificationId	string (^[a-zA-Z0-9-_]{1,64}+$)	nie (zalecany)	identyfikator weryfikacji nadany przez Partnera
email	string (zakres zgodny z EmailValidator z Apache Commons ver. 1.6)	tak	adres email klienta
type	enum: PERSONAL_VERIFICATION	tak	rodzaj weryfikacji do wykonania w Systemie
params	map<string, string>	tak	imię, nazwisko
partnerUuid	uuid	tak	tekstowy identyfikator Partnera
bankId	integer (wartość: 3)	tak	id banku PKO BP
component	enum: 1PLN	tak	

Krok 2. Inicjacja weryfikacji AIS

nazwa	typ i zakres danych	wymagany	opis
verificationId	string (^[a-zA-Z0-9-_]{1,64}+$)	nie (zalecany)	identyfikator weryfikacji nadany przez Partnera
email	string (zakres zgodny z EmailValidator z Apache Commons ver. 1.6)	tak	adres email klienta
type	enum: PERSONAL_VERIFICATION	tak	rodzaj weryfikacji do wykonania w Systemie

nazwa	typ i zakres danych	wymagany	opis
params	map<string, string>	tak	NRB
partnerUuid	uuid	tak	tekstowy identyfikator Partnera
bankId	integer (wartość: 3)	tak	id banku PKO BP
component	enum: AIS	tak	

Initiate – przykład żądania i odpowiedzi w przypadku dwuetapowej weryfikacji w PKO BP

Request 1:

```
{
  "partnerUuid": "<partnerUuid>",
  "type": "PERSONAL_DATA",
  "email": "jan@example.com",
  "verificationId": "ABC1234",
  "component": "1PLN",
  "bankId": 3,
  "params": {
    "firstName": "jan",
    "lastName": "niezbędny"
  }
}
```

Response 1:

```
{
  "status": "OK",
  "description": null,
  "hash": null,
  "redirectUrl": "https://id-hub-accept.bm.pl/api/verification/v1.0/start/U5F9VMY8WV",
  "orderUuid": "2ed3575a-37a6-487a-a993-65h3b6e4e607"
}
```

Request 2:

```
{
  "partnerUuid": "<partnerUuid>",
  "type": "PERSONAL_DATA",
  "email": "jan@example.com",
  "verificationId": "ABC1234",
  "component": "AIS",
  "bankId": 3,
  "params": {
    "bankAccountNumber": "12345678901234567890123456"
  }
}
```

Response 2:

```
{
  "status": "OK",
  "description": null,
  "hash": null,
  "redirectUrl": "https://id-hub-accept.bm.pl/api/verification/v1.0/start/U5F9VMY8WV",
  "orderUuid": "2ed3575a-37a6-487a-a993-kju3b6e4e766"
}
```

Start

GET /api/verification/v1.0/start/<code>

Metoda służy przekierowaniu klienta do Komponentu weryfikacyjnego w celu kontynuacji procesu sprawdzenia tożsamości. Przekierowanie powinno być wykonane metodą http GET.

Jedynym parametrem metody jest unikalny identyfikator nadany przez System w metodzie initiate.

Zaimplementowanie obsługi tej metody nie jest obligatoryjne. System Partnera może całkowicie polegać na adresie, jaki pojawił się w polu „redirectUrl", odpowiedzi na żądanie inicjacji weryfikacji.

Metoda start jest metodą jednorazową. Klient tylko jeden raz może zostać przekierowany do komponentu, po czym kod ulega przeterminowaniu.

Metoda Start dla komponentu mObywatel

Start GET /verification/v1.1/start/mobywatel/{code}

POST /verification/v1.0/qr-code/refresh/mobywatel

Metoda dedykowana dla komponentu MOBYWATEL. Służy do pozyskania kodu QR (opcjonalnie również kodu 6-cyfrowego), który powinien być wyświetlony Klientowi na stronie Partnera. Kod QR jest udostępniany pod adresem URL (Uwaga! Nie jest to link do przekierowania Klienta). Dodatkowo jest dostarczany kod możliwy do skopiowania i wklejenia w aplikacji mobilnej. Jest to rozwiązanie dla ścieżki Klienta w formie mobile, gdzie kod QR nie powinien być wyświetlany. Odświeżenie przeterminowanego kodu polega nie na zainicjowaniu nowej weryfikacji, a na wywołaniu metody /verification/v1.0/qr-code/refresh/mobywatel. W odpowiedzi ID HUB zwróci nowy kod, również z określonym terminem ważności.

Metoda Start i Refresh dla komponentu mObywatel - przykład żądania (dla metody Refresh) i odpowiedzi (dla metod Start i Refresh)

Request:

```
{
    "partnerUuid": "cc955e86-f78f-45fd-a6c8-115ae2be65d2",
    "orderUuid": "2bc300b6-ec61-481f-a51f-114f112e9c63"
}
```

Response:

```
{
status  "OK"
description     null
redirectUrl
"https://id-hub-accept.bm.pl/api/verification/v1.1/start/mobywatel/RTRHJZ6NBH"
orderUuid       "5db98409-ebf4-45ec-ac53-1c5yh64573b1"
}
```

Zawartość rediredtUrl:

```
{
qrCode
"iVBORw0KGgoAAAANSUhEUgAAAfQAAAH0AQAAAADjreInAAADeklEQVR4Xu2aMW7rMBBEabhw6PoKDqaczQfBD
I6cIwP2dmKclM8AP+AF8sZhrSy32rZrhYKUn5V/pMbaRP5ttIn8y3kT6ZbyN9Mt9G+mS+jfTJfDvJH20ifzbaR
P5rk8U9WU88eVMWzLOZSXc9meGK26mDc/Dn/jWqBS6vri8ZLSrUD3WvWV5qhadDcPmecyAA+nX3j8UXazspiZlIM
tEnBXTubNQ+Px8v/8SIhyK9PD/zXBvPlReQRPtWmXc5Za1MpzNHjz5kfjufKYmYJ2mfD/rKsAmedinsvxfFXwd3X
qb7ZV5s0Pw38RM89lo1FkSfWCNDLfRiTz/5UP/1d7n/jSt2gLLWfNz/oUF63cvPlB+DJeILM4HaV234+zXvpSzM9
IQO76gcO8VvOH8mW8qJlnHpctnA7dU1RVwrSOIubNj8PjS0UWpG2aeASekH6trdw8ZJ7L0Tz05v/5oUkj5o+igMr
ZN/43b/5IHv5fY1lbmf4Nyn+ZP8ybP5gvx/B/1QLspPlj8z+qbs8yb34QXp5eMH/wOE0qQ/4NQinkZvPmuYzArxD
+/xJQ9f+FVav/U7Ry8+YH4pOOlckYS5XsGV39x/nDvPnD+aKAWCrpo0Xhp2c7St8v5s2Pwme25/mhpo2tqoK/q5W
DZxS52bx5aBC++D+xJ8P/GZlRKuYPQjv/J14Q8+bH4LmD6vxRoF0paTd/fO3f5s0fx4f/S+b05HgBLXD6if0bCZv
/Y2ve/CD82p4VlL3B523+qFVxQV51FDGv1fyhPEz/5v+A0pZZu3raSpk3PwqP8eIO0+vvHzlGaVTNGxTzxzf+N2/
+SP5WMlOMF5BKYVIm9OT3i4hCeJZ588Pw+2PZe0l1/oDwgBkNfm3l5s2Pw+MYTTve9GL+iK6eA6ql2ODNmx+E3wk
87Q2n76/CZ5TSAxDN5s0HtuoQ/iljF01o5RIgrA9G4yrUBJYyb57L8Twz9dL3gf6tY/ifOiNV0Sf9HwnmuZi/Kf0
4Xsey9wuTMt//ru9VpbWVmzc/Hk9FKUQ1Sgek03Z+Nm9+GP6GiPyfC1Tnj6tuhSjkmjc/Ds91M72OMX8UCKL/uUv
xgmje/DB81bQ5fcFvduo0af6Ylbt7gHnzA/D/LPNtpE/m20ifzLeRPplvI30y30b6ZL6N9Ml8G+mT+TbSJ/NtpE+
/5v8AWojQj+GRP14AAAAASUVORK5CYII="
qrCodeText      "8;D;1;;;811;;1732622229639;1732622230;1732622290;5db98409-ebf4-45ec-
ac53-1c5a2c4573b1;0;1100002;;"
orderUuid       "5db98409-ebf4-45ec-ac53-1c5yh64573b1"
validTo "2020-11-26 12:58:10"
}
```

Przykład struktury dla odpowiedzi z endpointu https://id-hub-accept.bm.pl/api/verification/v1.1/start/mobywatel/{code}' z rozszerzoną opcją obsługi kodów 6-cyfrowych:

```
{
qrCode
"iVBORw0KGgoAAAANSUhEUgAAAfQAAAH0AQAAAADjreInAAADeklEQVR4Xu2aMW7rMBBEabhw6PoKDqaczQfBD
I6cIwP2dmKclM8AP+AF8sZhrSy32rZrhYKUn5V/pMbaRP5ttIn8y3kT6ZbyN9Mt9G+mS+jfTJfDvJH20ifzbaR
P5rk8U9WU88eVMWzLOZSXc9meGK26mDc/Dn/jWqBS6vri8ZLSrUD3WvWV5qhadDcPmecyAA+nX3j8UXazspiZlIM
tEnBXTubNQ+Px8v/8SIhyK9PD/zXBvPlReQRPtWmXc5Za1MpzNHjz5kfjufKYmYJ2mfD/rKsAmedinsvxfFXwd3X
qb7ZV5s0Pw38RM89lo1FkSfWCNDLfRiTz/5UP/1d7n/jSt2gLLWfNz/oUF63cvPlB+DJeILM4HaV234+zXvpSzM9
IQO76gcO8VvOH8mW8qJlnHpctnA7dU1RVwrSOIubNj8PjS0UWpG2aeASekH6trdw8ZJ7L0Tz05v/5oUkj5o+igMr
ZN/43b/5IHv5fY1lbmf4Nyn+ZP8ybP5gvx/B/1QLspPlj8z+qbs8yb34QXp5eMH/wOE0qQ/4NQinkZvPmuYzArxD
+/xJQ9f+FVav/U7Ry8+YH4pOOlckYS5XsGV39x/nDvPnD+aKAWCrpo0Xhp2c7St8v5s2Pwme25/mhpo2tqoK/q5W
DZxS52bx5aBC++D+xJ8P/GZlRKuYPQjv/J14Q8+bH4LmD6vxRoF0paTd/fO3f5s0fx4f/S+b05HgBLXD6if0bCZv
/Y2ve/CD82p4VlL3B523+qFVxQV51FDGv1fyhPEz/5v+A0pZZu3raSpk3PwqP8eIO0+vvHzlGaVTNGxTzxzf+N2/
+SP5WMlOMF5BKYVIm9OT3i4hCeJZ588Pw+2PZe0l1/oDwgBkNfm3l5s2Pw+MYTTve9GL+iK6eA6ql2ODNmx+E3wk
87Q2n76/CZ5TSAxDN5s0HtuoQ/iljF01o5RIgrA9G4yrUBJYyb57L8Twz9dL3gf6tY/ifOiNV0Sf9HwnmuZi/Kf0
4Xsey9wuTMt//ru9VpbWVmzc/Hk9FKUQ1Sgek03Z+Nm9+GP6GiPyfC1Tnj6tuhSjkmjc/Ds91M72OMX8UCKL/uUv
xgmje/DB81bQ5fcFvduo0af6Ylbt7gHnzA/D/LPNtpE/m20ifzLeRPplvI30y30b6ZL6N9Ml8G+mT+TbSJ/NtpE+
/5v8AWojQj+GRP14AAAAASUVORK5CYII="
qrCodeText      "8;D;1;;;811;;1732622229639;1732622230;1732622290;5db98409-ebf4-45ec-
ac53-1c5a2c4573b1;0;1100002;;"
orderUuid       "5db98409-ebf4-45ec-ac53-1c5yh64573b1"
validTo "2020-11-26 12:58:10"
}
```

```
4Xsey9wuTMt//ru9VpbWVmzc/Hk9FKUQ1Sgek03Z+Nm9+GP6GiPyfC1Tnj6tuhSjkmjc/Ds91M72OMX8UCKL/uUv
xgmje/DB81bQ5fcFvduo0af6Ylbt7gHnzA/D/LPNtpE/m20ifzLeRPplvI30y30b6ZL6N9Ml8G+mT+TbSJ/NtpE+
/5v8AWojQj+GRP14AAAAASUVORK5CYII="
qrCodeText      "8;D;1;;;811;;1732622229639;1732622230;1732622290;5db98409-ebf4-45ec-
ac53-1c5a2c4573b1;0;1100002;;"
orderUuid       "5db98409-ebf4-45ec-ac53-1c5yh64573b1"
validTo "2020-11-26 12:58:10"
mobywatel6DigitCode     "123456"
}
```

Result

/api/verification/v3.0/result

Result – Parametry wejściowe

Metoda służy pobraniu wyniku weryfikacji. Odpowiedzią jest dokument ze statusem weryfikacji, informacją o użytym Komponencie oraz odnośnikami do raportów, o ile wybrany Komponent został skonfigurowany tak, by dodatkowe informacje o kliencie obliczyć i dostarczyć.

W każdym komponencie weryfikacyjnym istnieje możliwość rezygnacji z przeprowadzenia weryfikacji. Jeżeli klient jawnie dokona takiego wyboru lub też porzuci proces na określony w konfiguracji przedział czasu, weryfikacja w Systemie otrzymuje status ABANDONED. W takiej sytuacji odpowiedź z endpointu nie zawiera żadnych dodatkowych danych o kliencie i jego danych. ID-HUB dostarcza funkcjonalność ręcznej zmiany wyniku weryfikacji poprzez Panel Merchanta ID HUB.

W trybie DATA_HARVEST należy inaczej interpretować pole obiektu odpowiedzi - „result". Znacząca staje się tylko wartość PENDING, która stanowi, iż wynik pobierania danych nie jest jeszcze gotowy. Gdy dane uda się pobrać, pole „result" będzie miało wartość POSITIVE i będzie oznaczać, że załączony do obiektu odpowiedzi raport jest możliwy do pobrania.

Result – Parametry wejściowe

nazwa	typ	wymagany	opis
orderUuid	uuid	tak	Identyfikator weryfikacji
partnerUuid	uuid	tak	identyfikator Partnera

Result – Parametry wyjściowe

nazwa	typ	wymagany	opis
result	enum	nie	Status weryfikacji. Dopuszczalne wartości: ABANDONED – klient porzucił i nie ukończył procesu weryfikacji; REJECTED_BY_USER - klient jawnie zrezygnował z dalszej weryfikacji Ten wynik jest możliwy do uzyskania jedynie w modelu paywall, gdzie znajduje się przycisk "Rezygnuję"; w przypadku wyboru tej opcji Klient jest przekierowany na stronę rezygnacji lub - jeśli ta nie została zdefiniowana - stronę błędu; POSITIVE - dane klienta zweryfikowane jako zgodne; NEGATIVE - dane klienta są niezgodne
verificationId	string	nie	Identyfikator weryfikacji nadany przez Partnera
systemsUsed	enum	tak	Użyty Komponent. Dopuszczalne wartości: AIS; 1PLN; PHOTO; MOBYWATEL
resultDetails	map<string, string>	tak	Mapa parametrów otrzymanych w metodzie initiate, w formacie <klucz>:<status>, gdzie „klucz" zgodny jest z kluczami w mapie parametrów metody initiate, a „status" to pole słownikowe enum[POSTIVIE|NEGATIVE], które określa, czy wartość wejściowa została zweryfikowania, czy nie
data	map<string, object>	tak	Mapa obiektów z zestawem danych zadeklarowanych w metodzie /initiate i pozyskanych z komponentu weryfikacyjnego (użytych do przeprowadzenia weryfikacji oraz dodatkowych danych przekazywanych bez porównania). W przypadku braku zdefiniowania konkretnych pól, zwracane są wszystkie pozyskane i obsługiwane dane.
dataComponent	map<string, object>	tak	Mapa obiektów z zestawem danych pozyskanych z komponentu weryfikacyjnego. MOBYWATEL: pełny zestaw danych otrzymanych z Systemu mObywatel PHOTO: pusty obiekt AIS: BANK_ACCOUNT_NUMBER, FULL_ADDRESS oraz FULL_NAME 1PLN: niepocięte dane nadawcy przelewu

nazwa	typ	wymagany	opis
addons	map<string, string>	nie	Mapa dodatkowych parametrów zwróconych przez komponenty weryfikacyjne. Mogą to być np. adresy do wygenerowanych raportów. Szczegółowe listy zwracanych parametrów znajdują się w paragrafach opisujących komponenty weryfikacyjne MOBYWATEL: adres do pobrania pełnego zestawu danych w oryginalnej strukturze z Systemu mObywatel; adres zdjęcia, jeśli jest zawarte w pobranym zestawie danych PHOTO: verificationProblems, bioStatus, documentStatus, overallStatus AIS: adresy do wygenerowanych raportów 1PLN: pusty obiekt
status	enum	nie	Status odpowiedzi. Dopuszczalne wartości: OK - weryfikacja zakończona poprawnie PENDING – weryfikacja nie jest jeszcze gotowa i należy ponowić zapytanie za kilka sekund; ERROR – błąd weryfikacji. Przyczyną błędu mogło być np. utracone połączenie z bankiem Klienta
description	string	nie	Dodatkowy komentarz związany z akcją. Dla statusu=OK komunikat informacyjny. Dla statusu=ERROR komunikat błędu

Result - Przykład żądania i odpowiedzi

Request:

```
{
    "partnerUuid": "cc955e86-f78f-45fd-a6c8-615ae2be65d2",
    "orderUuid": "2bc300b6-ec61-481f-a51f-114f662e9c63"
}
```

Response dla komponentu AIS:

```
{
  "status": "OK",
  "description": null,
  "result": "NEGATIVE",
  "verificationId": null,
  "systemsUsed": [
    "AIS"
  ],
  "resultDetails": {
    "firstName": "NEGATIVE",
    "lastName": "NEGATIVE",
    "residenceAddressPostalCode": "NEGATIVE",
    "residenceAddressStreet": "NEGATIVE",
    "residenceAddressHouseNumber": "NEGATIVE",
    "bankAccountNumber": "POSITIVE",
    "residenceAddressCity": "NEGATIVE"
  },
```

```
  "data": {
    "provided": {
      "firstName": "Nowak",
      "lastName": "Jan",
      "city": "Gdańsk",
      "street": "Grunwladzka",
      "bankAccountNumber": "542490000545251587838726 90",
      "postCode": "80-180",
      "streetHouseNumber": "3"
    },
    "obtained": {
      "city": "warszawa",
      "street": "dobra",
      "bankAccountNumber": [
        "542490000545251587838726 90"
      ],
      "postCode": "01-100",
      "individuals": [
        {
          "firstName": "tomek",
          "lastName": "widelec"
        }
      ],
      "streetHouseNumber": "1"
    }
  },
  "addons": {
   "financialAggregatedByMonthsReportUrl":
"https://id-hub-accept.bm.pl/ais/report/2f9a6e5d-5ac2-4b00-a3c6-cffb9380ae9a/FINANCIAL_A
GGREGATED_BY_MONTHS",
    "rawTransactionsGroupedByBbanReportUrl":
"https://id-hub-accept.bm.pl/ais/report/2f9a6e5d-5ac2-4b00-a3c6-cffb9380ae9a/RAW_TRANSAC
TIONS_WITH_POST_BALANCE_GROUPED_BY_BBAN_STREAMED",
    "lambdaReportUrl":
"https://id-hub-accept.bm.pl/ais/v2/report/2f9a6e5d-5ac2-4b00-a3c6-cffb9380ae9a",
    "accountDataReportUrl":
"https://id-hub-accept.bm.pl/ais/report/2f9a6e5d-5ac2-4b00-a3c6-cffb9380ae9a/ACCOUNT_DAT
A"
  },
  "dataComponent": {
    "BANK_ACCOUNT_NUMBER": "542490000545251587838726 90",
    "FULL_ADDRESS": "ul. Dobra 1 01-100 Gdańsk",
    "FULL_NAME": "Tomek Widelec"
  }
}
```


Response dla komponentu 1PLN:


```
{
  "status": "OK",
  "description": null,
  "result": "NEGATIVE",
  "verificationId": null,
  "systemsUsed": [
    "1PLN"
  ],
  "resultDetails": {
    "firstName": "POSITIVE",
    "lastName": "POSITIVE",
    "residenceAddressPostalCode": "NEGATIVE",
```

```
    "residenceAddressStreet": "NEGATIVE",
    "residenceAddressHouseNumber": "POSITIVE",
    "residenceAddressFlatNumber": "NEGATIVE",
    "bankAccountNumber": "NEGATIVE",
    "residenceAddressCity": "NEGATIVE",
    "residenceAddressStaircaseNumber": "POSITIVE"
  },
  "data": {
    "provided": {
      "streetFlatNumber": "1",
      "firstName": "Jan",
      "lastName": "Kowalski",
      "city": "Sopot",
      "street": "Powstańców Warszawy",
      "postCode": "81-718",
      "bankAccountNumber": "93124059347537181120097148",
      "streetStaircaseNumber": "A",
      "streetHouseNumber": "6"
    },
    "obtained": {
      "streetFlatNumber": "3",
      "unseparatedData": "Jan Kowalski Jasna 6a/3 10-234 Warszawa",
      "city": "warszawa",
      "street": "jasna",
      "postCode": "10-234",
      "bankAccountNumber": [
        "96109010301793218160815294"
      ],
      "individuals": [
        {
          "lastName": "kowalski",
          "firstName": "jan"
        }
      ],
      "streetStaircaseNumber": "a",
      "streetHouseNumber": "6"
    }
  },
  "addons": {},
  "dataComponent": {
    "UNSEPARATED_DATA": "Jan Kowalski Jasna 6a/3 10-234 Warszawa"
  }
}
```


Response dla komponentu PHOTO:


```
{
  "status": "OK",
  "description": null,
  "result": "NEGATIVE",
  "verificationId": null,
  "systemsUsed": [
    "PHOTO"
  ],
  "resultDetails": {
    "firstName": "POSITIVE",
    "lastName": "POSITIVE",
  },
  "data": {
    "provided": {
```

```
      "firstName": "Jan",
      "lastName": "Kowalski"
    },
    "obtained": {
      "placeOfBirth": "WARSZAWA",
      "idDocumentIssueState": null,
      "gender": "M",
      "maidenName": "KOWALSKI",
      "fullName": "JAN KOWALSKI",
      "dateOfBirth": "1981-01-02",
      "thirdName": null,
      "individuals": [
        {
          "lastName": "KOWALSKI",
          "firstName": "JAN"
        }
      ],
      "idDocumentNumber": "ZZC108201",
      "idDocumentType": "IDENTITY_CARD",
      "nationality": "POL",
      "idDocumentExpiryDate": "2031-08-02",
      "idDocumentIssuingDate": "2021-08-02",
      "idDocumentIssueCountry": "Poland",
      "pesel": "81010200131",
      "secondName": null
    }
  },
"addons": {
    "verificationProblems": "printed_face_match_manually_entered_face_photo",
    "documentStatus": "VERIFIED",
    "bioStatus": "SUSPICIOUS",
    "overallStatus": "SUSPICIOUS"
  },
  "dataComponent": {}
}
```


Response dla komponentu MOBYWATEL:


```
{
  "status": "OK",
  "description": null,
  "result": "POSITIVE",
  "verificationId": null,
  "systemsUsed": [
    "MOBYWATEL"
  ],
  "resultDetails": {
    "firstName": "POSITIVE",
    "lastName": "POSITIVE"
  },
  "data": {
    "provided": {
      "lastName": "NOWAK",
      "firstName": "TERESA"
    },
    "obtained": {
      "placeOfBirth": "warszawa",
      "idDocumentType": "IDENTITY_CARD",
      "gender": "F",
      "idDocumentIssuingAuthority": "PREZYDENT M. ST. WARSZAWY",
```

```
      "idDocumentExpiryDate": "2030-04-15",
      "dateOfBirth": "1985-12-05",
      "pesel": "85120500779",
      "individuals": [
        {
          "firstName": "teresa",
          "lastName": "nowak"
        }
      ]
      "idDocumentNumber": ZOP843564
    }
  },
  "addons": {
    "resultRawDataProviderUrl":
"https://id-hubaccept.bm.pl/api/verification/v1.0/result/raw-dataprovider",
    "resultFaceImageUrl":
"https://id-hubaccept.bm.pl/api/verification/v1.0/result/id-document-faceimage"
},
"dataComponent": {
   "personalDataSet": [
    {
      "key": "LAST_NAME",
      "value": "NOWAK"
      },
      {
      "key": "PLACE_OF_BIRTH",
      "value": "WARSZAWA"
      },
      {
      "key": "GENDER",
      "value": "F"
      },
      {
       "key": "PESEL",
       "value": "85120500779"
      },
      {
      "key": "FIRST_NAME",
      "value": "TERESA"
      },
      {
      "key": "DATE_OF_BIRTH",
      "value": "1985-12-05"
      }
      ],
      "addressDataSet": [],
      "documentDataSet": [
      {
      "key": "ID_DOCUMENT_NUMBER",
      "value": "ZOP843564"
      },
     {
      "key": "ID_DOCUMENT_FACE_IMAGE",
      "value": "/9j/ +yZnLfTrt7+DSd70QWrRbv8 (…) 6oQnNW1BnFwB1j3bSC+//Z"
      },
      {
      "key": "ID_DOCUMENT_ISSUING_AUTHORITY",
      "value": "PREZYDENT M. ST. WARSZAWY"
      },
      {
      "key": "ID_DOCUMENT_EXPIRY_DATE",
      "value": "2030-04-15"
```

```
      },
      {
      "key": "ID_DOCUMENT_TYPE",
      "value": "IDENTITY_CARD"
      }
]
}
}
```

Health-Check

GET /api/monitoring/health-check

Metoda jest przeznaczona do testowania dostępności sieciowej API ID HUB. Odpowiedzią na żądanie powinny być słowo: OK i kod odpowiedzi http: 200. Każdy inny kod i każda inna odpowiedź oznacza wystąpienie problemów z działaniem ID-HUB.

Health-Check komponentów PHOTO i MOBYWATEL

GET {{host}}/api/monitoring/{{partnerUuid}}/health-check/components

Istnieje możliwość sprawdzenia statusu komponentów MOBYWATEL i PHOTO aktywnych na wskazanym koncie Partnera (PartnerUUID). Statusy mogą przyjmować wartości: ACTIVE - komponent działa poprawnie, bez żadnych zakłoceń INACTIVE - komponent jest aktualnie niedostępny MAINTENANCE ONGOING - komponent jest tymczasowo wyłączony z powodu zaplanowanych prac serwisowych UNSUPPORTED - health-check dla tego komponentu nie jest obsługiwany

Przykładowa odpowiedź dla Partnera z aktywnymi komponentami:

```
[
  {
    "systemCode": "MOBYWATEL",
    "status": "ACTIVE"
  },
  {
    "systemCode": "PHOTO",
    "status": "ACTIVE"
  },
]
```

Przykładowa odpowiedź dla Partnera, który nie posiada skonfigurowanych komponentów

```
[]
```

Przykładowa odpowiedź dla Partnera, który nie istnieje

```
{
    "status": "ERROR",
    "description": "invalid partner UUID"
}
```

BankList-notification (PUSH)

System oferuje mechanizm powiadamiania o zmianach na liście banków. Realizuje to przez notyfikację/push na ustalony adres.

Adres, na który system powinien wysyłać powiadomienia powinien zostać dostarczony przez stronę integrującą się z HUBem.

System wysyła puste żądanie (POST) i spodziewa się pustej odpowiedzi http z kodem 204.

Domyślnie HUB wysyła jedno powiadomienie i nie próbuje ponawiać go, nawet jeśli nie otrzymał kodu odpowiedzi http 204. Ponawianie powiadomień jest opcją, którą można włączyć na życzenie Partnera.

Result-notification (PUSH)

HUB może wysyłać do systemu Partnera powiadomienie, informujące o możliwości pobrania, gotowego wyniku weryfikacji.

Aby wykorzystać tę funkcjonalność, dzięki której znika konieczność cyklicznego odpytywania o rezultat weryfikacji, Partner musi przygotować endpoint, który obsłuży żądanie http POST.

nazwa	typ	wymagany	opis
orderUuid	uuid	tak	Identyfikator weryfikacji
partnerUuid	uuid	tak	Identyfikator partnera

Na życzenie Partnera notyfikacja PUSH może być zabezpieczona metodą HMAC.

Result-notification (PUSH) – Odpowiedź

Odpowiedzią na takie żądanie powinna być pusta odpowiedzi, z kodem http 200. Po otrzymaniu jej, System uznaje powiadomienie za dostarczone. W przeciwnym wypadku ponawia żądanie z malejącą częstotliwością, aż do momentu otrzymania oczekiwanej odpowiedzi.

UWAGA: Lista pól notyfikowanych do Partnera może być zwiększana. Prosimy o uwzględnienie w implementacji możliwości płynnego pojawiania się nowych pól w obiekcie powiadomienia.

Malejąca częstotliwość oznacza ponowienia w kolejnych iteracjach, gdzie każda kolejna próba odbywa się po dłuższym odstępie czasu niż poprzednia. Odstępy są liczone zgodnie z ciągiem Fibonacciego, zgodnie z poniższą tabelą:

Próba ponownego dostarczenia	Odstęp w minutach od poprzedniej próby
1	1
2	2
3	3
4	5

Próba ponownego dostarczenia	Odstęp w minutach od poprzedniej próby
5	8
6	13
…	t = (t-1) + (t-2)

Powrót klienta z komponentu do systemu Partnera

Klient kończąc proces weryfikacji w HUBie jest przekierowywany do systemu Partnera na ustalony adres powrotu. Wyjątkiem jest komponent mObywatel, w którym nie pojawiają się przekierowania na strony Autopay.

Są trzy adresy powrotu:

1. Adres sukcesu – klient kierowany jest po poprawnie zakończonym procesie. Uwaga – negatywną weryfikację nadal traktujemy jako poprawnie zakończony proces.
2. Adres porażki – klient kierowany jest w momencie zajścia sytuacji kryzysowej, błędu systemu. Weryfikacji nie udało się przeprowadzić.
3. Adres cofnięcia - klient jest kierowany na ten adres w przypadku wyboru opcji "wstecz" w przeglądarce, po przekierowaniu do banku. Jeśli nie został zdefiniowany, jest tosami z adresem błędu.

Powrót do systemu partnera wykonywany jest na adresy dokładnie taki, jak podano w konfiguracji w karcie wdrożenia.

Wzbogacenie adresu URL o identyfikator weryfikacji

Jest możliwość wzbogacenia adresu URL o jeden z poniższych dynamicznych parametrów:

parametr	opis
orderUuid	Identyfikator weryfikacji
verificationId	Identyfikator weryfikacji nadany przez Partnera w fazie inicjacji weryfikacji

UWAGA: Do adresu powrotu dodajemy tylko jeden z powyższych parametrów, nie łączymy ich.

Ręczna zmiana wyniku weryfikacji

W pewnych sytuacjach wynik weryfikacji może być dyskusyjny. Dane inicjacyjne skrzyżowane z danymi pozyskanymi ze źródeł mogą skutkować rezultatami trudnymi w ocenie nie tylko dla systemu informatycznego, ale także dla człowieka. Przykładami takich sytuacji są m.in.: próby interpretacji imion, nazwisk i adresów klientów spoza Polski, współwłasności rachunków bankowych, błędne lub niestandardowe notacje adresowe wprowadzone do systemów źródłowych.

Wynik weryfikacji zdeterminowany przez dane przetworzone w sposób niezgodny z oczekiwaniem Partnera ma możliwość być zmieniony na oczekiwany, przez pracownika operacyjnego po stronie Partnera.

Funkcjonalność ręcznego zmieniania wyników weryfikacji jest dostępna w Panelu Merchanta ID HUB. W celu uzyskania dostępu do Panelu, naley się skontaktować z Opiekunem.

Kształt odpowiedzi metody Result z włączoną możliwością ręcznej zmiany wyniku weryfikacji

```
{
  "status": "OK",
  (...)
  "addons": {
    "resultChangeUrl": "https://id-hub.bm.pl/v/:uuid1/:uuid2"
   }
}
```

W węźle "addons" w obiekcie zwracanym przez metode Result znajduje się pole "resultChangeUrl".

Po stwierdzeniu niewłaściwego wyniku weryfikacji, należy wejść pod pozyskany adres URL. Jest on unikalny, przypisany do konkretnej weryfikacji. Po załadowaniu się widoku należy uwierzytelnić swoją tożsamość przy pomocy loginu i hasła. Udane logowanie przekieruje nas na widok podsumowania wyniku i formatki zmiany wyniku weryfikacji.

Zmiana wyniku możliwa jest tylko raz. Możliwe kierunki zmiany wyniku to przejście ze stanu POSITIVE na NEGATIVE i odwrotnie. Nie ma możliwość zmiany innych statusów, np. ABANDONED.

Przeprowadzona zmiana wyniku weryfikacji generuje powiadomienie PUSH pod wskazany przez Partnera adres do odbierania powiadomień.

Kształt odpowiedzi metody Result po przeprowadzeniu ręcznej zmiany wyniku weryfikacji

```
{
  "status": "OK",
  (...)
  "addons": {
    "resultSetManually": true,
    "resultSetManuallyAt": "2022-10-10 12:34:22",
    "resultSetManuallyBy": "jkowalski"
   }
}
```

W węźle "addons" w obiekcie zwracanym przez metodę Result znajdują się pola: "resultSetManually", "resultSetManuallyAt", "resultSetManuallyBy".

Figurowanie powyższych kluczy w węźle "addons" niesie informację, że weryfikacja została zmodyfikowana ręcznie, wskazuje datę kiedy do modyfikacji doszło oraz jej autora.

Potwierdzenia wykonanych weryfikacji

System oferuje funkcjonalność dostarczania potwierdzeń wykonanych weryfikacji 1PLN i AIS.

Potwierdzenie wykonanej weryfikacji jest podpisanym cyfrowo dokumentem pdf, który wysyłany jest

pocztą elektroniczną pod ustalony przez Partnera adres email (adres obsługiwany przez Partnera; nie adres Klienta).

Generacja potwierdzenia AIS jest wykonywana automatycznie, do 24 godzin od momentu zakończenia weryfikacji, po poprawnym pobraniu raportów przy użyciu metody /result.

Potwierdzenie weryfikacji 1 PLN jest generowane po złożeniu zamówienia w panelu Scribe. Zamówienie może być złożone w dowolnym momencie po przeprowadzeniu weryfikacji.

Rekomendacje deweloperskie

Implementując obiekty transportowe do API Systemu, sugerujemy taką konfigurację narzędzia serializacji i deserializacji JSON, aby tolerowała ona pojawianie się nowych pól lub ich brak. Rozwój komponentów weryfikacyjnych może skutkować pojawianiem się nowych raportów tudzież obiektów dostarczających szczegółowych danych o kliencie. Podstawową odpowiedzią na taki rozwój wypadków jest wersjonowanie API (widoczne w adresach URL), dopuszczamy jednak (nie chcąc doprowadzić do rozdrobnienia interfejsu) możliwość drobnych modyfikacji bez wersjonowania całych ścieżek.

Bezpieczeństwo

Sieć

Bezpieczeństwo sieciowe zapewniamy Partnerom poprzez udostępnienie usługi protokołem HTTPS. Każdy endpoint lub przekierowanie będą przedstawiać się sieciowo certyfikatem wystawionym dla domeny: *.bm.pl. Prócz obsługi ruchu poprzez HTTPS, mamy domyślnie ustawione filtrowanie adresów IP dla żądań przychodzących do API Systemu.

Jeśli te dwie metody zabezpieczenia sieciowego nie są dla Partnera wystarczające, dopuszczamy możliwość stworzenia dedykowanego tunelu sieciowego, którym aplikacja Partnera będzie mogła komunikować się z usługami Systemu. Zestawienie takiego tunelu wymaga odrębnych ustaleń i procesu, ponieważ nie jest to rozwiązanie dostępne w podstawowej konfiguracji.

Istnieje także możliwość kontroli ruchu między systemami Partnera i HUBem poprzez mechanizm tzw. "dwustronnego SSL-a". W tym modelu integracji system Partnera przedstawia się podpisanym przez Autopay kluczem. Wdrożenie tej metody zabezpieczania, podobnie jak zestawianie tunelu, wymaga odrębnych ustaleń między Partnerem, a Autopay.

Dane osobowe i raporty

ID-HUB działający w trybie AIS generuje raporty z danymi, które wysyła do systemu Partnera oraz do weryfikującego się użytkownika. Raporty te mogą zwierać dane osobowe takie jak imię i nazwisko, adres, historię transakcji z rachunku bankowego, etc. Kluczową dla bezpieczeństwa informacji jest kwestia dostarczenia tych informacji pod poprawny adres email.

Na etapie integracji z ID-HUB Partner winien określić czy adresy email jakie będzie przekazywał w momencie inicjacji weryfikacji są adresami potwierdzonymi przez użytkownika.

Jeśli tak, ID-HUB będzie wysyłał wiadomości mejlowe pod wskazane adresy z linkiem do strony oraz hasłem do raportów.

Jeśli Partner nie jest pewien czy podane przez użytkowników adresy są poprawne, administrator ID-HUB ustawia w konfiguracji odpowiedni tryb pracy. W rezultacie ustawienia go, ID-HUB w trakcie inicjacji weryfikacji będzie wymagał podania numeru telefonu. Po wygenerowaniu raportu, system wyśle email pod niezweryfikowany adres i SMS pod wskazany numer. W treści emaila będzie znajdować się link do strony www, na której można pobrać raporty. Strona będzie zabezpieczona hasłem, którego treść użytkownik otrzyma w SMSie.

Gdy użytkownik otworzy widok, na którym znajdzie linki do raportów ze swoimi danymi, będzie miał możliwość pobrać je przez 20 minut. Po upływie tego czasu linki staną się nieaktywne, aż do czasu ponownego załadowania strony i autoryzacji hasłem.

Uwierzytelnianie

Oprócz zabezpieczeń w warstwie transportowej, System stara się zabezpieczyć komunikację także w zakresie integralności przesyłanych danych. W tym celu przygotowane została metoda weryfikacji poprawności przesyłanych komunikatów o umownej nazwie „HMAC".

Polega na wyliczeniu z ciała żądania sumy kontrolnej przy użyciu jednej z obsługiwanych funkcji:

- HmacSHA256,
- HmacSHA512

Proces liczenia sumy kontrolnej angażuje tajny, specyficzny dla każdego z Partnerów klucz, który przekazywany jest razem z pozostałymi parametrami integracyjnymi.

Partner, który w procesie integracji wybierze metodę autentykacji HMAC, zobowiązany jest załączać w każdym requeście wykonywanym metodą inną niż metoda GET, dwa nagłówki:

- Hmac-Algorithm - zawierający nazwę funkcji użytej do wygenerowania sumy kontrolnej (wartość powinna zawierać się w podanej wyżej liście obsługiwanych funkcji)
- Hmac - zawierający wyliczoną wartość sumy kontrolnej

Rekomendujemy niniejszy sposób autoryzowania żądań z następujących powodów:

1. Obiekty transportowe nie muszą być wzbogacane o pole nieniosące treści biznesowej
2. Kolejność pól w obiektach transportowych jest bez znaczenia
3. Dane zabezpieczające żądania nie są elementem treści żądania, są więc nieco trudniejsze do podsłuchania/podejrzenia

Brak lub nieprawidłowa wartość nagłówka Hmac-Algorithm skutkuje kodem odpowiedzi 400.

Brak lub nieprawidłowa wartość nagłówka Hmac skutkuje kodem 401.

Przykład sposobu liczenia sumy kontrolnej (JAVA)

```
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.util.Base64;

class Authenticator {

    public static String calculateSignature(String algorithm, String secretKey, byte[]
requestPayload) {
        Mac mac = Mac.getInstance(algorithm);
```

```
        mac.init(new SecretKeySpec(secretKey.getBytes(), algorithm));
        mac.update(requestPayload);
        byte[] signature =  mac.doFinal();

        return Base64.getEncoder().encodeToString(signature);
    }
}
```

Przykład sposobu liczenia sumy kontrolnej (PHP)

```
<?php
    $code = hash_hmac('sha256', $requestPayload, $secretKey, true);
    echo base64_encode($code);
?>
```

Przykład sposobu liczenia sumy kontrolnej (C#)

```
using System;
using System.Security.Cryptography;
using System.Text;
public class Program
{
        public static void Main()
        {
                string secretKey = "mdk7G86HJVjhgUGh865434dsr5";
        // Podpis liczony jest dokładnie z tych bajtów, które są wysyłane w ciele
żądania.
        // Obiekt serializujemy do kompaktowej postaci - bez znaków nowej linii i
zbędnych spacji.
        string payload = "{\"partnerUuid\":\"11a122bb-a111-a22a-
eeee-22a222a2a2a2\",\"email\":\"test@autopay.pl\",\"type\":\"PERSONAL_VERIFICATION\",\"p
arams\":{\"firstName\":\"Jan\",\"lastName\":\"Kowalski\",\"residenceAddressStreet\":\"Po
wstańców
Warszawy\",\"residenceAddressHouseNumber\":\"6\",\"residenceAddressStaircaseNumber\":\"A
\",\"residenceAddressFlatNumber\":\"1\",\"residenceAddressPostalCode\":\"81-718\",\"resi
denceAddressCity\":\"Sopot\"},\"bankId\":\"25\",\"verificationId\":\"test\"}";

        string hmac = CalculateHmacSha512(secretKey, payload);
        Console.WriteLine("HMAC: " + hmac);
        // Oczekiwany wynik dla powyższego klucza i payloadu:
        // HMAC-SHA512 (Base64):
ThXWB6tA5aOiEmRMNy4yYUZi04b2T5pzKmcJVaK7YPDjn7EIjarImV3GQAqNyk/F+Hlit9dcopM4HOsyNao8/g==
        }

    public static string CalculateHmacSha512(string key, string data)
    {
        using (var hmacsha512 = new HMACSHA512(Encoding.UTF8.GetBytes(key)))
        {
            byte[] hashValue = hmacsha512.ComputeHash(Encoding.UTF8.GetBytes(data));
            return Convert.ToBase64String(hashValue);
        }
    }
}
```

WSKAZÓWKA: Przykłady w innych językach programowania znajdziesz na blogu Joe Kampschmidta.

UWAGA: Suma kontrolna liczona jest dokładnie z tych bajtów ciała żądania, które są wysyłane - serwer nie normalizuje ani nie przekształca payloadu przed weryfikacją. Podpis po stronie klienta musi być więc policzony z identycznej sekwencji bajtów, jaka trafia do wysyłki. Należy zwrócić szczególną uwagę na niewidoczne znaki nowej linii (`\n`, `\r`) oraz wcięcia, które prowadzą do różnych wyników podpisów po stronie klienta i serwera (skutkuje to kodem 401). Najbezpieczniej jest serializować obiekt do postaci kompaktowej - bez znaków nowej linii i zbędnych spacji oraz podpisywać dokładnie ten sam string, który jest wysyłany.

BasicAuth

Uwierzytelnianie żądań http za pomocą mechanizmu BasicAuth stosowane jest w wybranych elementach systemu, które System chce zabezpieczyć przed przypadkowym pobraniem danych przez niepowołanych użytkowników.

W bieżącej wersji dokumentacji i systemu za pomocą metody BasicAuth zabezpieczone zostały adresy pobierania raportów w komponencie AIS.

Login i hasło potrzebne do zbudowania nagłówka „Authorization" Partner otrzymuje w momencie rozpoczęcia integracji z Systemem, w karcie wdrożeniowej.

NONE

We wstępnej fazie integracji, w środowisku testowym, mamy możliwość wyłączenia mechanizmu kontroli integralności danych. Zalecamy stosowanie tej konfiguracji tylko w środowisku testowym, w początkowej fazie integracji.