
Santander Consumer Bank
ul. Strzegomska 42c
53-611 Wrocław
tel. 71 387 74 04
wsparcie@santanderconsumer.pl



Dokumentacja techniczna interfejsu (Web Service)
dla systemu sprzedaży internetowej
eRaty Santander Consumer Bank

2017-05-18
(Wersja_3.19)

Spis treści

1 Wstęp	4
1.1 Strony komunikacji	4
1.2 Jak przeprowadzić proces integracji bezpośrednio z SCB?	4
1.3 Sposoby implementacji połączenia z certyfikatem	6
1.3.1. PHP.	6
1.3.2. C#.	6
1.3.3. JAVA.....	7
1.4 Błędy operacji	7
2 Algorytm użycia metod	8
3 Metody.....	10
3.1 Metody konfiguracyjne	10
3.1.1 isActive	10
3.1.2 pobierzParametrySklepu	11
3.1.3 pobierzLinieKredytowe	12
3.1.4 pobierzSłowniki.....	13
3.1.5 pobierzKlauzule.....	15
3.1.6 pobierzKredytPunkty.....	16
3.2 Metody produkcyjne.....	17
3.2.1 zlozWniosek.....	17
3.2.2 zapiszRezygnacje	22
3.3 Metody informacyjne.....	22
3.3.1 wykonajSymulacje	22
3.3.2 sprawdzStanWniosku	24
3.3.3 sprawdzStanWnioskuIdZewn	25
3.3.4 getInfoWn	26
4 Słowniki	29
4.1 godziny	29
4.2 gtu	29
4.3 powodRezygnacji	30
4.4 sposobPodpisaniaUmowy	30
4.5 stanCywilny	30
4.6 statusMieszkania – nie używane	31
4.7 typDrugiegoDokumentu.....	31
4.8 wojewodztwo – nie używane	31

4.9 wykształcenie – nie używane.....	32
4.10 zdrojDochodu.....	32
4.11 statusWniosku i statusKredytu.....	33
4.11.1 eHP Kurier.....	33
4.11.2 eHP PL.....	34
4.12.3 eHP POS.....	34
5 Inne parametry słownikowe.....	35
5.1 idKlauzuli.....	35
5.2 potwierdzaTowary.....	35
5.2 systemZrodlowy.....	35
5.3 trybSklepu, trybWniosku.....	35
5.4 typProduktu.....	35

1 Wstęp

Niniejszy dokument jest specyfikacją protokołu komunikacji usługi eRaty Santander Consumer Bank z dowolnym innym systemem informatycznym Klientów do obsługi sprzedaży przy pomocy protokołu SOAP, czyli serwisów webowych (**WebServices**) poprzez publiczny Internet.

1.1 Strony komunikacji

W wymianie informacji uczestniczą dwie strony:

- aplikacja eRaty, pełniąca rolę serwera udostępniającego pewien zestaw funkcji, (nazywany w dalszej części **Systemem**),
- system informatyczny sklepu integrującego się z aplikacją eRaty (nazywany w dalszej części **Sklepem**).

Stroną aktywną jest **Sklep**, który wysyła żądanie do **Systemu** i oczekuje na odpowiedź. Adres do komunikacji z **WebService'em** to

<https://api.eraty.pl/eRatyService/eRatyService.asmx>

Istnieje możliwość prób integracji sklepu na testowym środowisku, które może zostać wystawione dla sklepu. Adres serwisu testowego to.

<https://testapi.eraty.pl/eRatyService/eRatyService.asmx>

1.2 Jak przeprowadzić proces integracji bezpośrednio z SCB?

Mamy dwa sposoby przeprowadzenia przez sklepy integracji bezpośrednio.

- PEŁNA INTEGRACJA – pełna integracja ze wszystkimi rodzajami metod wystawionymi w WS-ie dla kontrahentów.

- INTEGRACJA HYBRYDOWA – integracja poprzez eWniosek w części zawierania umów a używająca tylko metod informacyjnych WS-u w celu uzyskania statusu wniosku.

Niniejszy opis dotyczy głównie pełnej integracji.

Proces integracji bezpośrednio jest złożony i wymaga wykonania wielu czynności zarówno po stronie **Systemu** jak i **Sklepu**. Algorytm podłączania nowego **Sklepu** z integracją bezpośrednią jest następujący.

1. **Sklep** powinien podpisać umowę o współpracy z Santander Consumer Bank.
2. **Sklep** powinien zgłosić swojemu Opiekunowi chęć zintegrowania bezpośrednio z **Systemem**.
3. W oparciu o unikalny numer w **Systemie** zostaje utworzony punkt sprzedaży.
 - 3.1. Założony zostaje nowy **Sklep** w trybie pracy „**Testowy**”, na który **Sklep** będzie mógł składać wnioski testowe. Wnioski te nie będą procesowane w SCB jak produkcyjne, a jedynie wejdą do **Systemu** po to, by można było zweryfikować poprawność ich złożenia, a następnie będą anulowane. Założenie sklepu powoduje utworzenie numeru sklepu potrzebnego do logowania do **WebService'u**.
 - 3.2. **Sklep** będzie miał aktywowane kanały sprzedaży (typy produktów) zgodne z ustaleniami na poziomie Opiekuna Punktu Sprzedaży z Przedstawicielem **Sklepu** (należy pamiętać, że tylko w zakresie ustalonych kanałów **Sklep** będzie mógł przekazywać wnioski do **Systemu**).
 - 3.3. **Sklep** będzie miał skonfigurowane produkty kredytowe.
 - 3.4. W zależności od ustaleń **Sklep**, będzie miał włączony lub wyłączony wymóg potwierdzania dostępności towarów na poziomie każdego wniosku z osobna. **Uwaga:** punkty sprzedaży z wymogiem będą musiały przed każdym wnioskiem potwierdzić dostępność towaru jedynie w aplikacji **Panel Sklep**.

- 3.5. Nadanie dostępu do aplikacji **Panel Sklep**. Użytkownik jest bardzo ważny, aby rozpocząć pracę, gdyż trzeba w **Panelu Sklep** zdefiniować adresy email do obsługi procesu umów internetowych.
4. Po skonfigurowaniu nowego punktu sprzedaży w **Systemie**, Opiekun Sklepu przekaze Przedstawicielowi Sklepu następujące dane.
- 4.1. Unikalny login do **WebService'u**. W razie integracji z użyciem środowiska testowego login może być taki sam lub inny w zależności od konfiguracji.
- 4.2. Hasło do **WebService'u**. W razie integracji z użyciem środowiska testowego hasło do niego będzie inne niż na środowisko produkcyjne.
- 4.3. Numer **Sklepu** w systemie SCB.
- 4.4. Login i hasło dla użytkownika aplikacji **Panel Sklep**. W razie integracji z użyciem środowiska testowego zarówno login jak i hasło mogą być inne niż na produkcję.
- 4.5. Instrukcję integracji w wersji obowiązującej w danym momencie.
5. Dodatkowo po stronie **Sklepu** wymagane jest użycie certyfikatu w celu nawiązania komunikacji z **WebService'em**. Certyfikat **Sklepu**, hasło do certyfikatu oraz certyfikaty urzędów zostaną dostarczone przez Opiekuna Sklepu. Po instalacji certyfikatu w celu uniknięcia problemów z jego użyciem należy zaufać certyfikatowi urzędu wystawionego dla Banku, czyli Finanse Dla Domu. Instrukcja implementacji połączenia do WebService'u za pomocą certyfikatu dla różnych platform programistycznych została opisana w rozdziale 1.3.
- 5.1. **Sklep** może dokonać weryfikacji czy posiada poprawny certyfikat i czy **WebService** działa poprzez wywołanie metody **isActive**.
6. Pracownik **Sklepu** powinien zalogować się do aplikacji **Panel Sklep** (adres <https://sklep.eraty.pl/>) w celu weryfikacji i parametryzacji.
- 6.1. W zakładce „**Ustawienia**” w sekcji „**Lista kontaktów, na które będą przesyłane informacje o wnioskach ratalnych**” pracownik **Sklepu** powinien wprowadzić przynajmniej jeden aktualny adres mailowy, na który mają przychodzić komunikaty z kolejnych kroków przetwarzania wniosku kredytowego Klienta.
- 6.2. W zakładce „**Ustawienia**” w sekcji „**Produkty ratalne**” pracownik **Sklepu** może sprawdzić czy i jakie produkty ratalne ma połączone w **Sklepie**.
7. **Sklep** dokonuje integracji swojego oprogramowania sprzedażowego zgodnie z dostarczoną instrukcją integracji **eRatyService**. Wysyłka wniosków odbywa się za pomocą metody **zlozWniosek**.
8. Jeśli wnioski testowe trafiły do systemu centralnego bez błędów walidacji, które mogą być zwrócone w odpowiedzi metody **zlozWniosek**, to **Sklep** może sprawdzić ich status oraz jakość odebranych danych po stronie SCB za pomocą metod informacyjnych.
- 8.1. **sprawdzStanWniosku** – weryfikacja statusu wniosku oraz statusu kredytu w SCB znajdujących po numerze wniosku SCB otrzymanym w metodzie **zlozWniosek**.
- 8.2. **sprawdzStanWnioskuIdZewn** – weryfikacja statusu wniosku oraz statusu kredytu w SCB znajdujących po numerze zamówienia w sklepie.
- 8.3. **getInfoWn** – weryfikacja podstawowych danych wniosku w SCB znajdujących po numerze wniosku SCB.
9. Wszelkie ewentualne błędy albo problemy będą **Sklep** zgłasza do Opiekuna lub bezpośrednio na Helpdesk SCB – wsparcie@santanderconsumer.pl.
10. **Sklep** po wykonaniu integracji i zweryfikowania jej wyników zgłasza gotowość do pracy Opiekunowi Sklepu.
11. IT w SCB weryfikuje prawidłowość przesyłanych przez **Sklep** wniosków testowych. Jeśli integracja została wykonana prawidłowo a dane wniosków w **Systemie** są poprawne to zmieniający jest tryb pracy **Sklepu** na „**Produkcyjny**”.

12. Klienci **Sklepu** mogą rozpocząć procesowanie wniosków kredytowych na sprzedawane w sklepie towary.

1.3 Sposoby implementacji połączenia z certyfikatem

Dla różnych platform programistycznych trzeba będzie użyć różnych sposobów implementacji wywołania WebService'u z użyciem certyfikatu dostarczonego przez Opiekuna Sklepu. Poniżej zostały opisane trzy rozwiązania dla najczęściej spotykanych platform programistycznych używanych w sklepach internetowych.

1.3.1. PHP.

1. Na komputerze powinien zostać zainstalowany certyfikat formacie .pfx.
2. Po poprawnym zainstalowaniu certyfikatu oraz wpisaniu odpowiedniego hasła (otrzymanego również od Opiekuna Sklepu) będzie możliwe otworzenie w przeglądarce strony z WSDL-em WebService'u pod następującym adresem:

```
https://api.eraty.pl/eRatyService/eRatyService.asmx?WSDL
```

3. Za pomocą **openssl** należy przerobić certyfikat w formacie .pfx na certyfikat w formacie .pem. Tu też konieczne będzie użycie hasła certyfikatu. Komenda zmieniająca certyfikat.pfx w certyfikat.pem wygląda następująco:

```
pkcs12 -in ścieżka_do_certyfikatu/certyfikat.pfx -out certyfikat.pem -nodes -clcerts
```

5. W kodzie aplikacji wywołującej WebService należy zaimplementować poniższą sekwencję kodu:

```
$cert = 'ścieżka_do_certyfikatu/certyfikat.pem';  
$wsdl = 'https://api.eraty.pl/eRatyService/eRatyService.asmx?WSDL';  
$this->Soap = new SoapClient($wsdl, array("local_cert" => $cert));
```

1.3.2. C#.

1. Na komputerze powinien zostać zainstalowany certyfikat formacie .pfx przysłany przez Opiekuna Sklepu.
2. Po poprawnym zainstalowaniu i wpisaniu odpowiedniego hasła (otrzymanego od Opiekuna Sklepu) będzie możliwe otworzenie w przeglądarce strony z WSDLeM serwisu:

```
https://api.eraty.pl/eRatyService/eRatyService.asmx?WSDL
```

3. Za pomocą **openssl** należy przerobić certyfikat w formacie .pfx na certyfikat w formacie .pem. Tu też konieczne będzie użycie hasła certyfikatu. Komenda zmieniająca certyfikat.pfx w certyfikat.pem wygląda następująco:

```
pkcs12 -in ścieżka_do_certyfikatu/certyfikat.pfx -out certyfikat.pem -nodes -clcerts
```

4. Za pomocą **openssl** należy przerobić certyfikat w formacie .pem na certyfikat w formacie .cer. Do zmiany certyfikat.pem w certyfikat.cer powinno się użyć niniejszej komendy:

```
x509 -inform PEM -in ścieżka_do_certyfikatu/certyfikat.pem -outform DER
```

certyfikat.cer

5. Następnie należy dodać w systemie dwie referencje: dla serwisu i dla biblioteki.

a. Web Reference dla serwisu:

- użycie prawego przycisku na projekcie i należy wybrać „Add Service Reference”;
- w nowo otwartym oknie powinno się wybrać opcję „Advanced”, a następnie „Add Web Reference”;
- w kolejnym oknie powinien zostać wpisany adres serwisu w pole URL a następnie należy użyć „Go”;
- powinno się nadać nazwę WebService’u, np. eRatyService w pole Web reference name i kliknąć w „Add Reference”.

b. Reference dla biblioteki System.Security.dll:

- użycie prawego przycisku na projekcie i należy wybrać „Add Reference”;

- w nowo otwartym oknie powinno się kliknąć w zakładkę „.NET” i wyszukać opcję System.Security;
 - należy zaznaczyć „System.security” i kliknąć OK.
6. Na początek w kodzie klasy, która jest odpowiedzialna za wywołanie jest konieczne dodanie odwołania do serwisu i biblioteki.

```
using System.Security.Cryptography.X509Certificates;
using {nazwa_projektu}.eRatyService;
```

7. W samej aplikacji należy zastosować następujący kod wywołania:

```
string certPath = {ścieżka do certyfikatu/certyfikat.cer};
eRatyService.eRatyService service = new eRatyService.eRatyService();
service.ClientCertificates.Add(X509Certificate.CreateFromSignedFile(certPath));
var result = service.sprawdzPESEL(id, pesel);
```

1.3.3. JAVA.

W środowisku JAVA można zastosować wiele rozwiązań w zależności od używanego framework'a. Nie ma jednak z reguły potrzeby zmiany sposobu wywołania metod WebService'u w kodzie aplikacji. Konieczny za to jest import certyfikatu do tak zwanego „TrustStore”, przy pomocy narzędzia JVM „keytool” oraz dodanie informacji dla aplikacji aby wiedziała gdzie znajduje się certyfikat.

1. Na komputerze powinien zostać zainstalowany certyfikat formacie .pfx przysłany przez Opiekuna Sklepu.
2. Po poprawnym zainstalowaniu i wpisaniu odpowiedniego hasła (otrzymanego od Opiekuna Sklepu) będzie możliwe otworzenie w przeglądarce strony z WSDLem serwisu:

<https://api.eraty.pl/eRatyService/eRatyService.asmx?WSDL>

3. Certyfikat w formacie .pfx należy zaimportować do formatu .jks (java keystore – jest to format dla JVM). Konieczne jest do tego hasło certyfikatu, to samo które zostało wpisane wcześniej.

- a. Poprzez konsolę należy przejść do katalogu, w którym jest zainstalowane JDK i należy wpisać:

```
keytool -importkeystore -srckeystore ścieżka_do_certyfikatu/certyfikat.pfx -
srcstoretype pkcs12 -destkeystore certyfikat.jks -deststoretype JKS
```

- b. Po wpisaniu komendy uzyska się prośbę o dwukrotne wpisanie nowego hasła, które musi mieć minimum 5 znaków i które należy samodzielnie podać.

- c. Następnie wyświetlona zostanie prośba o podanie hasła certyfikatu.pfx i po poprawnym jego wpisaniu certyfikat zostanie dodany do magazynu certyfikatów JVM.

4. W kodzie aplikacji należy dodać następującą linijkę:

```
System.setProperty("javax.net.ssl.trustStore", ścieżka_do_pliku_cacerts);
```

5. W przypadku używania aplikacji klienckiej, z której wywoływany będzie WebService należy pamiętać aby dodać właściwość do serwerowego JVM. Przykładowo wpis w setenv.bat z serwera Tomcat będzie wyglądać następująco:

```
set "JAVA_OPTS=... -
Djavax.net.ssl.trustStore=%CATALINA_HOME%/trustStore/certyfikat.jks"
```

1.4 Błędy operacji

W wypadku niepowodzenia wywołania operacji **System** sygnalizuje błąd zwracając komunikat błędu (wyjątek SOAP). Zawiera on dodatkowe dane błędu, w postaci prostego dokumentu XML, którego węzeł faultString zawiera dwa podwęzły:

KOD – zawierający kod błędu,

OPIS – zawierający tekstowy opis błędu.

Przykład błędu:

```
<soap:Body>
<soap:Fault>
<faultCode>
    soap:Server
</faultCode>
<faultString>
    &lt;KOD&gt;13&lt;/KOD&gt;&lt;OPIS&gt;&lt;/OPIS&gt;
</faultString>
<detail/>
</soap:Fault>
</soap:Body>
```

W poniższej tabeli zaprezentowano komunikaty ogólne, które mogą się pokazać przy użyciu każdej metody. Każda metoda będzie miała uzupełniona o komunikaty specyficzne dla niej.

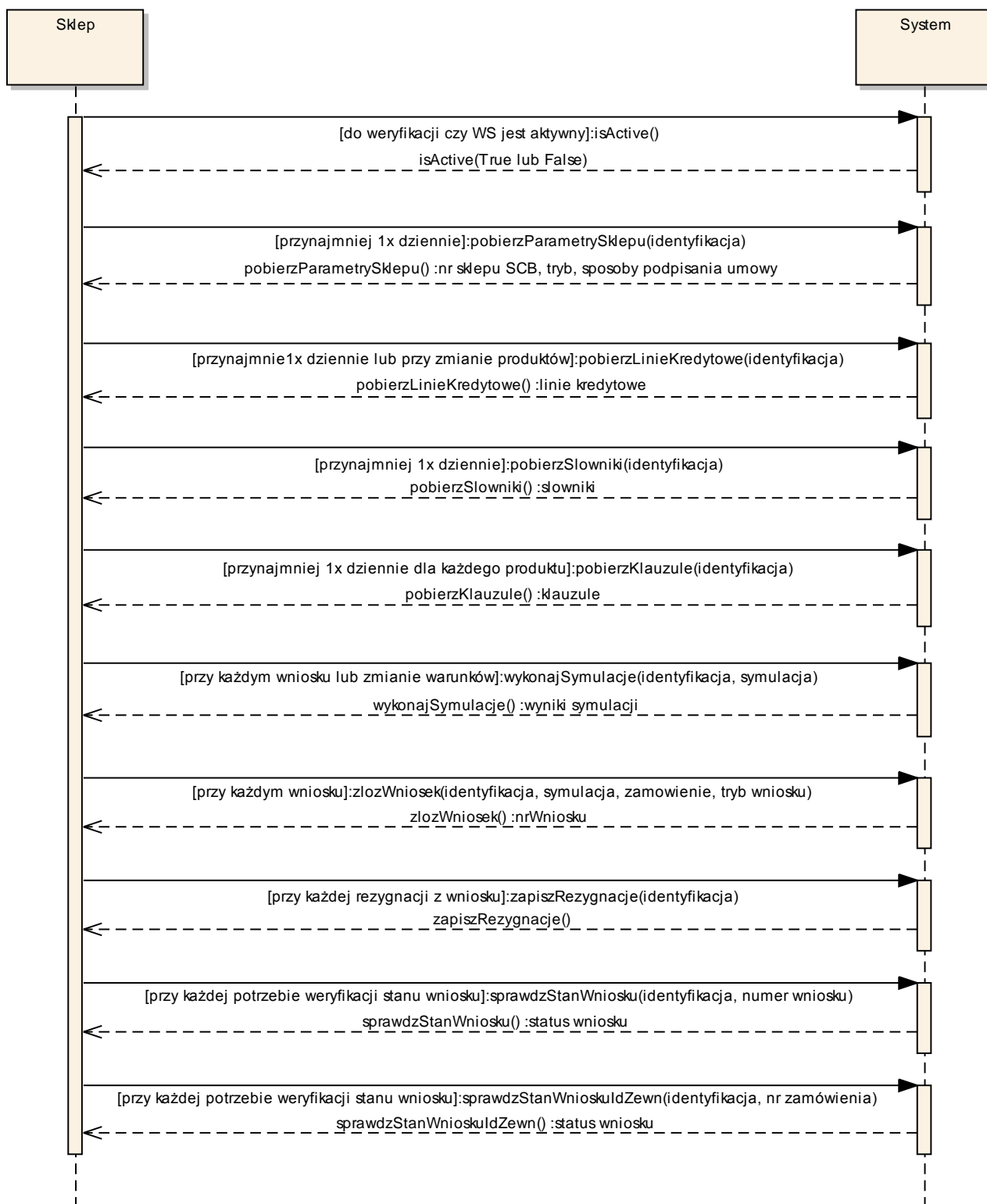
kod błędu	Opis
1	Błędny parametr
2	Błąd logowania
3	Brak uprawnień do sklepu
4	Błąd wewnętrzny systemu eRaty
5	Niepoprawny typ produktu
6	Produkt niedostępny na sklepie
8	Brak sklepu w systemie
9	Brak uprawnień do metody
199	Błąd ogólny walidacji: + nazwa pola
603	Przerwa techniczna

2 Algorytm użycia metod

Do komunikacji z **Systemem** został utworzony zestaw metod, które zostały dokładniej opisane w rozdziale 3. Metody można podzielić na trzy rodzaje.

1. Metody konfiguracyjne - to metody służące do skonfigurowania aplikacji do sprzedaży kredytów po stronie **Sklepu**.
2. Metody produkcyjne - to metody służące do zapisu danych po stronie **Systemu**.
3. Metody informacyjne - to metody informacyjne wykonujące wyliczenia lub weryfikację danych po stronie **Systemu**.

Kolejność wywołania metod w procesie sprzedaży kredytów została opisana na poniższym diagramie sekwencji.



3 Metody

We wszystkich metodach jest używany obiekt **identyfikacja**. Służy on do autentykacji w metodach webserwisów. Poniżej został podany dokładny opis identyfikacji.

identyfikacja	standardowy obiekt identyfikacja typ: identyfikacja, typ złożony, zdefiniowany w ramach serwisu webowego		
	własności		
	login	login użytkownika technicznego sklepu typ: string	
	haslo	hasło użytkownika technicznego typ: string	
	numerSklepu	numer sklepu w systemie banku typ: long	
	typProduktu	typ produktu(kanał sprzedaży) – wartość słownikowa typ: int	
		kod	opis
		8	eHP Kurier
		24	eHP PL (przelew)
		25	eHP POS (w sklepie)
wariantSklepu	wariant sklepu Klienta – parametr przekazywany dla zgodności z poprzednimi wersjami systemu (należy przekazywać wartość 1) typ: int		
zrodloWniosku	opis słowny skąd podchodzi wniosek, dla integracji bezpośredniej można wysłać „integracja_XXXXX”, gdzie XXXXX to numer sklepu w SCB. typ: string		

Uwaga. „typProduktu” obecnie oznaczony kodem „8” jest odpowiednikiem kodu „0”, który występował w starym systemie.

3.1 Metody konfiguracyjne

3.1.1 isActive

Metoda służy do weryfikacji czy **WebService** działa w danym momencie i czy jest dostępny.

Sygnatura metody.

isActive();

Częstotliwość wywoływania metody.

Metoda powinna być wywołana przynajmniej raz dziennie lub po każdej zmianie konfiguracji sklepu w **Systemie** SCB.

Metoda może być także wywołana przed złożeniem wniosku aby sprawdzić czy nie ma obecnie przerwy technicznej (szczególnie w godzinach nocnych).

Parametry wejściowe.

Brak

Zwracana wartość.

opis
typ: boolean, zawierający informację czy web service będzie przyjmował wnioski (true – można wysłać wniosek, false – przerwa techniczna)

3.1.2 pobierzParametrySklepu

Metoda służy do pobrania podstawowych parametrów sklepu na podstawie przysłanego obiektu **identyfikacji**. Zwrotne wartości metody ustawiają główne parametry pracy **Sklepu** przy tworzeniu wniosków ratalnych.

Sygnatura metody.

parametrySklepu pobierzParametrySklepu(identyfikacja identyfikacja);

Częstotliwość wywoływania metody.

Metoda powinna być wywołana przynajmniej raz dziennie lub po każdej zmianie konfiguracji sklepu w Systemie SCB.

Parametry wejściowe.

Parametry	opis
identyfikacja	standardowy obiekt identyfikacja typ: identyfikacja, typ złożony, zdefiniowany w ramach serwisu webowego

Zwracana wartość.

parametry	opis
typ: parametrySklepu, typ złożony, zdefiniowany w ramach serwisu webowego, zawierający informacje o parametrach sklepu	
czyPrzyciskPrzelicz	globalny parametr czy wyświetlać przycisk „ Przelicz ” w aplikacji eWniosek typ: boolean
czyZmigrowany	znacznik czy sklep jest zmigrowany ze starego systemu (0 – niezmigrowany, 1 – zmigrowany) typ: boolean
potwierdzaTowary	znacznik czy sklep potwierdza towary (0 – nie potwierdza, 1 – potwierdza) typ: boolean
szablon	identyfikator szablonu wyglądu strony aplikacji eWniosek (przekazywana wartość 1) typ: string
ziarnoWplaty	globalny parametr wielkości ziarna symulacji na suwaku w aplikacji eWniosek typ: int
systemZrodlowy	typ systemu źródłowego sklepu (obecnie system SCB przekazywana wartość 0, stary system przekazywana wartość 3) typ: int
trybSklepu	tryb działania sklepu, gdzie 1 to tryb testowy a 2 to tryb produkcyjny, numer ten powinien się przenieść do metody zlozWniosek do pola trybWniosku typ: int
numerSklepu	wynikowy numer sklepu a więc numer sklepu w SCB (sklepy ze starego systemu miały inną numerację, która jest mapowana na numerację SCB) typ: string
sposobPodpisaniaUmowy	tablica obiektów zawierająca aktywne sposoby podpisania umowy (wynikające z kanałów przypisanych do Sklepu) typ: sposobPodpisaniaUmowy, typ złożony, zdefiniowany w ramach serwisu webowego

Własności	
typProduktu	numer kanału ze słownika typów produktów typ: int
Opis	nazwa kanału typ: string
Klauzula	klauzula do wyświetlenia na pierwszej stronie przed wyborem kanału sprzedaży typ: string
czyUbezpieczenie	Określa czy dla kanału dostępne jest ubezpieczenie typ: boolean

Błędy.

kod błędu	opis
10	Błąd parametryzacji sklepu na poziomie kanału sprzedaży

3.1.3 pobierzLinieKredytowe

Metoda służy do pobrania podstawowych parametrów produktów ratalnych dla sklepu. Jako dane podajemy parametry pozwalające na identyfikację Klienta, w odpowiedzi dostajemy zestaw obiektów zawierających dane produktów ratalnych.

Sygnatura metody.

linieKredytowe pobierzLinieKredytowe (identyfikacja identyfikacja, String[] kodyLiniiKredytowych);

Częstotliwość wywoływania metody.

Metoda powinna być wywołana przynajmniej raz dziennie lub po każdej zmianie konfiguracji produktów Sklepu w Systemie SCB.

Parametry wejściowe.

parametry	opis
identyfikacja	standardowy obiekt identyfikacja typ: identyfikacja, typ złożony, zdefiniowany w ramach serwisu webowego
kodyLiniiKredytowych	tablica kodów produktów ratalnych, Klient opcjonalnie może podać kody produktów ratalnych aby pobrać ich podstawowe dane. W przypadku podania pustej tablicy metoda zwraca informacje o wszystkich produktach ratalnych dostępnych na sklepie. typ: String[]

Zwracana wartość.

parametry	opis	
podstawowe parametry produktów ratalnych typ: linieKredytowe, typ złożony, zdefiniowany w ramach serwisu webowego		
nazwaSklepu	nazwa sklepu internetowego typ: string	
liniaKredytowa	lista obiektów opisujących parametry produktu ratalnego typ: contractTypeLocal, typ złożony, zdefiniowany w ramach serwisu webowego	
	własności	opis
	kodLiniiKredytowej	numer produktów ratalnych (aby zidentyfikować który produkt ratalny klient wybrał) typ: string
	opis	nazwa opisowa produktu ratalnego typ: string
minimalnaLiczbaRat	minimalna liczba rat dla tego produktu ratalnego	

		typ: int
	maksymalnaLiczbaRat	maksymalna liczba rat dla tego produktu ratalnego typ: int
	dostepneLiczbyRat	możliwe ilości rat w ramach produktu ratalnego - zapis zerojedynkowy (przykładowo dla dostępnych rat z przedziału od 3 do 6 będzie to 001111) typ: string
	numerSklepu	numer sklepu w systemie SCB typ: long
	ubezpieczenie	link do strony na serwerach banku na której zamieszczone jest OWU typ: string
	idUbezpieczenia	identyfikator przypisanego do produktu ubezpieczenia (0 – brak ubezpieczenia) typ: int
	opisUbezpieczenia	opis ubezpieczenia jako stała obecnie wartość „Ubezpieczenie na życie – MetLife TUnŻiR S.A.” typ: string

Błędy.

kod błędu	Opis
21	Brak aktywnych produktów ratalnych
22	Błędne identyfikatory produktów ratalnych

3.1.4 pobierzSłowniki

Metoda służy do pobrania wartości słownikowych. Jako dane wejściowe podane są parametry pozwalające na identyfikację **Sklepu**, a w odpowiedzi zwracany zostaje zestaw wartości słownikowych, które umożliwią zebranie wszystkich danych wymaganych do złożenia wniosku kredytowego.

Uwaga. Sklepy zintegrowane na bazie poprzedniej instrukcji integracji (2.x) otrzymają nowe wartości słownikowe pochodzące od SCB i te powinny być zwracane we wniosku. Należy zwrócić uwagę na korzystanie z nowych kodów słownikowych. Wszelkie kodowanie wartości na sztywno w sklepach jest zabronione.

Sygnatura metody.

słowniki pobierzSłowniki(identyfikacja identyfikacja);

Częstotliwość wywoływania metody.

Metoda powinna być wywołana przynajmniej raz dziennie aby pobrać aktualne wartości wszystkich słowników.

Parametry wejściowe.

parametry	opis
identyfikacja	standardowy obiekt identyfikacja typ: identyfikacja, typ złożony, zdefiniowany w ramach serwisu webowego

Zwracana wartość.

parametry	opis
zestaw wartości słownikowych	typ: słowniki, typ złożony, zdefiniowany w ramach serwisu webowego
godziny	godziny kontaktu z klientem a wartość (Value) zwracana jest w metodzie zlozWniossek w polach godzOd i godzDo typ: TreeMap<integer, string>
	key wartość słownikowa typ: integer
	value opis wartości

		typ: string
gtu	branża towaru, gdzie key zwracana w metodzie zlozWniosek w polu idGtu przy każdym towarze typ: TreeMap<integer, string>	
	Key	wartość słownikowa typ: integer
	Value	opis wartości typ: String
powodRezygnacji	powód rezygnacji klienta z procesowania wniosku kredytowego, który jest zwracany w metodzie zapiszRezygnacje (key) typ: TreeMap<integer, string>	
	Key	wartość słownikowa typ: integer
	Value	opis wartości typ: string
sposobPodpisaniaUmowy	sposób dostarczenia umowy to lista kanałów sprzedaży zwracana w metodzie zlozWniosek w polu sposobPodpisUmowy typ: TreeMap<integer, string>	
	Key	wartość słownikowa typ: integer
	Value	opis wartości typ: string
stanCywilny	stan cywilny Klienta zwracany w metodzie zlozWniosek w polu stanCywilny (key) typ: TreeMap<integer, string>	
	key	wartość słownikowa typ: integer
	value	opis wartości typ: string
statusMieszkania	status zamieszkania klienta – w chwili obecnej parametr nie jest używany w informacji zwrotnej ze sklepu typ: TreeMap<integer, string>	
	key	wartość słownikowa typ: integer
	value	opis wartości typ: string
typDrugiegoDokumentu	typ drugiego dokumentu tożsamości który zwracany jest (key) w metodzie zlozWniosek w polu serDokumentuTozs2 typ: TreeMap<integer, string>	
	key	wartość słownikowa typ: integer
	value	opis wartości typ: string
województwo	województwo – w chwili obecnej parametr nie jest używany w informacji zwrotnej ze sklepu typ: TreeMap<integer, string>	
	key	wartość słownikowa typ: integer
	value	opis wartości typ: string
wykształcenie	wykształcenie klienta – w chwili obecnej parametr nie jest używany w informacji zwrotnej ze sklepu typ: TreeMap<integer, string>	

	key	wartość słownikowa typ: integer	
	value	opis wartości typ: string	
zrodloDochodu	typ źródła dochodu Klienta zwracany (key) w metodzie zlozWniosek w polu zrodloDochodu typ: TreeMap<Integer, ZrodloDochodu>		
	key	wartość słownikowa typ: integer	
	value	obiekt opisujący dane źródło dochodu typ: zrodloDochodu, typ złożony, zdefiniowany w ramach serwisu webowego	
		własności	
		opis	opis wartości typ: string
parametry	przyjmuje wartości ciągu 12 znaków z zakresu 0-3 określających wymagalność pól dotyczących dochodu a dokładniejszy opis znajduje się w opisie słownika zrodloDochodu (4.10) typ: string(20)		

3.1.5 pobierzKlauzule

Metoda służy do pobrania klauzul prawnych. Jako dane wejściowe są podawane parametry pozwalające na identyfikację Sklepu oraz kanału, a w odpowiedzi zwracany jest zestaw klauzul prawnych dla kanału sprzedaży (typu produktu) wraz z podstawowymi informacjami na ich temat.

Sygnatura metody.

klauzule pobierzKlauzule (identyfikacja identyfikacja);

Częstotliwość wywoływania metody.

Metoda powinna być wywołana przynajmniej raz dziennie dla każdego udostępnionego

Sklepowi kanału sprzedaży (typu produktu). Każdy kanał może posiadać inne treści klauzul wymaganych przez SCB w procesie tworzenia wniosku kredytowego.

Parametry wejściowe.

parametry	opis
identyfikacja	standardowy obiekt identyfikacja typ: identyfikacja, typ złożony, zdefiniowany w ramach serwisu webowego

Zwracana wartość.

parametry	opis	
	obiekt zawierający listę klauzul prawnych typ: ArrayOfClausulaLocal, typ złożony, zdefiniowany w ramach serwisu webowego, zawierający listę obiektów klauzula	
klauzule	obiekt zawierający dane klauzuli typ: clausulaLocal, typ złożony, zdefiniowany w ramach serwisu webowego	
	item	dane klauzuli
	idKlauzuli	identyfikator klauzuli typ: int
	nazwa	nazwa klauzuli pozwalająca na jej identyfikację typ: string
	tresc	treść klauzuli do wyświetlenia klientowi zanim wyrazi zgodę

		typ: string
	wymaganaOdpowiedz	czy dana klauzula wymaga odpowiedzi, gdzie 0 to nie wymaga a 1 wymaga typ: int
	wymaganaZgoda	czy zgoda na daną klauzulę jest wymagana, gdzie 0 to nie wymaga a 1 wymaga zgody typ: int

3.1.6 pobierzKredytPunkty

Metoda służy do pobrania punktów odbioru towaru dla sklepu ze zdefiniowanym kanałem eHP POS. Na wejściu podawane są dane identyfikacyjne sklepu. W odpowiedzi zwracana jest lista punktów odbioru towarów dla tego kontrahenta z pełnym zestawem parametrów. **Sygnatura metody.**

klauzule pobierzKredytPunkty (identyfikacja identyfikacja);

Częstotliwość wywoływania metody.

Metoda powinna być wywołana przynajmniej raz dziennie ale tylko dla tych **Sklepów**, które posiadają w swojej ofercie kanał sprzedaży eHP POS.

Parametry wejściowe.

Parametry	Opis
identyfikacja	standardowy obiekt identyfikacja typ: identyfikacja, typ złożony, zdefiniowany w ramach serwisu webowego

Zwracana wartość.

	obiekt zawierający listę punktów odbioru typ: ArrayOfCreditPointLocal, typ złożony, zdefiniowany w ramach serwisu webowego, zawierający listę obiektów punktów odbioru	
kredytPunkty	obiekt zawierający dane kredyt punktu typ: creditPointLocal, typ złożony, zdefiniowany w ramach serwisu webowego	
	item	dane kredyt punktu
	adres	obiekt zawierający adres kredyt punktu typ: adres, typ złożony, zdefiniowany w ramach serwisu webowego
	Własności	
	kodPocz	kod pocztowy typ: string(5)
	miasto	miasto typ: string(50)
	ulica	ulica typ: string(50)
	nrDomu	nr domu typ: string(7)
	nrMieszkania	nr mieszkania typ: string(7)
	godzOtwarcia	godziny pracy sklepu – parametr nie jest obecnie używany typ: string

	id	Identyfikator sklepu – parametr nie jest obecnie używany typ: long
	kanal	typ produktu (kanał sprzedaży) – wartość słownikowa typ: string
	nrTelefonu	numer telefonu do sklepu typ: string
	numerSklepu	numer sklepu w systemie SCB, który jest punktem odbioru (lub kredyt punktem) typ: string
	region	identyfikator rejonu w którym znajduje się sklep typ: int
	wspolrzedneGeo	współrzędne GPS kredyt punktu typ: string

3.2 Metody produkcyjne

3.2.1 zlozWniosek

Metoda służy do zapisu wniosku o przyznanie kredytu. Jako dane podawane są dane identyfikacyjne **Sklepu**, dane symulacji oraz parametry opisujące ten wniosek. W odpowiedzi wysyłany jest numer wniosku, pod którymi została zapisana wiadomość. Umożliwi on odpytanie o status przetwarzania wniosku w SCB podczas procesu rozpatrzenia wniosku kredytowego w SCB.

Uwaga: użycie metody w trybie produkcyjnym wniosku jest jednoznaczne ze złożeniem wniosku kredytowego w SCB.

Sygnatura metody.

String zlozWniosek (identyfikacja identyfikacja, symulacja symulacja, zamowienie zamowienie);

Częstotliwość wywoływania metody.

Metoda powinna być wywołana przy każdym wniosku kredytowym po zebraniu wszystkich danych klienta wymaganych do składania wniosku.

Parametry wejściowe.

parametry	Opis		
identyfikacja	standardowy obiekt identyfikacja typ: Identyfikacja, typ złożony, zdefiniowany w ramach serwisu webowego, wymagane		
symulacja	obiekt zawierający dane potrzebne do przeprowadzenia symulacji typ: symulacja, typ złożony, zdefiniowany w ramach serwisu webowego, wymagane		
zamowienie	obiekt zawierający dane zamówienia z systemu klienckiego typ: zamowienie, typ złożony, zdefiniowany w ramach serwisu webowego, wymagane		
	klient	dane klienta typ: klient, typ złożony, zdefiniowany w ramach serwisu webowego, wymagane	
		Własności	
		pesel	PESEL typ: string(11), wymagane
		imie	imię typ: string(35), wymagane
	nazwisko	nazwisko	

			typ: string(35), wymagane									
	adres	<p>obiekt zawierający adres zamieszkania Klienta</p> <p>typ: adres, typ złożony, zdefiniowany w ramach serwisu webowego, wymagane</p> <p style="text-align: center;">własności</p> <table border="1"> <tr> <td>kodPocz</td> <td>kod pocztowy typ: string(5)</td> </tr> <tr> <td>miasto</td> <td>Miasto typ: string(50), wymagane</td> </tr> <tr> <td>ulica</td> <td>ulica typ: string(50), wymagane</td> </tr> <tr> <td>nrMieszkania</td> <td>nr domu typ: string(7), wymagane</td> </tr> <tr> <td>nrDomu</td> <td>nr domu typ: string(7)</td> </tr> </table>	kodPocz	kod pocztowy typ: string(5)	miasto	Miasto typ: string(50), wymagane	ulica	ulica typ: string(50), wymagane	nrMieszkania	nr domu typ: string(7), wymagane	nrDomu	nr domu typ: string(7)
kodPocz	kod pocztowy typ: string(5)											
miasto	Miasto typ: string(50), wymagane											
ulica	ulica typ: string(50), wymagane											
nrMieszkania	nr domu typ: string(7), wymagane											
nrDomu	nr domu typ: string(7)											
	adresKorespondencyjny	<p>obiekt zawierający adres korespondencyjny Klienta</p> <p>typ: adres, typ złożony, zdefiniowany w ramach serwisu webowego, wymagane</p>										
	email	<p>e-mail Klienta poprzez który będzie można się komunikować z Klientem i będzie on mógł procesować wniosek za pomocą aplikacji Panel Klient</p> <p>typ: string(35), wymagane</p>										
	telefonDomowy	<p>telefon domowy Klienta (9 cyfr)</p> <p>typ: string(9)</p>										
	telefonSluzbowy	<p>telefon służbowy (9 cyfr)</p> <p>typ: string(9)</p>										
	komorkaPrywatna	<p>telefon komórkowy prywatny Klienta (9 cyfr)</p> <p>typ: string(9)</p>										
	komorkaSluzbowa	<p>telefon komórkowy służbowy Klienta (9 cyfr)</p> <p>typ: string(9)</p>										
	typDokumentuTozs	<p>typ dokumentu tożsamości – wartość słownikowa ale w SCB jest możliwa wysyłka tylko wartości 9 a więc dowodu osobistego</p> <p>typ: int, wymagane</p>										
	serDokumentuTozs	<p>seria dokumentu tożsamości</p> <p>typ: string(3), wymagane</p>										
	nrDokumentuTozs	<p>numer dokumentu tożsamości</p> <p>typ: string(20), wymagane</p>										
	typDokumentuTozs2	<p>typ drugiego dokumentu tożsamości(musi być różny od typDokumentuTozs) – wartość słownikowa</p> <p>typ: int, nie wymagane</p>										
	serDokumentuTozs2	<p>seria drugiego dokumentu tożsamości ze słownika typDrugiegoDokumentu</p> <p>typ: string(3), nie jest wymagane</p>										

		nrDokumentuTozs2	numer drugiego dokumentu tożsamości typ: string(20), nie jest wymagane	
		statusMieszkaniowy	status mieszkaniowy – wartość słownikowa typ: int, wymagane	
		stanCywilny	stan cywilny – wartość słownikowa ze słownika stanCywilny typ: int, wymagane	
		sposobPodpisUmowy	sposób podpisania umowy będący słownym opisem kanału podpisania umowy, a więc wartości zwracane to 8, 24 lub 25, zgodny ze słownikiem sposobPodpisaniaUmowy . Pole dla zgodności ze starym systemem. typ: string, wymagane	
		liczbaCzlonkowGosp Domowego	liczba osób na utrzymaniu kredytobiorcy typ: int, wymagane	
		godzOd	preferowana godzina kontaktu do – wartość słownikowa zgodna ze słownikiem godziny typ: string(2)	
		godzDo	preferowany sposób podpisania umowy – wartość słownikowa zgodna ze słownikiem godziny typ: string	
		województwo	województwo – wartość słownikowa typ: int, nie jest wymagane	
		wykształcenie	wykształcenie klienta – wartość słownikowa typ: int, nie jest wymagane	
		identyfikatorKredytP unktu	identyfikator kredyt punktu (punktu odbioru) w którym ma być podpisana umowa kredytowa, który jest wymagany przy składaniu wniosku w kanale eHP POS typ: int	
		dochod	pełne dane dotyczące źródła dochodu klienta typ: dochod, typ złożony, zdefiniowany w ramach serwisu webowego, wymagane	
			własności	
			zrodloDochodu	typ źródła dochodów – wartość słownikowa typ: double(10, 2)
			zrodloDochoduDataOd	data, od której obowiązuje dane źródło dochodów, format RRRR-MM-DD typ: string

			zrodloDochoduDataDo	data, do której obowiązuje dane źródło dochodów, format RRRR-MM-DD typ: string
			nazwaFirmy	nazwa firmy typ: string(60)
			Nip	NIP typ: long(10)
			regon	regon typ: long(9)
			stanowisko	stanowisko pracy klienta typ: string(50)
			adres	Obiekt zawierający adres miejsca pracy typ: adres, typ złożony, zdefiniowany w ramach serwisu webowego
			numerSwiadczenia	Numer świadczenia emerytalnego/rentowego typ: string(30)
			dochodNetto	dochód netto, wartość liczbowa z separatorem w postaci kropki typ: string(10)
			zawod	Zawód wykonywany – wartość słownikowa obecnie nie jest używana typ: int
	towar		element listy zawierający dane n-tego towaru, typu towary typ: towary, typ złożony, zdefiniowany w ramach serwisu webowego, wymagane	
			Własności	
			idTowaru	identyfikator ze sklepu n-tego towaru typ: string(20), wymagane
			nazwaTowaru	nazwa n-tego towaru typ: string(50), wymagane
			wartoscTowaru	cena n-tego towaru, z

				separator w postaci kropki typ: double(11,2), wymagane
			liczbaSztukTowaru	ilość jednostek n-tego towaru typ: double(7,2), wymagane
			jednostkaTowaru	jednostka miary n-tego towaru na wniosku, proponowane wartości to „szt.”, „kg”, „m”, „m2”, „m3”, „litr” lub „inne” typ: string(10), wymagane
			idGtu	branża towaru – wartość słownikowa ze słownika gtu typ: int
	odpNaKlauzule			Lista odpowiedzi na klauzule prawne Utrzymywana dla zgodności ze starym systemem. Parametr ma formę ciągu znaków 0-1, gdzie 1 –zgodna, 0– brak zgody. Przykładowo, gdy były dwie klauzule z pytaniami to ma on dwa znaki długości, np. „10”, gdzie odpowiedź na pierwsze pytanie jest zgodą a na drugie jej brakiem. typ: string, wymagane
	sposobDostarczeniaTowaru			sposób dostarczenia towaru typ: string (128), wymagane
	nrZamowieniaSklepu			nr zamówienia wygenerowanego przez sklep typ: string(64), wymagane
	klauzula	lista klauzul występujących na wniosku typ: klauzulaOdpowiedz, typ złożony		
Własności				
		idKlauzuli	identyfikatory klauzul pobieranych metodą pobierzKlauzule (nowe pole dla eWniosku) typ: int, wymagane jak występuje klauzula	
		stanZgody	stan zgody 0-brak, 1-zgodna, 2-niezostała zaznaczona typ: int, wymagane jak występuje klauzula	
	trybWniosku	tryb wniosku zgodny z polem trybSklepu 1– testowy, 2– produkcyjny typ: int		

Zwracana wartość.

parametry	Opis
nrWniosku	nr wniosku typ: string

Błędy.

kod błędu	Opis
21	Brak aktywnych produktów ratałnych
22	Błędne identyfikatory produktów ratałnych
33	Błędna wartość koszyka
35	Błąd symulacji, przekroczony limit kredytu
36	Błąd symulacji, za niska kwota kredytu
52	Niewłaściwa liczba rat dla produktu
53	Zbyt wysoka kwota dochodu
103	Błędna data zatrudnienia – od
105	Błędna data zatrudnienia – do
106	Brak zgody na przetwarzanie danych osobowych
107	Brak potwierdzenia zapoznania się z warunkami ubezpieczenia
112	Brak punktu odbioru (PS wymieniony jako kredyt punkt nie jest przypisany do punktów odbioru kontrahenta)

3.2.2 zapiszRezygnacje

Metoda służy do zapisu powodów rezygnacji z wniosku kredytowego wskazanych przez Klienta, który w trakcie składania wniosku rezygnuje z wnioskowania o kredyt (przed wysłaniem wniosku do SCB).

Sygnatura metody.

`zapiszRezygnacje(identyfikacja identyfikacja, int powodId, string powod, string klientIP);`

Parametry wejściowe.

Parametry	Opis
identyfikacja	standardowy obiekt identyfikacja typ: identyfikacja, typ złożony, zdefiniowany w ramach serwisu webowego
powodId	Numer powodu rezygnacji (identyfikator ze słownika powodRezygnacji) typ: int
powod	Słowny powód rezygnacji (opis powodu rezygnacji) typ: string
klientIP	adres IP klienta typ: string

Zwracana wartość.

Brak.

Błędy.

kod błędu	opis
200	Nieprawidłowy identyfikator powodu rezygnacji X

3.3 Metody informacyjne

3.3.1 wykonajSymulacje

Metoda służy do przeprowadzenia symulacji kredytu. Jako dane wejściowe podane powinny zostać parametry pozwalające na przeprowadzenie symulacji. Natomiast w odpowiedzi wysyłany jest obiekt zawierający wynik symulacji.

Sygnatura metody.

`wynikSymulacji wykonajSymulacje(identyfikacja identyfikacja, symulacja symulacja);`

Częstotliwość wywoływania metody.

Symulacja powinna być wykonana przy tworzeniu każdego nowego wniosku kredytowego oraz przy każdej zmianie warunków kredytowych dla wniosku. Zmianami takimi są korekta wartości koszyka zakupowego, ilości rat, wpłaty własnej, produktu ratalnego czy też zgody na ubezpieczenie.

Parametry wejściowe.

parametry	opis	
identyfikacja	standardowy obiekt identyfikacja typ: identyfikacja, typ złożony, zdefiniowany w ramach serwisu webowego	
symulacja	obiekt zawierający dane potrzebne do przeprowadzenia symulacji typ: symulacja, typ złożony, zdefiniowany w ramach serwisu webowego	
	własności	opis
	kodLiniiKredytowej	produkt ratalny na którym tworzony jest wniosek typ: string, wymagane
	liczbaRat	liczba rat typ: int, wymagane
	wplataWlasna	wpłata własna typ: double
	wartoscTowarow	wartość zamówionych towarów, razem z kosztem przesyłki, który także może być kredytowany, a wartość towarów powinna być z przedziału od 100 do 50 000 złotych typ: double, wymagane
	karencja	liczba dni karencji typ: int
	zgodaNaUbezpieczenie	cecha oznaczająca czy klient wyraził zgodę na ubezpieczenie typ: boolean
	symulator	znacznik określający czy dane przeliczenie idzie z symulatora czy jest wykonywane na przeliczanym wniosku, będzie przyjmował wartości 1 dla wywołań z symulatora (metoda wykonajSymulacje) i 0 dla wywołań z wniosku (metoda zlozWniosek) typ: int

Zwracana wartość.

Parametry	Opis
obiekt zawierający wynik symulacji typ: wynikSymulacji, typ złożony, zdefiniowany w ramach serwisu webowego	
kwotaKredytu	kwota kredytu typ: double
wysokoscMiesiecznejRaty	wysokość miesięcznej raty typ: double
wysokoscRocznejStopyProc	wysokość rocznej stopy procentowej typ: double
APR	APR - rzeczywista roczna stopa oprocentowania typ: double
kosztKredytu	całkowity koszt kredytu typ: double

wplataWlasna	wpłata własna typ: double
liczbaRat	liczba rat typ: int
ubezpieczenie	link do konkretnego dokumentu opisującego OWU ubezpieczenia, a jeśli kredyt jest bez ubezpieczenia parametr pozostaje pusty typ: string
klauzula	treść klauzuli prawnej dotyczącej wyników symulacji wymaganej do wyświetlenia klientowi typ: string

Uwaga. Obsługiwany we wcześniejszych integracjach parametr SposobPodpisUmowy nie będzie przesyłany w nowej integracji. Wartości sposobów podpisania będą obsługiwane w metodzie **pobierzSlovniki**. Należy dokonać stosownych zmian.

Błędy.

kod błędu	opis
21	Brak aktywnych produktów ratałnych
22	Błędne identyfikatory produktów ratałnych
33	Błędna wartość koszyka
35	Błąd symulacji, przekroczony limit kredytu
36	Błąd symulacji, za niska kwota kredytu
52	Niewłaściwa liczba rat dla produktu
54	Za niska wpłata gotówkowa

3.3.2 sprawdzStanWniosku

Polecenie służy do sprawdzenia stanu przetwarzania wniosku. Na wejściu podawane są dane identyfikacyjne **Sklepu** oraz numer wniosku. W odpowiedzi wysyłane są dane opisujące status wniosku oraz status kredytu.

Sygnatura metody.

string sprawdzStanWniosku(identyfikacja identyfikacja, string nrWniosku);

Częstotliwość wywoływania metody.

Wywołanie metody powinno być wykonywane w każdym momencie kiedy operator chce zweryfikować jaki jest stan wniosku kredytowego. Może to być proces automatyczny ustawiony raz lub kilka razy dziennie na wszystkich aktywnych wnioskach albo jednorazowe wywołanie na żądanie użytkownika na jednym wniosku.

Parametry wejściowe.

parametry	Opis
Identyfikacja	standardowy obiekt identyfikacja typ: identyfikacja, typ złożony, zdefiniowany w ramach serwisu webowego
nrWniosku	numer wniosku w systemie SCB typ: string(64)

Zwracana wartość.

parametry	Opis
	obiekt zawierający informacje na temat procesowania się wniosku typ: infoWniosekTo, typ złożony, zdefiniowany w ramach serwisu webowego
nrUmowy	numer umowy kredytowej w systemie SCB typ: string

nrZamowieniaSklep	numer zamówienia przychodzący ze sklepu typ: string (64)
wplataWlasna	wysokość wpłaty własnej typ: double
nrWniosku	numer wniosku w systemie SCB typ: string
statusWniosku	stan przetwarzania wniosku typ: string
statusKredytu	status kredytu informujący czy został udzielony czy nie typ: string
wartoscKoszyka	wartość wszystkich towarów procesowanych w SCB na wniosku, pole służy do weryfikacji czy wartość zamówienia ze sklepu zgadza się z wartością towarów procesowaną w SCB typ: double
nrUmowyElrat	numer umowy w systemie ElRat – pole obecnie nie jest używane typ: long
numerZewn	numer systemu zewnętrznego – pole obecnie nie jest używane typ: string
infoZezwolen	informacja o warunkach otrzymania kredytu – pole obecnie nie jest używane typ: string
numerZezwolen	numer systemu eRaty – obecnie używane do przechowywania numeru wniosku typ: string
errkod	numer błędu – pole obecnie nie jest używane typ: int
message	opis błędu – pole obecnie nie jest używane typ: string

Błędy.

kod błędu	opis
101	Błędny numer wniosku
102	brak praw do wniosku

3.3.3 sprawdzStanWnioskuIdZewn

Kolejna metoda, także służy do pobierania informacji o wniosku. Jako dane wejściowe podawane są parametry pozwalające na identyfikację Klienta oraz numer zamówienia w sklepie internetowym.

W odpowiedzi wysyłany jest obiekt zawierający bieżący status wniosku oraz status kredytu.

Sygnatura metody.

infoWniosekTO sprawdzStanWnioskuIdZewn(identyfikacjaidentyfikacja, string nrZamowieniaZewn);

Częstotliwość wywoływania metody.

Wywołanie metody powinno być wykonywane w każdym momencie kiedy operator albo system Sklepu chce zweryfikować jaki jest stan wniosku kredytowego. Może to być proces automatyczny ustawiony raz lub kilka razy dziennie na wszystkich aktywnych wnioskach albo jednorazowe wywołanie na żądanie użytkownika na jednym wniosku.

Parametry wejściowe.

parametry	opis
------------------	-------------

identyfikacja	standardowy obiekt identyfikacja typ: identyfikacja, typ złożony, zdefiniowany w ramach serwisu webowego
nrWniosku	numer zamówienia w sklepie typ: string

Zwracana wartość.

Parametry	Opis
	obiekt zawierający informację o danym wniosku typ: infoWniosekTO, typ złożony, zdefiniowany w ramach serwisu webowego, zawierający informacje o wniosku
nrUmowy	numer umowy kredytowej w systemie SCB typ: string
nrZamowieniaSklep	numer zamówienia przychodzący ze sklepu typ: string (64)
wplataWlasna	wysokość wpłaty własnej typ: double
nrWniosku	numer wniosku w systemie SCB typ: string
statusWniosku	bieżący status wniosku typ: string
statusKredytu	status kredytu informujący czy został udzielony czy nie typ: string
wartoscKoszyka	wartość wszystkich towarów procesowanych w SCB na wniosku, pole służy do weryfikacji czy wartość zamówienia ze sklepu zgadza się z wartością towarów procesowaną w SCB typ: double
numerZewn	numer zamówienia przychodzący ze sklepu typ: string
nrUmowyElrat	numer wniosku z systemu SCB – pole obecnie nie jest używane typ: bigInteger
numerZezwolen	identyfikator wniosku ze starego systemu Zezwoleń – obecnie używane do przechowywania numeru wniosku typ: string
errkod	numer błędu – pole obecnie nie jest używane typ: int
message	opis błędu – pole obecnie nie jest używane typ: string

Błędy.

kod błędu	Opis
51	Problem spójności danych, znaleziono kilka wniosków z takim samym numerem zamówienia
101	Błędny numer wniosku
102	brak praw do wniosku

Uwaga. Usunięto błędy 108 i 500, przy czym dotychczasowy kod 500 będzie występował, jako 603.

3.3.4 getInfoWn

Metoda służy do pobierania informacji o wniosku. Jako dane wejściowe wysyłane są parametry pozwalające na identyfikację **Sklepu** oraz numer wniosku.

W odpowiedzi przekazywany obiekt zawierający dane dotyczące wniosku kredytowego.

Sygnatura metody.

wniosekDTO getInfoWn(identyfikacja identyfikacja, long nrInternetowyWn);

Parametry wejściowe.

parametry	opis
identyfikacja	standardowy obiekt identyfikacja typ: identyfikacja, typ złożony, zdefiniowany w ramach serwisu webowego
nrWniosku	numer wniosku w systemie SCB typ: long

Zwracana wartość.

parametry	opis	
obiekt zawierający informację o danym wniosku typ: wniosekDTO, typ złożony, zdefiniowany w ramach serwisu webowego, zawierający informacje o wniosku		
wniosekDTO	obiekt transportowy zawierający informacje o wniosku typ: wniosekDTO, typ złożony, zdefiniowany w ramach serwisu webowego	
nrSklepu	numer sklepu w systemie SCB typ: string	
wariantSklepu	wariant sklepu – przekazywana zawsze wartość 1 typ: string	
adresZamieszkania	obiekt transportowy zawierający adres zamieszkania klienta typ: adresDTO, typ złożony, zdefiniowany w ramach serwisu webowego	
	własności	
	nazwisko	Nazwisko Klienta typ: string
	imie	Imię Klienta typ: string
	ulica	ulica typ: string
	nrDomu	nr domu typ: string
	kodPocztowy	kod pocztowy typ: string
	miasto	miasto typ: string
	adresKorespondencyjny	obiekt transportowy zawierający adres korespondencyjny klienta typ: adresDTO
własności		
ulica		ulica typ: string
nrDomu		nr domu typ: string
kodPocztowy		kod pocztowy typ: string
Email	email klienta typ: string	
dataWprowadzenia	data wprowadzenia typ: date	
statusWniosku	identyfikator statusu wniosku typ: short	

internetowyNrWniosku	internetowy numer wniosku typ: string	
nrUmowy	numer umowy kredytowej typ: string	
wplataWlasna	kwota wpłaty własnej typ: bigDecimal	
wartoscKoszyka	wartość wszystkich towarów procesowanych w SCB na wniosku, pole służy do weryfikacji czy wartość zamówienia ze sklepu zgadza się z wartością towarów procesowaną w SCB typ: bigDecimal	
dataKontakt	ostatnia data kontaktu typ: date	
liczbaTelefonow	liczba telefonów typ: long	
tel_domowy	numer telefonu domowego typ: string	
tel_sluzbowy	numer telefonu służbowego typ: string	
tel_komorkowy_prywatny	numer prywatnego telefonu komórkowego typ: string	
tel_komorkowy_sluzbowy	numer służbowego telefonu komórkowego typ: string	
towary	lista zamówionych towarów dla danego wniosku typ: List<towaryDTO>, typ złożony, zdefiniowany w ramach serwisu webowego	
	własności	
	idTowaruWSklepie	identyfikator jednoznacznie identyfikujący towar w sklepie typ: string
	id	klucz główny typ: long
	nazwa	nazwa towaru typ: string
	czydostepny	czy towar jest dostępny typ: boolean
	czyusuniety	czy towar jest usunięty typ: boolean
	szczegoly	szczegóły zamówienia, np. ilość towaru typ: string
	dataPotwierdzenia	Ostatnia data zmiany statusu dostępności towaru typ: date
liczbaRat	liczba rat typ: short	

Błędy.

kod błędu	opis
101	Błędny numer wniosku
102	Brak praw do wniosku

Uwaga. Usunięto błąd 500, który będzie występował, jako 603.

Metoda może zwrócić błąd ogólny nr 9, który wynika z uprawnień przypisanych do Sklepu.

4 Słowniki

Metoda **pobierzSłowniki** zwraca listę słowników z parametrami opisanymi poniżej (lista aktualna na dzień 2014-12-22).

4.1 godziny

Godziny kontaktu są przekazywane na poziom sklepu w słowniku **godziny** w postaci identyfikatora oraz jego części opisowej. Są tu przekazywane godziny preferowanego przez klienta kontaktu w sprawie zawarcia umowy kredytowej.

Wartości opisowe wybrane przez klienta przychodzą w polach **godzOd** oraz **godzDo** w metodzie **zlozWniosek**.

Poniżej zostały wylistowane wartości zwracane w słowniku.

Identyfikator	Wartość wyświetlana
8	08:00
9	09:00
10	10:00
11	11:00
12	12:00
13	13:00
14	14:00
15	15:00
16	16:00
17	17:00
18	18:00
19	19:00
20	20:00
21	21:00

4.2 gtu

Branże sprzedawanych towarów są przekazywane do sklepu w słowniku **gtu** w postaci identyfikatora oraz opisu branży. Powinny one być przypisane do poszczególnych towarów sprzedawanych w sklepie.

Identyfikatory branż wybranych przez klienta przychodzą wraz z poszczególnymi towarami w polu **idGtu** w węźle **towary** w metodzie **zlozWniosek**.

Poniżej zostały wylistowane wartości zwracane w słowniku.

Identyfikator	Wartość opisowa branży
1	Sprzęt RTV,SAT
2	AGD(sprz.zmech)
3	Meble
4	Sprzęt komp
5	Art.sport.i.tur
6	Odzież
7	Mat./Chem.bud.
8	Narz.i maszyny
9	Masz.i.urz.biur
10	Sprz.myśl.i.węd
11	Instr.muzyczne
12	Sprzęt fotogr.
13	Sprzęt medyczny

14	Art.motor.
15	Art.prod.rolnej
16	Art.wyp.mieszk.
17	Tel.komórkowe
18	Opał
20	Motorowery
21	Kursy pr. jazdy
22	Kasy fiskalne
23	Ubezp.kom.P.T.F
24	Stolarka otw.
25	Telewizja cyfr.
27	Usł.Internetowe
28	Usł.Edukacyjne
29	Usł.Turystyczne
30	Usługi Medyczne
90	Usł.inne
99	Inne towary

4.3 powódRezygnacji

Powody rezygnacji z zawarcia umowy ratalnej są przekazywane w słowniku **powódRezygnacji**. Powody rezygnacji są zwracane do systemu SCB po rezygnacji klienta z zakupu na raty, kiedy wybiera on z listy wyboru jeden z wymienionych powodów. Poniżej zostały wylistowane wartości zwracane w słowniku.

Identyfikator	Powód rezygnacji
1	zdecydowałem się w inny sposób sfinansować zakup
2	zrezygnowałem z zakupu
3	oferta kredytu jest zbyt droga
4	formularz jest zbyt długi i niejasny
5	nie posiadam w tym momencie danych potrzebnych do wypełnienia formularza
6	inne powody, jakie?

4.4 sposóbPodpisaniaUmowy

Wszystkie dostępne w Systemie SCB kanały sprzedaży (typy produktów) są przekazywane w słowniku **sposóbPodpisaniaUmowy** w postaci numeru kanału sprzedaży oraz słownego opisu nazwy sposobu podpisania umowy przez Klienta. Poniżej zostały wylistowane wartości zwracane w słowniku.

Identyfikator	Opis sposobu podpisania umowy
8	Kurier
24	Przelew
25	W punkcie odbioru towaru

4.5 stanCywilny

Możliwe opcje stanu cywilnego Klienta są przekazywane w słowniku **stanCywilny** w postaci identyfikatora oraz opisu słownego stanu cywilnego.

Identyfikator stanu wybranego przez klienta w **Sklepie** przychodzi w polu **stanCywilny** w metodzie **zlozWniosek**.

Poniżej zostały wylistowane wartości zwracane w słowniku.

Identyfikator	Opis stanu cywilnego
1	Panna/Kawaler
2	Zamężna/Żonaty
3	Inne

4.6 statusMieszkania – nie używane

Statusy mieszkania są przekazywane w słowniku **statusyMieszkania** w postaci identyfikatora oraz opisu formy własności miejsca zamieszkania Klienta. Cecha obecnie nie jest wykorzystywana w procesie SCB i utrzymywana dla zgodności z poprzednimi wersjami **WebService'ów**.

Poniżej zostały wylistowane wartości zwracane w słowniku.

Identyfikator	Opis statusu zamieszkania
1	mieszkanie własne
2	mieszkanie spółdzielcze
3	mieszkanie komunalne
4	mieszkanie wynajęte
5	mieszkanie inne
6	własny dom
7	wynajęty dom
8	u rodziny

4.7 typDrugiegoDokumentu

Możliwe do wykorzystania przez Klienta typy drugiego dokumentu poświadczającego jego tożsamość są przekazywane w słowniku **typDrugiegoDokumentu** w postaci identyfikatora oraz opisu słownego typu dokumentu.

Identyfikator typu drugiego dokumentu wybranego przez klienta w **Sklepie** przychodzi w polu **typDokumentuTozs2** w metodzie **zlozWniosek**.

Poniżej zostały wylistowane wartości zwracane w słowniku.

Identyfikator	Opis typu drugiego dokumentu
1	Paszport
2	Prawo jazdy
3	Książeczka wojskowa
4	Legitymacja ubezpieczeniowa
16	Legitymacja emeryta lub rencisty

4.8 wojewodztwo – nie używane

Województwa są przekazywane w słowniku **wojewodztwo** w postaci identyfikatora oraz nazwy województwa. Cecha obecnie nie jest wykorzystywana w procesie SCB i utrzymywana dla zgodności z poprzednimi wersjami **WebService'ów**.

Poniżej zostały wylistowane wartości zwracane w słowniku.

Identyfikator	Wartość opisowa
1	Dolnośląskie
2	kujawsko-pomorskie
3	Lubuskie
4	Łódzkie
5	Lubelskie
6	Mazowieckie
7	Małopolskie
8	Opolskie
9	Podlaskie
10	Podkarpackie
11	Pomorskie
12	Śląskie
13	Świętokrzyskie
14	warmińsko-mazurskie
15	Wielkopolskie
16	Zachodniopomorskie

4.9 wykształcenie – nie używane

Wykształcenie klientów jest przekazywane w słowniku **wykształcenie** w postaci identyfikatora oraz opisu słownego wykształcenia. Cecha obecnie nie jest wykorzystywana w procesie SCB i utrzymywana dla zgodności z poprzednimi wersjami **WebService'ów**. Poniżej zostały wylistowane wartości zwracane w słowniku.

Identyfikator	Wartość opisowa
1	podstawowe
2	zawodowe
3	średnie
4	licencjat
5	wyższe

4.10 źródłoDochodu

Dostępne w systemie typy źródła dochodu są przekazywane w słowniku **źródłoDochodu** w postaci identyfikatora, opisu słownego nazwy źródła dochodu oraz ciągu 12 cyfr z zakresu od 0 do 3 określających wymagalności poszczególnych parametrów dotyczących opisu źródła dochodu dla określonego jego typu. Wymagalność pola może przyjmować następujące wartości od 0 do 3.

- 0 - parametr niewymagany.
- 1 - parametr potrzebny do częściowego zapisu wniosku.
- 2 - parametr potrzebny do kompletnego zapisania wniosku.
- 3 - parametr dodatkowy, opcjonalny.

Kolejne pozycje w ciągu dwunastu cyfr określają następujące parametry dochodu (numer pozycji oraz nazwa).

- 1 - data, od której obowiązuje dane źródło dochodu (pole: `dochod.zrodloDochoduDataOd`).
- 2 - data, do której obowiązuje dane źródło dochodu (pole: `dochod.zrodloDochoduDataDo`).
- 3 - nazwa firmy (pole: `dochod.nazwaFirmy`).
- 4 - adres firmy (pole: `dochod.adres`).
- 5 - stanowisko Klienta (pole: `dochod.stanowisko`).
- 6 - zawód wykonywany przez Klienta (pole: `dochod.zawod`).
- 7 - numer świadczenia (pole: `dochod.numerSwiadczenia`).
- 8 - numer NIP firmy (pole: `dochod.nip`).
- 9 - nr regon firmy (pole: `dochod.regon`).
- 10 - gmina (pole: `dochod.gmina`).
- 11 - wysokość dochodu netto (pole: `dochod.dochodNetto`).
- 12 - typ źródła dochodu (pole: `dochod.zrodloDochodu`).

Typ źródła dochodu wybranego przez Klienta w **Sklepie** przychodzi w polu **źródłoDochodu** w metodzie **zlozWniosek**.

Poniżej zostały wylistowane wartości zwracane w słowniku dla poszczególnych wartości źródeł dochodu.

Identyfikator	Opis źródła dochodu	zrodloDochoduDataOd	zrodloDochoduDataDo	nazwaFirmy	adres	stanowisko	zawod	numerSwiadczenia	nip	regon	gmina	dochodNetto	zrodloDochodu
12	Zatrudniony na umowę o pracę	2	3	2	2	2	0	0	3	3	0	1	1
2	Emerytura	0	0	0	0	0	0	1	0	0	0	1	1
1	Renta	0	3	0	0	0	0	1	0	0	0	1	1
4	Prowadzenie gospodarstwa rolnego	0	0	0	0	0	0	0	0	0	2	1	1
3	Własna działalność gospodarcza	1	0	2	2	0	0	0	2	3	0	1	1
9	Pracownicy sfery budżetowej	1	3	2	2	2	0	0	3	3	0	1	1
11	Inne zatrudnienie	1	3	0	2	3	0	0	3	0	0	1	1

4.11 statusWniosku i statusKredytu

Dostępne w systemie SCB statusy kredytów zależą od tego w jakim kanale umowa jest sprzedawana (eHP Kurier, eHP PL i eHP POS). Statusy kredytów można podzielić na cztery typy.

1. **Procesowany**: status tymczasowy. Wniosek jest w trakcie procesowania w Banku.
2. **Do potwierdzenia**: status tymczasowy. Wskazuje na oczekiwanie Banku na reakcję sklepu, który powinien potwierdzić dostępność towarów.
3. **Wydaj_towar**: status ostateczny. W wyniku decyzji kredytowej, Bank wydał zgodę na zawarcie umowy o kredyt i udzielił kredytu Kredytobiorcy. Bank zakończył procesowanie wniosku. Należność została przekazana do rozliczenia.
4. **Odmowa**: status ostateczny. Wniosek w trakcie procesu został anulowany.

Poniżej zostało przedstawione przełożenie statusów wniosków na statusy kredytu na poszczególnych kanałach. Może się jednak zdarzyć, że zostanie zwrócony jakiś inny, wewnętrzny dla banku status wniosku. W takim przypadku należy uznać wniosek jako będący w trakcie procesowania.

4.11.1 eHP Kurier

Status	Nazwa statusu wniosku	Status kredytu
1	Wniosek przyjęty	Procesowany
3	Przyjęty do realizacji	Procesowany
6	Wniosek rozpatrywany	Procesowany
7	Wstępna zgoda	Procesowany
8	Rozpatrzony negatywnie	Odmowa
9	Anulowany	Odmowa
10	Dokumenty wysłane do klienta	Procesowany
11	Dokumenty w weryfikacji	Procesowany
14	Kredyt uruchomiony	Wydaj_towar
17	Rezygnacja klienta	Odmowa

19	Zarchiwizowany	Wydaj_towar
21	Towary do zatwierdzenia	Do potwierdzenia
22	Wstępna zgoda indywidualna	Procesowany
24	Brak towaru	Odmowa
27	Rozpatrzony negatywnie indywidualnie	Odmowa
32	Odłożony podczas weryfikacji	Procesowany
33	Wstępna zgoda indywidualna z wymogiem zaświadczenia	Procesowany
38	Rozpatrzony negatywnie po ponownej weryfikacji	Odmowa
44	Czeka na uruchomienie	Wydaj_towar
46	Wstępna zgoda z zaświadczeniem	Procesowany
52	Anulowanie na rzecz nowej umowy	Procesowany

4.11.2 eHP PL

Status	Nazwa statusu wniosku	Status kredytu
1	Wniosek przyjęty	Procesowany
3	Przyjęty do realizacji	Procesowany
6	Wniosek rozpatrywany	Procesowany
7	Wstępna zgoda	Procesowany
8	Rozpatrzony negatywnie	Odmowa
9	Anulowany	Odmowa
10	Dokumenty wysłane do klienta	Procesowany
11	Dokumenty w weryfikacji	Procesowany
14	Kredyt uruchomiony	Wydaj_towar
17	Rezygnacja klienta	Odmowa
19	Zarchiwizowany	Wydaj_towar
21	Towary do zatwierdzenia	Do potwierdzenia
22	Wstępna zgoda indywidualna	Procesowany
24	Brak towaru	Odmowa
26	Przypomnienie ponowne o braku wpłaty	Procesowany
27	Rozpatrzony negatywnie indywidualnie	Odmowa
32	Odłożony podczas weryfikacji	Procesowany
33	Wstępna zgoda indywidualna z wymogiem zaświadczenia	Procesowany
38	Rozpatrzony negatywnie po ponownej weryfikacji	Odmowa
39	Potwierdzone zawarcie umowy	Wydaj_towar
46	Wstępna zgoda z zaświadczeniem	Procesowany

4.12.3 eHP POS

Status	Nazwa statusu wniosku	Status kredytu
1	Wniosek przyjęty	Procesowany
3	Przyjęty do realizacji	Procesowany
6	Wniosek rozpatrywany	Procesowany
7	Wstępna zgoda	Procesowany
8	Rozpatrzony negatywnie	Odmowa
9	Anulowany	Odmowa
21	Towary do zatwierdzenia	Do potwierdzenia
22	Wstępna zgoda indywidualna	Procesowany
24	Brak towaru	Odmowa
25	Dokumenty potwierdzone	Procesowany
27	Rozpatrzony negatywnie indywidualnie	Odmowa
32	Odłożony podczas weryfikacji	Procesowany
33	Wstępna zgoda indywidualna z wymogiem zaświadczenia	Procesowany
39	Potwierdzone zawarcie umowy	Wydaj_towar
41	Przekazany do rozliczenia w CRU	Wydaj_towar
46	Wstępna zgoda z zaświadczeniem	Procesowany
47	Przekazany do POS	Procesowany

52	Anulowanie na rzecz nowej umowy	Procesowany
53	Przyjęty do realizacji SKL	Procesowany

5 Inne parametry słownikowe

5.1 idKlazuLi

Klazure zwracane obecnie na poziomie produktów zdalnych.

IdklazuLi	Nazwa
6	Kontakt
7	Zapoznanie OWU

5.2 potwierdzaTowary

Identyfikator	Wartość opisowa
0	Sklep nie potwierdza towarów – wnioski wpadają bezpośrednio do obsługi w SCB
1	Sklep potwierdza towary – wnioski muszą zostać potwierdzone w Panelu Sklep przed ich obsługą w SCB

5.2 systemZrodlowy

Identyfikator	Wartość opisowa
0	System SCB
3	Stary system ex-Żagiel

5.3 trybSklepu, trybWniosku

Określa tryb pracy Sklepu w systemie SCB.

Identyfikator	Wartość opisowa
1	testowy
2	produkcyjny

5.4 typProduktu

Określa kanał sprzedaży inaczej zwany typem produktu ratalnego obsługiwanym w SCB.

Identyfikator	Wartość opisowa
8	eHP Kurier
24	eHP PL (przelew)
25	eHP POS (w sklepie)