

[CPS] Api dla aplikacji webowej w promocji "naklejkowej" dla JMP

Historia modyfikacji:

Wersja	Data modyfikacji	Opis modyfikacji	Autor
1.0	2016-06-29	Draft	Romuald Bąk + Michał Włodarczyk
1.1	2016-07-04	Aktualizacji API, dodanie obsługi kont klienta.	Romuald Bąk
1.2	2016-07-05	Wersjonowanie danych klienta. Umożliwienie częściowej modyfikacji danych klienta	Tomasz Kruk
1.3	2016-07-20	Zmiana body dla usługi "Pobranie danych klienta". Wprowadzenie portfeli i statystyk. Zmiana nazwy aplikacji na Customer Portal Server.	Romuald Bąk
1.4	2016-07-21	Poprawka literówki w nazwie portfela: Strickers2016->Stickers2016	Romuald Bąk
1.5	2016-07-28	Rozszerzenie interfejsu danych klienta o pola adresowe. Zmiana interfejsu logowania: dodanie pola 'Hasło', usunięcie pola 'CardPin'. Dodanie metody do nadania hasła.	Tomasz Kruk
1.6	2016-07-29	Poprawa formatu numeru telefonu (bez +) Zwracanie aktualnego statusu klienta w odpowiedzi usługi modyfikującej dane klienta Zmiana sposobu wyznaczania znacznika danych klienta - ETag	Romuald Bąk + Tomasz Kruk
1.7	2016-08-03	Zwracanie informacji o wersji portfeli	Filip Frątczak

1. Uczestnicy procesu

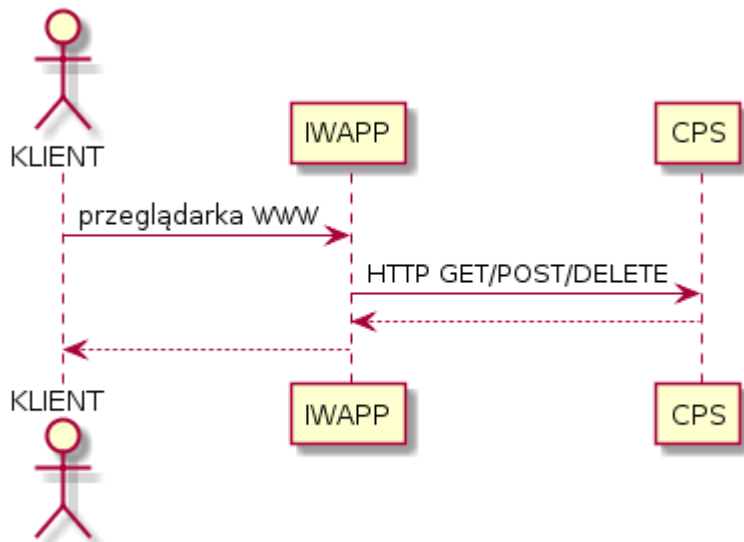
Isobar WebApp (IWAPP) - aplikacja webowa z której korzystają klienci Biedronki uczestniczący w promocji "naklejkowej"

Customer Portal Server (CPS) - serwer aplikacji udostępniający usługi do wyświetlania i zarządzania danymi osobowymi klientów, wyświetlania stanu punktów zgromadzonych na karcie i zarządzania relacjami pomiędzy kartami.

Klient - Klient sklepu Biedronka

2. Komunikacja między uczestnikami procesu

Typowy schemat realizacji żądania



Komunikacja jest realizowana za pomocą usług HTTP REST. Dane użyte w żądaniach i odpowiedziach są serializowane z użyciem JSON.

Komunikacja wymaga zestawienia bezpiecznego połączenia między serwerami zabezpieczonego protokołem TLS 1.2.

Odpowiednie żądania i odpowiedzi powinny deklarować nagłówki: `Accept` i `Content-Type` z wartością: `application/json`.

Statusy błędów są zwracane za pomocą standardowych kodów HTTP Status Codes. W sytuacjach w których niezbędne jest dostarczenie dodatkowych informacji o przyczynie błędów, usługi zwracają rezultat w postaci:

```
{
  "HttpCode": "403",
  "MessageCode": "NoAuthothorization",
  "Message": "Błędny numer karty lub pin"
}
```

Pole `MessageCode` zawiera identyfikator, który może być wykorzystany przez konsumenta usług do wyświetlenia odpowiedniego komunikatu dla klienta. W przypadku nieobsłużonych błędów pole `"MessageCode"` może zawierać wartość `"UnknownError"` i wówczas pole `"Message"` zawiera komunikat błędu zwracany przez serwer. W pozostałych sytuacjach pole `Message` zawiera komunikat, który może być ignorowany, jeśli aplikacja rozpoznała `MessageCode`.

3. Udostępnione usługi

3.1. Logowanie

Request

- POST `https://{host}/loyalty/v1/login`
- Body

```
{
  "CardNumber": "9012312312312",
  "Password": "3309",
}
```

Przy pierwszym logowaniu, jako hasło może być podany PIN karty.

Operacją wymaganą po pierwszym logowaniu jest zmiana hasła. Pozostałe metody zostaną zablokowane. Ich odblokowanie nastąpi po poprawnym nadaniu hasła.

Po zmianie hasła logowanie za pomocą PINu karty nie będzie możliwe.

Response

- Status = 200, Body:

```
{
  "SessionId": "652e6756162c4dbdb4b63dc6b29976d4",
}
```

```
}
```

Logowanie przebiegło poprawnie. Rezultat zawiera id sesji, które powinno być przekazywane we wszystkich kolejnych żądaniach w nagłówku HTTP "Authorization"

- Status = 403, Body:

```
{
  "HttpCode": "403",
  "MessageCode": "InvalidCredentials",
  "Message": "Błędny numer karty lub pin"
}
```

Błąd logowania.

Możliwe wartości "MessageCode":

InvalidCredentials

AccountTemporarilyLockedFor5Minutes

AccountLocked

3.2. Wylogowanie

Request

- POST https://{host}/loyalty/v1/logout
- Body - brak

Response

- Status = 204
- Status = 403, Body:

```
{
  "HttpCode": "403",
  "MessageCode": "NoAuthothorization",
  "Message": "Brak uprawnień"
}
```

Błąd logowania.

Możliwe wartości "MessageCode":

NoAuthorization

SessionExpired

3.3. Pobranie danych klienta

Request

- GET https://{host}/loyalty/v1/customers/current
Żądanie pobiera dane aktualnie zalogowanego klienta.
- Nagówek HTTP:
 - **If-None-Match** (domylnie brak) - ETag odesany we wcześniejszej odpowiedzi usugi. Jeeli ETag po stronie usugi jest inny lub nie zostanie przesany to zwracane s aktualne dane klienta, w przeciwnym razie zwracany jest kod bdu 304 (Not modified).

Response

- Status = 200,
Nagówek HTTP: **ETag** - znacznik danych klienta. Znacznik zmienia si wraz z modyfikacj danych klienta.
Body:

```
{
  "FirstName": "Adam",
  "LastName": "Bialecki",
  "PhoneNumber": "48602123789",
  "Email": "adam234.bialecki223@bialecki.com",
  "Street": "Gajowa",
}
```

```

"HouseNumber": "6",
"ApartmentNumber": "8",
"PostalCode": "00-950",
"City": "Poznań",
"Status": "Active",
"PersonalDataProcessingConsent": true,
"EmailCommunicationConsent": true,
"SmsCommunicationConsent": false,
"PreferredLanguage": "pl-PL",
"PreferredStore": "3456",
"LoyaltyCard": {
  "CardNumber": "9949837485921",
  "Wallets": [
    {
      "Id": "Stickers2016",
      "Balance": 10.0,
      "Stat": {
        "Lost": 0.22,
        "TransferredOut": 132.00,
        "TransferredIn": 9.12,
        "Issued": 199.22
      }
    }
  ],
  "Version": 2
},
{
  "Id": "SomeOtherWallet",
  "Balance": 0.0,
  "Stat": {
    "Lost": 0.22,
    "TransferredOut": 132.00,
    "TransferredIn": 9.12,
    "Issued": 199.22
  }
},
"Version": 1
}
],
"MasterCardReference": {
  "CardNumber": "9949837485333",
  "OwnerFirstName": "Maria",
  "OwnerLastName": "Błaszczuk"
},
"SlaveCardReferences": [
  {
    "CardNumber": "9949837485333",
    "OwnerFirstName": "Maria",
    "OwnerLastName": "Błaszczuk",
    "Wallets": [
      {
        "Id": "Stickers016",
        "Balance": 10.0,
        "Stat": {
          "TransferredOut": 132.00
        }
      }
    ]
  }
],
{
  "CardNumber": "9949837485332",
  "OwnerFirstName": "Jan",
  "OwnerLastName": "Pachlowski",
  "Wallets": [
    {
      "Id": "Stickers2016",
      "Balance": 10.0,
      "Stat": {

```

```

        "TransferredOut":132.00
      }
    }
  ]
}
],
"CustomerProperties":[
  {
    "Name":"Property1",
    "Value":"Value1",
    "ReadOnly":false,
    "ValidationRegex":"^a.*z$",
    "Required":true
  },
  {
    "Name":"Zgoda na otrzymanie nagrody w postaci bonu wysłanego pocztą elektroniczną",
    "Value":false,
    "ReadOnly":true
  }
],
"Messages":[
  {
    "Id":"d939ae873ada4d0db805953b96299016",
    "Message":"Twój numer telefonu nie mógł być poprawnie zweryfikowany. Sprawdź czy podałeś
poprawny numer telefonu lub skontaktuj się z BOK."
  },
  {
    "Id":"362ac541b22b4eeba0e7105fbd5fe9d",
    "Message":"Zgromadziłeś wystarczającą liczbę punktów aby uzyskać nagrodę. Zapraszamy po odbiór
do dowolnego sklepu Biedronka."
  }
]
}

```

- "CustomerProperties" zawiera właściwości "dynamiczne" klienta, które użytkownik systemu Perspectiv może dodać z poziomu aplikacji. Ten mechanizm będzie używany do zarządzania dodatkowymi zgodami klienta na specyficzne procesy związane z obsługą programu lojalnościowego.

Możliwe wartości "Status":

- Active - aktywny,
- WaitingForActivationSmsConfirmation - czeka na aktywację,
- NotActivated - nieaktywowany,
- Disabled - zablokowany.

Możliwe wartości "Accounts.Name":

- AccumulatedStickers - obecna liczba naklejek do odebrania,
 - LostStickers - liczba naklejek jakie utraci klient (nie chciał ich odebrać w POS),
 - IssuedStickers - liczba naklejek wydanych klientowi
 - TransferredStickers - liczba punktów przeniesionych z karty *Slave* do karty *Master*.
- Status = 304, Dane klienta nie zostały zmodyfikowane,

Body:

```

{
  "HttpCode": "304",
  "MessageCode": "CustomerNotModified",
  "Message": "Dane klienta nie zostały zmodyfikowane"
}

```

- Status = 403, Patrz opis błędu w punkcie: 3.2

3.4. Modyfikacja danych klienta

Request

- Nagówek HTTP:
 - **If-Match** - powinien zawierać **ETag** odesany podczas pobrania danych klienta, patrz punkt 3.3. Jeżeli **ETag** po stronie usługi jest taki sam to dane klienta zostaną zmodyfikowane, w przeciwnym razie zwracany jest kod bdu 412 (Precondition failed).
ETag określa kolejne wersje danych klienta i jest numerowany od 0.
- POST <https://{host}/loyalty/v1/customers/current>
Kompletna modyfikacja danych klienta,
Body:

```
{
  "FirstName": "Adam",
  "LastName": "Bialecki",
  "PhoneNumber": "48602123789",
  "Email": "adam234.bialecki223@bialecki.com",
  "Street": "Gajowa",
  "HouseNumber": "6",
  "ApartmentNumber": "8",
  "PostalCode": "00-950",
  "City": "Poznań",
  "PersonalDataProcessingConsent": true,
  "EmailCommunicationConsent": true,
  "SmsCommunicationConsent": false,
  "PreferredLanguage": "pl-PL",
  "PreferredStore": "4532",
  "CustomerProperties": {
    "Property1": "Value1",
    "Property2": true
  }
}
```

- PATCH <https://{host}/loyalty/v1/customers/current>
Czciowa modyfikacja danych klienta,
Body:

```
{
  "Email": "adam.bialecki@bialecki.com",
  "PreferredStore": ""
}
```

Response

- Status = 200
Zmiana pól w kliencie (imi oraz numeru telefonu) może mieć wpływ na zmianę statusu, dlatego w odpowiedzi zwracany jest aktualny status klienta.
Body:

```
{
  "Status": "Active"
}
```

Możliwe wartości "Status":

- Active - aktywny,
 - WaitingForActivationSmsConfirmation - czeka na aktywację,
 - NotActivated - nieaktywowany,
 - Disabled - zablokowany.
- Status = 403, Patrz opis bdu w punkcie: 3.2
 - Status = 400, bdné dane - w danym istnieją pola, które nie zostały wyspecyfikowane w dokumentacji

```
{
  "HttpCode": "400",
}
```

```
"MessageCode": "NotSupportedProperties",
"Message": "Podane pola nie odpowiadają specyfikacji"
}
```

- Status = 412 - dane zostay zmodyfikowane przez innego uytownika lub proces, Body:

```
{
  "HttpCode": "412",
  "MessageCode": "CustomerModifiedByAnotherUserOrProcess",
  "Message": "Dane klienta zostały zmodyfikowane przez innego uytownika lub proces"
}
```

- Status = 422 - bd walidacji. Body:

```
{
  "ValidationErrorCodes": [
    { "Property": "PhoneNumber", "Error": "Invalid", "Value": "AAAA" },
    { "Property": "FirstName", "Error": "Missing" },
    { "Property": "CustomerProperties.Property1", "Error": "Invalid", "Value": "Value1" } ]
}
```

Możliwe wartości "ValidationErrorCodes.Error":

Missing - brak wymaganej danej,

Invalid - błędny format,

InvalidReadOnlyProperty - pole z CustomerProperties jest tylko do odczytu,

OverridingReadOnlyProperty - pole z CustomerProperties jest niezgodne z oczekiwanym typem

"Value" zawiera warto powodujc bd walidacji.

3.5. Pobranie imienia klienta

Request

- GET `https://{host}/loyalty/v1/customers/{cardNumber+phoneNumber}/first-name`

Response

- Status = 200, Body:

```
"Andrzej"
```

- Status = 403, Patrz [opis bdu](#) w punkcie: 3.2
- Status = 404 - klient o zadanym numerze karty i telefonu nie istnieje. Body:

```
{
  "HttpCode": "404",
  "MessageCode": "CustomerNotFoundByCardAndPhoneNumber",
  "Message": "Klient o zadanym numerze karty i telefonu nie istnieje"
}
```

3.6. Powiązanie karty Single z kartą Single/Master

Request

- POST `https://{host}/loyalty/v1/customers/current/master-card/{MasterCardNumber}`
- Body - brak

Response

- Status = 204
- Status = 403, Patrz [opis bdu](#) w punkcie: 3.2

- Status = 404 - klient identyfikowany po MasterCardNumber nie istnieje. Body:

```
{
  "HttpCode": "404",
  "MessageCode": "CustomerNotFoundByMasterCardNumber",
  "Message": "Klient o zadanym numerze karty nie istnieje"
}
```

- Status = 422 - błąd walidacji. Body:

```
{
  "HttpCode": "422",
  "MessageCode": "SlaveCustomerCannotBecomeMaster",
  "Message": "Karta o zadanym numerze karty jest kartą Slave"
}
```

Możliwe wartości "MessageCode":

SlaveCustomerCannotBecomeMaster

SlaveCountLimitExceeded (przekroczona konfigurowalna ilość (domyślnie 10) kart Slave powiązanych z zadaną kartą Master)

3.7. Usunięcie powiązania karty Slave z kartą Master

Request

- DELETE `https://{host}/loyalty/v1/customers/current/master-card/`

Response

- Status = 204
- Status = 403, Patrz [opis błędu](#) w punkcie: 3.2

3.8. Oznaczenie komunikatu przeznaczonego dla klienta jako 'Przeczytany'

Request

- DELETE `https://{host}/loyalty/v1/customers/current/unread-messages/{Id}`

Response

- Status = 204
- Status = 403, Patrz [opis błędu](#) w punkcie: 3.2

3.9. Zmiana hasła klienta

Request

- POST `https://{host}/loyalty/v1/customers/current/password`

Body:

```
{
  "OldPassword": "Qwerty1@",
  "NewPassword": "Qwerty1#"
}
```

W przypadku nadania hasła po raz pierwszy OldPassword = PIN karty.

Response

- Status = 204
- Status = 403, Patrz [opis błędu](#) w punkcie: 3.2