

Priceless Specials Webview

Overview

This documentation describe integration into Priceless Specials platform using few services provided by Verestro.

We should highlight few main products:

- Full SDK (called in document WEB SDK) - web based service, that allows to integrate Priceless Specials product into partners application. It can be embedded in the mobile application or provide service after redirect to it from partners web panel.
- Goal widget - web based micro app, that can display selected reward and progress % for particular card info.
- Company API - API that allows partner to interact with Priceless Specials product specific data e.g., *customer point adjustment*. The more information about API specifications is [here](#).

API Domain

All methods are available on below URL except initialization method.

Endpoint URLs:

Stage: <https://rpm-management.upaidtest.pl/>,

Prod: <https://rpm.secure-verestro.com/>.

Security and authentication

For backend-backend communication, it is required to attach appropriate certificates to the requests sent. This applies to:

- Registration process - /company/enroll, company/add_card,
- Endpoints described in Chapter 6.

Method issuer/initialize_sdk does not require certificate.

Certificate generation process.

First, the Partner needs generate a CSR (Certificate Signing Request).

The command to generate the CSR:

```
openssl req -new -newkey rsa:4096 -keyout client.key -out client.csr -nodes -subj  
'/CN=bankname cert/emailAddress=rpm@bankname.pl'
```

Once the CSR is generated, it should be passed to Verestro. Verestro will sign the CSR and provide x509 certificates based on it:

- certificate CERT.CRT - used for backend-backend communication,
- certificate CA.CRT - (recommended) used for backend-backend communication,
- certificate to sign the trustedIdentityToken (one of the payload parameters of initializeSdkToken).

JWS tokens must have additional header:

| Name | Description |
|------|---|
| x5c | Parameter contains the X.509 certificates chain (RFC7515#section-4.1.6) |

PAN hash generating

PAN hash is an identifier in the Partner-Verestro communication and specifies the card for which the action should be performed.

It is used in:

- Endpoints described in Chapter 6,
- As a parameter in the trustedIdentityToken payload.

PAN hash generation process:

Hash value is as SHA-256 HMAC, please see links below for more details:

RFC: <https://tools.ietf.org/html/rfc4868#page-3>,

Wiki: <https://en.wikipedia.org/wiki/HMAC>.

Test and prod value of key used to calculate HMAC will be delivered by Verestro at the later stage of integration process.

To validate your implementation please check plain and hashed values below:

| Plaintext (HEX) | Hash |
|------------------------------|---|
| 5555444433332222 | 4f64c445c859f7e53209e0091a5faef7e8b3ebbad899fbf8c74df09a6bfe5646 |
| 6984576897634895763948576 | 4b2eab65ab16183fa6ac8a8b12ad690890db98c5ce20e6d56aa037b723bbe84 |
| someTestValue398048096859607 | 29596a78a7382e90159d8ec78a8d37baff57d05f676c0607dd7fb24b0396270ce |

Encrypted PAN number

The structure of encrypted card number sent in the requests for methods:

- POST company/enroll
- POST company/add_card
- POST company/card_replace

JWE structure

Header:

```
{  
"enc": "A256CBC-HS512",  
"alg": "RSA-OAEP-256"  
}
```

Payload:

```
"{"privateClaim":{"cardNumber":"5555666677778888"}}"
```

To encrypt the JWE, use the public key returned by GET company/public_key

Registration flow

Registration method are used to add the user and his cards to the MRS system and create an account on the Priceless Spiecial platform. Registration is possible without providing an email address, in which case the user will have to verify it in first log in.

Card status

This endpoint returns information about card and consents status in MRS. If the card does not exist in the system, then the API returns: **code 404**.

Headers:

| Name | Value |
|--------------|------------------|
| Accept | application/json |
| Content-Type | application/json |

Request Details:

HTTP GET/company/card_agreements_status/{panHash}

Responses

| Status, Body | Description |
|--|------------------------------------|
| HTTP_200, SUCCESS { "termsAndCondition": true, "disclosureOfPersonalData": true, "processingPersonalData": true, "marketingInfoEmail": true, "marketingInfoPhone": true, "errorMessage": null } | Success. |
| HTTP_200, SUCCESS { "termsAndCondition": null, "disclosureOfPersonalData": null, "processingPersonalData": null, "marketingInfoEmail": null, "marketingInfoPhone": null, "errorMessage": "CARD_IS_INACTIVE" } | Card is inactive in MRS System. |
| HTTP_200, SUCCESS { "termsAndCondition": true, "disclosureOfPersonalData": true, "processingPersonalData": true, "marketingInfoEmail": true, "marketingInfoPhone": true, "errorMessage": "USER_EMAIL_NOT_VERIFIED" } | User didn't confirm email address. |
| HTTP_404 | The card does not exist in system. |

Add User with Cards

Supported methods

User add - POST company/enroll.

Data structure

For POST methods request body should contain JSON with combination of objects mentioned below.

Request

| Field | Required | Type | Additional information | |
|-------------|--------------------------|------------------|---|--------------------|
| requestId | No | String | External User Identifier provided by client. It can be used as an identifier in system (in standard UUID4). | |
| firstName | Yes | String | | |
| lastName | Yes | String | | |
| phoneNumber | No | String | Should be passed with prefix. | |
| email | No | String | | |
| zipCode | No, recommended | String, optional | format: NN-NNN. | |
| birthDate | No, recommended | String, optional | format: YYYY-MM-DD. | |
| cards | Yes | Array | Array of Card Objects. | |
| | encryptedNumberCard | Yes | String | Encrypted PAN JWE. |
| | programId | No | String | |
| | programIdentifier | No | String | |
| | termsAndCondition | Yes | Boolean | true or false. |
| | disclosureOfPersonalData | Yes | Boolean | true or false. |
| | processingPersonalData | Yes | Boolean | true or false. |
| | marketingInfoEmail | Yes | Boolean | true or false. |
| | marketingInfoPhone | Yes | Boolean | true or false. |

Example:

```
{
  "requestId": "123e4567-e89b-12d3-a456-426614174000",
  "firstName": "John",
  "lastName": "Smith",
  "phoneNumber": "+48777666555",
  "email": "john.smith@gmail.com",
  "zipCode": "25-345",
  "birthDate": "1998-12-12",
  "cards": [
    {
      "encryptedNumberCard": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbnNyeXB0ZWROdW1iZXJkIjoiaWNT
      EzODQzMDMxMjA0MjA0MCJ9",
      "programId": 345353,
      "termsAndCondition": true,
      "disclosureOfPersonalData": true,
      "processingPersonalData": true,
      "marketingInfoEmail": true,
      "marketingInfoPhone": true
    }
  ]
}
```

Responses for Partners that use PAN (primary account number) as BCN (bank customer number)

| Status, Body | Description |
|--|---|
| HTTP_201, SUCCESS | Success (email/userId passed in a request). |
| HTTP_201, SUCCESS { "requestId": "123e4567-e89b-12d3-a456-426614174000" } | Success (email/userId passed in a request). requestid returned to add new cards if user does not verify email address. |

| | |
|--|---|
| <pre>HTTP_201, SUCCESS { "requestId": "123e4567-e89b-12d3-a456-426614174000", "failedPanHashes": [{ "panHash": "a7a89543br958437543bf984375bc3498nvuiey74783" }] }</pre> | Passed card/cards failed during adding to MRS system. |
| <pre>HTTP_400: CARD_ALREADY_EXIST { "failedPanHashes": [{ "panHash": "a7a89543br958437543bf984375bc3498nvuiey74783" }] }</pre> | Passed value is not unique in system. |
| HTTP_400: INVALID_VALUE_CARD | Passed Card value is invalid. |
| HTTP_400: INVALID_VALUE_EMAIL | Passed email is invalid. |
| HTTP_400: INVALID_VALUE_BIRTH_DATE | Passed Birth date is invalid. |
| HTTP_400: CARDS_LIMIT_EXCEEDED | Too many passed Cards. Limit 10 cards. |
| HTTP_403: ACCESS_DENIED | Method is not available. |
| HTTP_401: INVALID_CERTIFICATE | Passed certificate is invalid. |

Responses for banks that use clientId as BCN (bank customer number)

| Status, Body | Description |
|--|---|
| <pre>HTTP_201, SUCCESS { "clientId": "PXPYVJ2HZTW0CK0G4840K4WGGOWOKO" }</pre> | Success (email/userId passed in a request). |
| <pre>HTTP_201, SUCCESS { "clientId": "PXPYVJ2HZTW0CK0G4840K4WGGOWOKO", "requestId": "123e4567-e89b-12d3-a456-426614174000" }</pre> | Success (email/userId passed in a request). requestId returned to add new cards if user does not verify email address. |

| | |
|--|---|
| <pre>HTTP_201, SUCCESS { "clientId":"PXPYVJ2HZTW0CK0G4840K4WGGOWOKO ", "requestId": "123e4567-e89b-12d3-a456-426614174000", "failedPanHashes": [{ "panHash": "a7a89543br958437543bf984375bc3498nvuiey74783" }] }</pre> | Passed card/cards failed during adding to MRS system. |
| <pre>HTTP_400: CARD_ALREADY_EXIST { "failedPanHashes": [{ "panHash": "a7a89543br958437543bf984375bc3498nvuiey74783" }] }</pre> | Passed value is not unique in system. |
| HTTP_400: INVALID_VALUE_CARD | Passed Card value is invalid. |
| HTTP_400: INVALID_VALUE_EMAIL | Passed email is invalid. |
| HTTP_400: INVALID_VALUE_BIRTH_DATE | Passed Birth date is invalid. |
| HTTP_400: CARDS_LIMIT_EXCEEDED | Too many passed Cards. Limit 10 cards. |
| HTTP_403: ACCESS_DENIED | Method is not available. |
| HTTP_401: INVALID_CERTIFICATE | Passed certificate is invalid. |

Add Card to existing account

Supported methods

User add - POST company/add_card.

Data structure

For POST methods request body should contain JSON with combination of objects mentioned below.

Request

| Field | Required | Type | Additional information |
|-------|----------|--------|-------------------------------|
| email | No | String | Email or userId is mandatory. |


```

        "termsAndCondition":true,
        "disclosureOfPersonalData":true,
        "processingPersonalData":true,
        "marketingInfoEmail":true,
        "marketingInfoPhone":true
    }
]
}

```

Responses for banks that use PAN (primary account number) as BCN (bank customer number)

| Status, Body | Description |
|---|---|
| HTTP_201, SUCCES | Success. |
| HTTP_201, SUCCESS <pre> { "failedPanHashes": [{ "panHash": "a7a89543br958437543bf984375bc3498nvuiey74783" }] } </pre> | Passed card/cards failed during adding to MRS system. |
| HTTP_400: CARD_ALREADY_EXIST <pre> { "failedPanHashes": [{ "panHash": "a7a89543br958437543bf984375bc3498nvuiey74783" }] } </pre> | Passed value is not unique in system. Request is rejected, no cards have been added to the system. |
| HTTP_400: INVALID_VALUE_CARD | Passed Card value is invalid. |
| HTTP_400: INVALID_VALUE_EMAIL | Passed email is invalid. |
| HTTP_400: INVALID_VALUE_USERID | Passed userId is invalid |
| HTTP_400: CARDS_LIMIT_EXCEEDED | Too many passed Cards. Limit 10 cards. |
| HTTP_403: ACCESS_DENIED | Method is not available. |
| HTTP_401: INVALID_CERTIFICATE | Passed certificate is invalid. |

Responses for banks that use clientId as BCN (bank customer number)

| Status, Body | Description |
|---|---|
| HTTP_200, SUCCESS { "clientId" : "asdh-adss-sada-sadss" } | Success. |
| HTTP_200, SUCCESS { "clientId": "asdh-adss-sada-sadss", "failedPanHashes": [{ "panHash": "a7a89543br958437543bf984375bc3498nvuiey74783" }] } | Passed card/cards failed during adding to MRS system. |
| HTTP_400: CARD_ALREADY_EXIST { "failedPanHashes": [{ "panHash": "a7a89543br958437543bf984375bc3498nvuiey74783" }] } | Passed value is not unique in system. Request is rejected, no cards have been added to the system. |
| HTTP_400: INVALID_VALUE_CARD | Passed Card value is invalid. |
| HTTP_400: INVALID_VALUE_EMAIL | Passed email is invalid. |
| HTTP_400: INVALID_VALUE_USERID | Passed userId is invalid |
| HTTP_400: CARDS_LIMIT_EXCEEDED | Too many passed Cards. Limit 10 cards. |
| HTTP_403: ACCESS_DENIED | Method is not available. |
| HTTP_401: INVALID_CERTIFICATE | Passed certificate is invalid. |

Download key to encrypt PAN

Supported methods

Download key – HTTP GET /company/enrollment/public_key.

Data structure

Body of this method should be empty.

Responses

| Status, Body | Description |
|--|------------------------------------|
| <pre>HTTP_200, SUCCESS { "publicKey" : "LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUJlQ0lqQU5C Z2txaGtpRzl3MEJBUUWGUUFPQ0FnOEFNSUIDQ2dLQ0FnRU FwTS9IamdEZHXUXVVRWhRnkv2MgpHVHGX5SnIrcHBUDU1j ODFXTSt1Mm03ZDFVdnpHNXdUT1k3OVZVWGDdDdFBjMkNX Z05BcX0aGk4Ymp4WjhOSi9UCjRoMzhya2xoSWH2anJqQ0 w0NUIESUVqOTImREt1R1lySzBkQIY4eDhjT2drL09jaUhqblF TVytoL05RTFRpTHIKTHkzbzV0dDhOaVNVWVjuTjlmZEo3cyt aRjdFdU1kMkhuMm1iSkVOS2MwYXRXekZtZkFQbnM4SmRV WWdNWWQxSgpsbll0cXdXVWdGYUxnVUplaDhyZGRrd3lqe HdUZVZ5M0IFQ1hYMHNMcWlWTTZ2NURFWGk5Y1ZVV09PT 0FUSG1pCng0a2dreUNESG9uN3FZnJFaZEhQdVISOFRDxBXo xaW9sOGpLNkZTUdZjFsb01Qem9XVTVPeHFRTjVCMUJrdV MKOWJJUFZTZFIUYTdRanB5aGRHbGhSeWRrNFNVYmFTR2d 2cXc1N01VRXjkZ2NKOW4rV2hPNUorc0VOclpOSFZhdQo1cz lCaytiRiZuQXItZEtldEdQZzVtZC94Ulo0NzVwQ1FPVXR5UUEr NnduSitONTIWZlJnTINvaDhKK2lnYkhRCmJ2YTNqcjFqKzA1d kw1V1JhNnR1SzRoeTBQQ29JRutYZGNUTk1uZjdDa3dVUIFs aTIVdeJqOV05bHozL0VBTy8KWII5cnlXOWJqN0J0cTRMU0IIN FR2K2c2NDg4bXVXMENnSGIQRDJGV3NsWUdMcGQ2V3JvQl R0ZCtCajhhUk5jMQo5YXFmcE1DTmZBdUJlMm5Lc1ZITkxJR nlrbEY3TktVGRxcm8vcnBKUEFidFJjb0YvK3JtcUtHUVh2L0d uSWQyCjN3SzhMDUrNnZZeXhmcXNldUtUWGNNQ0F3RUF BUT09Ci0tLS0tRU5EIFBVQkxjQyBLRVktLS0tLQo=" }</pre> | <p>Success. Base 64 encrypted.</p> |

Connection to SDK and widgets

Initialization

To connect you need to send HTTP POST request to `/initialize_sdk` endpoint with required parameters. To choose what front-end way want to load, you need to provide `widget_type` parameter that determines also required parameters.

You can also add the parameter `"redirect": false`, which is optional - the default value is true. That causes that you are not redirected, but you receive a response 201 with the body (in this case, you have to redirect manually):

```
{
  "redirectUrl": "https://something.com"
}
```

Endpoint URLs:

Stage: <https://rpm.upaidtest.pl/issuer>,

Prod: <https://rpm.upaid.pl/issuer>.

Headers:

| Name | Value |
|--------------|------------------|
| Accept | application/json |
| Content-Type | application/json |

cURL example:

```
curl -X POST "http://rpm.upaidtest.pl/issuer/initialize_sdk" -H "Accept: application/json" -H "Content-Type: application/json" -d '{"initializeSDKToken": "\'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0cnVzdGVkSWRlbnRpdHlUb2t1biI6ImV5SmhiR2NpT2lKU1V6STF0aUlzSW5nMVl5STZxeUp0U1VsRk4zcERRMEYwWTB0RFVUkpiR1Z0YkVwS09FVnpWRUZPUW1kcmNXaHJhVWV1ZHpCQ1FWRnpSa0ZFUW1KTlVYTjNRMUZaUkZaUlVvZEZkMHBSVksVSRlZVMUNTvWRCTVZWRLWwRjNUR0pYUmpaaU0yUndXbGRPY21GWfZYaEZRURUW1kT1ZrSkJZMDFEUm1Sb1kyNU90bGxZWkdoTlVUuJNSRUZaUkZaUlVvEVRVlpXWtBkR2NGcEVSvLjOUWtWSFFURlZSVU4zZDB0VFZsRm5WVzA1ZG1SRFFrUlJWRUZsUm5jd2VVMUVRVEJ0YWtsNFRrUkJKNRTVVWkdGR2R6QXdUVVJCTUUXvVvkaE9SRUV3VGxSa1lVMUNaM2hHYWtGVlFtZE9Wa0pCVFUxRVZsWlJVVlZzUlV4V1FrSlZiRkpQVWxaSmQyZG5TV2x0UVRCSFExTnhSMU5KWwP0RVVVVknRVkZWUVVFMFNVTkVvMEYzWjJkSlMwRnZTVU5CVVSS2RXcHJTa0p2WjFoR1NFbHFVSEZEVUhgclRtY3ZRbnB1TkRWFZiAFFTVFoyVm5kdk1YSktkV2cxT0RoT1RqZGxNa2cxT1dVNVRWUkNUbXRpWVRJMFUjZHMlJUY1U5VVJuZFHZV0pZTlVvcLlYyMW1UbnBuU3pkV1UwZG5TMDlYTVRBMVowZDNUM1pQV1RKcFUwSlhka3BDVTNKRLZuTKNhVWhuY1U0NVZtcDBVMWt5YTBZNGFFeHRRM1F6U210dk1YVjBRVTF5UzB4SVUxRlNUbkV2VWVhGbmVXWnZOVXRpVXpkT2R6YzJTB1JTT0V4QmRswndTUzhyVDI1bmRUWkhVb1JIZDBNemRUTnhia05qZVZKU055dDVNSHBKUm5waU1YSm5SV3hNVkVzNFJXWjJVMl02Ww1NMvdFOXhUMjF3WkdjMUwzVlNTRXA1U25aRFQwTk9jRVp2V0M5YVNIaHpWmM1TmtweFlqQk9SVXRtY21WcmFERlRNBGRVYkhSRWQwaGlhR0pNZDB0b05HVXlhMDU0TDFaeUwwbFNARFp3TlhGQmRpdHVZMVl4YVZwVWJXcENRMLZ4Y0hwUk5XczFSa1JLVlcxb2RrdzFRVzFPY1d0b01GRkVar1ZvYTNKc1YwNU5RM0JXZVZCaGJFRlJWwGd5VmxawFNsRjJSMHcwWlZCUMIxUmhSVkk1VDJ4YVlVZVZaV0psZGxM2RmXBRXRaWmpCV05Hb3lPWGM0WmxSNFNXMuTREJLTTJOSlEwaHZiVU00Y2xoMlphWjZRVzlyWVd0T1QwTlJpV2w0WldGawREY3ZTMjFYVkhkYWRHw1lUamxwYXpSRVfUZEExVbkprSzFWU1VHbzFkRkjaV21WQ2FiAgHlVTF0ZVc0dlnsUlRaVZYS3l0WGNXeGhSR3gxVUd0MfdrbEpNVzEwTVhCVWmxVlBjekpwZURoa2VuQjBNMUkzYTBNNVJIVjBUVWxhU0VVMU4ycGtSV0YxTm1SWE4zUkLhMGRGUjFWUlNtaHZjBkZpVvdaNVFYbGxNMkZGTUdKa1ozUnBwV3R0VlM4e1MwRjVXRzU0ZG14d1prZGxTaTkWmpGTmNVbHhXR2x1YUhcQk1UUK9haTh5YVRBM05XYzJVSEF2Ym5KVVQyRkNjRmdyU3paMGNGUlFjemRaYUc1SVNIUnhSRE5CTUdsdmMyWjR0WFZ0VvVsRVFWRkJRazFCTUvkrFUzRkhVMGxpTTBSU1JvSkRkMVZCUVRSSlEwRlJRbXRsuWxVNE5tZERXbkp2TUM5b1dIUk9RUz1QYTFsdfZWwM90bWh5WmxNck5tUlFOME12VG1wd1FXRmp1R3BQWmtFdkx6Umlja3BLUVhGcU4wbElNVGxPWTJabVQwVkdNVTByYzI1TWJHwld0MU50T1RSblJ6WjNaRgc1U0RWUE9UZzNOVXA1ZUd4dmFXSk5UemN6TTNsVmIwWk5XVXN6VjJWTFIzaEJaekZhYjBGSVLXRTFNMlJ5VEU5bVQwYzJOMjldWjNKME1EUlLSFZzZVZCcFZxbHFjemxMYzBz1MRVNXBnBfPulRKRvZXWmpRWE5xSzi1ak1WTjVXa3hCUVZowFRtSTRiV2RPYm5aUVdVSnlARFk1Wm1KblpuVTFjVXRPTmk5cGVWSTRPVVjyVvD0WE1YTjJjWE56YwtnM1IxcFFRbTV0UkN0eVlrSlldWMkozTTB0M1NqSjFTR1pVZDNkNVVtTXdaRmxpWkRkM1l6Tmt0VVZEUjB0emFVbHNSa2RrVTNWUUsyaE1WazVVU21GVFNWZHFhVzVoV1RWYU9wVlBTmk12ZwtSMU1IUlPpVVIyV0ZWcWJVeExlamhsZDB0dVR6RjZSVWn4ZVVSUlQw0WpVRXBUyJwU2F
```

```
r0XBRa1l5V0dzNGVtazNjMFp0VWpkeE9HwkVkr3hDZGxob09IaFhjWEIwZVhKUVLVZE1hMWhMV2xsUVNuSXliRl13UjBKa
E5pdHRUVWhqWkdaSVVXTkpTQzlhS3poU2JEbDVUemx1ZEdGd2VGWnNlVVpuYVdGbk9WwmpVa1JTYmsxeE5GRmia1J0V1V
SWLEzcERRV1pQT0VwYWNXZG1iVzLXUlDWNV6RmxPSFpaWlhGc2VFa3ZjbmR5TjAxd2RYTXplblZYWmtablNUVldiMEpJY
20xNWlWnTRkMkppVWxCdVQxY3hibWxVU2tRNWRFRldUemw2U0ZVM1pIbE5WbHBETkhKc2RIVTnJRU4xUnk5RlDtczV0MjQ
yV2xwTlRVbzJNbU00WjJwUlpIbHVRMk4xUm10dU5FWXlURVpUT1doelltZfPMMFpRTnpKSWmwazBPR1o2WkcxemNtVnJjM
HhwTjBREkswTmhlRzF6WlhoTGFWZDZSakYwUWxnMk5GwjFWMGc1VEdoWGMbDFMM0JJT1ZkUmQwSlDnWHBJTW5vc1pIVlh
aemgzS3p0aWVUVm5QVDBpWFgwLlCxC5DbLRI5mpoRkRCekhYY0I3TjZsNFMtMS02Wk9TVngzc1hsRkg4TnV0T3dVY2p0Z
mk4VE01SUtSaFlGMElFVWFDTjAxU1hHdGdzY3VJ0GZ3ampPVVnlEw5UblhN2xRXzB4aHF1V0Y2aWZBNnRBTfBWVDFZa0V
oaHRTT293Y3hWUVc1aF9yRU9QbmxLWmFTR2QzQ21WdU02dkdodFVsdWh5UXpRakloTFhTaHBpQLLPLThSaG14dnI2Ykc5Q
XA3WVlyRGV5Ywgcly2T1F0ek9GNkh0NUM4ZUFlyMrvMvExcVQxQ2pUUVVTVQR4NElMREpFRVhLejg4ZE1aWEV1Nk5ZWXE
wZ0RgBgpqa3VIMUxCTzldB2ZRbDY4NGpJSkFpeDFTVfVoS3R1M0Z5RwT0Uy1Wd05jMWZBb3RLZEV2VknZ0o5ZEpmSFd0e
mpoNEVrSlVxU1ltTE450HVBnkLYSnh0Nk0wLURoUW0yNW83M3Q2Y19ZZjEwbXhvWDFIRFFsZGk5WXI3c1NaMGVRSy10c2F
6MjLSMkxZMHhMnFneVBnRULCZFp6TE95M0JJbmw3NF9yeU5VSHFwWF9oUk9Md2JpUW5CLUVjT1ZWVnQ4Szz4bDBmRDM4Y
U5IaGJVUVd3TjJhbmFWUFNsNXZkaLF0R2tuUThFV3Y3bHZ1d0xSUThuSlp4Ukg0akdnMjRBV1hLY1BsNDJyejhWLWJSQmN
hcVpIMDFPMjJ10EF0aXN1MDLTSXY0aGFtekdFaklEVERjNGR5VUNTRTkbmlqU0NadkRUVXY3RFEzd0dUaVJzZ1BJSjEx0
HpueS1EZk9xaDVSWhcxUS1LVzBSd3lkZfVsM3dLX1VjcEM0TkdWYkwzWFiwTDFqVHVlWwJYZkh2aTlMd1k2RlJYU2tEN2V
tcyIsInBhbkhhc2gi0iJhNjU40DM1ZDEwZwVjMTQ3YjJiNDFhMwYwZmJiMjg0YWNkMjZiZTEzZTgzYmQwNtVtYwQwZTQ1M
zE5NWViNjhiIiwib3MiOiJ3aW5kb3dzIiwiaGV2aWwzSWQiOiJjdxN0b21lcilpZC0yMDAwIiwicmV0dXJuVXJsIjoiaHR
0cDovL3d3dy5leGFtcGxlLmNvbS9yZXR1cm4iLCJrZWVwU2Vzc2lvblVybCI6Imh0dHA6Ly93d3cuZXhhbXBsZS5jb20va
2VlcF9zZXNzaW9uX2hhc2giLCJ3aWRnZXRUeXBlioiZnVsbCIsImNvbXBhbmlJZCI6InRlc3Rjc3N1ZXIiLCJleHAiOiI
xNjEwOTY5Mjc2OTE0In0.R-BGPxI_zaVs20eISQVgxtBLEdZkI0mAFvQSRLAISb8\"}"
```

Parameters:

| Name | Required | Type | Description |
|--------------------|----------|--------|--|
| initializeSDKToken | yes | string | JWT token signed by private key delivered by Partner. |
| redirect | no | bool | The default is redirect true. When you set it as false, you will receive a response 201. |

Token payload data:

| Name | Required | Type | Description |
|----------------------|----------|--------|------------------------------|
| trustedIdentityToken | yes | string | Trusted identity token. |
| panHash | yes | string | PAN hash - described in 3.2. |

| | | | |
|----------------|---------------------------|--------|--|
| deviceId | yes | string | Unique ID of device. Base64 encoded. It is a string of numbers and letters that identifies every individual device. It is generated on the device and can be retrieved by any app session with the same way. It should be generated with various parameters like, serial number, os version, screen information, device build, device number, battery information, so it can provide unicity from each device. |
| os | yes | string | One of: "android", "ios", "linux", "macos", "windows". |
| publicKey | no | string | This is public key from device biometrics authentication module. Base64 encoded. |
| returnUrl | yes - for WEB integration | string | Url which will allow to return form RPM to External Issuer System. Recommended in web integration. |
| errorUrl | no | string | Url which will allow to redirect, in case of any errors. Recommended in web integration. <i>Planned for the future.</i> |
| logoutUrl | no | string | Url which will be triggered when log out will appear. Recommended in web integration. If this parameter is not present, user will set default login page of program! <i>Planned for the future.</i> |
| keepSessionUrl | yes - for WEB integration | string | Value signed by EIS, it allows RPM to periodically extend user session in EIS. EIS should be able to validate if it is properly sign with their key. |

Android

```
Android.showMessage(message);
```

Android developers must binding javascript code to android code using @JavascriptInterface annotation. More info [here](#).

iOS

```
window.webkit.messageHandlers.payment.postMessage({ status: message });
```

More info [here](#).

Partner can react on those in various ways that suits given use case or interface.

Events list

There are some rare cases that error will occur, here are list of most common:

- 'ERROR: Missing parameters',
- 'ERROR: Missing rewards',
- 'ERROR: Session expired'.

WEB application

If there is an error, and returnUrl is set, app will redirect to returnUrl so partner can react.

If there is no returnUrl, error page will be displayed.

Initialization of Priceless Specials Webview

Solution is prepared to open in webview and should be implemented in a way, where no iframe is used at all!
Please make sure support for local storage and cookies is enabled.

Base flow:

- Banking mobile app authenticates user,
- Bank's backend system generates trusted identity token,
- Mobile application initializes Native SDK – developed by Bank,
- Native SDK generates missing data:
 - Unique device ID,
 - System name,
 - Public key.

Native SDK opens Priceless Specials Webview by POST request, like described below.

To initialize Priceless Specials Webview within mobile application, it is required to make POST request to RPM backend (API) with parameters (section 3.3). Make sure that parameter widgetType is set to "full".

Example **POST** request body:

```
{  
  
  initializeSDKToken:  
  eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IjZSI6InRlc3QiLCJpYXQiOiJ  
  E1MTYyMzkwMjJ9.tRF6jrkFnCfv6ksyU-JwVq0xsW3SR3y5cNueSTdHdAg  
  
}
```

Partner will perform POST action /initialize_sdk with parameters as described above.

It will trigger communication between WEB APP and RPM API to log in user by provided trusted identity.

Closing of the session

Integration with Web application

In this type, if parameter returnUrl is present, in GUI customer will see button to go back, and will be redirected to returnUrl.

Integration with Mobile application

In mobile integration, parent mobile app can close web view at any time, but in case of some errors or losing session, WEB APP will trigger JavaScript events to inform webview about it.

Session support mechanism

Only for WEB flow!

Partner should share endpoint, that will allow to extend user session, so if user will come back to issuer service, it should be still logged in.

This endpoint should be able to validate, if it is secure to extend user session, so keepSessionUrl should be signed by issuer.

WEB APP will periodically call this endpoint (Request GET) to avoid session timeout on issuer side. After user decide to return, WEB APP will log user out and redirect to partner site (returnUrl).

SDK Customization

Font styles and colors

Priceless Specials SDK allows to customize some parts of application layout:

- Header color,
- Footer color,
- Main font color,
- Font type (in the entire application).

Card selection screen

During the integration process, the partner may choose to enable/disable the card selection screen.

This functionality is enabled by default.

Additional API endpoints

Endpoints that are not required. They are used to get additional information from the system via the API and allow additional functionalities.

The more information about API specifications is [here](#).

Revision #10

Created 7 July 2022 18:00:08

Updated 2 May 2024 11:00:20