OTP Webshop migration to the SimplePay system

Technical documentation

15.08.2020

Contents

Document History

Date	Version	Change
29.05.2019	190529	Original issue
04.06.2019	190604	Adding analytical formats
08.08.2019	190808	Charging cards previously registered with OTP in the case of API v2
28.08.2019	190828	Improving the name given to service environments
16.09.2019	190916	Clarifications - Introduction Further recommended modifications - Payment on a mobile device
13.05.2020	200513	Possible generation of VPOS key on the merchant's admin site
24.06.2020	200624	3DS process for payment transactions involving three parties Introduction of new CSV analytical format
15.08.2020	200815	3DS process for payment transactions involving two parties

1 Introduction

The OTP Mobil system also allows the asynchronously operating Webshop 5.0 interface of the OTP Bank available as of 2017(http://simplepartner.hu/download.php?target=webshop5) to be used for the SimplePay system.

The interface identical with OTP Bank "Webshop" (Middleware, MW) service provides a possibility to initiate transactions via the SimplePay system with the technical solution that was implemented earlier for the use of the OTP Bank's system.
Those who have used the bank's system earlier can join the SimplePay system after only a few modifications.

In case of migration, we assume that bankcard payment on the merchant's website is carried out using Webshop 5.0.

This documentation helps merchants who use the "Webshop" service to easily switch to SimplePay.
The documentation focuses on VPOSs, service availability (test and live), settlements, testing and activation processes.

This documentation does not discuss the technical operation of the Webshop or the SimplePay service.

The documentation and sample code for OTP Bank Webshop 5.0 are available at the following URL:
http://simplepartner.hu/download.php?target=webshop5

1.1 PSD2, SCA, EMV 3D Secure

PSD2: second Payment Services Directive of the European Union on digital financial services, prescribing the application of two-factor authentication or Strong Customer Authentication to bank card payments.

EMV 3D Secure (3DS) on the other hand is a card company standard for the technical implementation of strong customer automation. **The deadline for the introduction of the 3DS technology in Hungary is September 30, 2020.**

See **Appendix 1** for more information on the legislative background of strong customer authentication.

The technical solution to strong customer authentication is under active development in the SimplePay system, as a consequence of which this documentation may be changed frequently! This documentation's latest version can always be downloaded from the following URL:

http://simplepartner.hu/download.php?target=otpvposhu

1.2 3DS implementation in the SimplePay system for Webshop service

Strong customer authentication, effective September 30, 2020, will require extended transaction data to be sent upon initiating a bank card payment transaction. The implementation of strong customer authentication in the OTP Bank Webshop service is

described in the Webshop 5.1 version's documentation, which is accessible at the following URL:
http://simplepartner.hu/download.php?target=webshop51dochu

The implementation of version 5.1 will remain accessible only in the SimplePay system, therefore it will be available only for merchants switching from the OTP Bank Webshop service to the OTP Mobile SimplePay service.
The initiation of transactions with extended data content is described in "**Transaction Types**" hereunder.

Important:
The above mentioned standard applies to the introduction of 3DS version **2.x**. Incidentally, 3DS **1.0** customer authentication is already functioning in the SimplePay system, but it does not require any change affecting merchants.
The currently operating 3DS 1.0 in SimplePay is scheduled to be replaced by version 2.x by the September deadline.

1.3 Compatibility

Initiating the transactions may fail due to compatibility issues in the following cases:
- the merchant system operates in accordance with technical criteria predating the OTP Bank Webshop 5.0 version
- transaction initiation uses the "startWorkflow" method

1.4 Unsupported functions
- interface for maintaining customer registration data
- SZÉP card acceptance

1.5 SimplePay system documentations

The documentation and sample code for OTP Mobil SimplePay are available at the following URL: All technical details not discussed in this documentation shall be governed by the text contained therein.

Three-party payments
Sample code: http://simplepartner.hu/download.php?target=v21sdk
Documentation:
Hungarian version: http://simplepartner.hu/download.php?target=v21dochu
English version: http://simplepartner.hu/download.php?target=v21docen

Two-party payment by a stored card:
Sample code: http://simplepartner.hu/download.php?target=v21cardstoragesdk
Documentation:
Hungarian version:
http://simplepartner.hu/download.php?target=v21cardstoragedochu
English version: http://simplepartner.hu/download.php?target=v21cardstoragedocen

Two-party payments by card details:
Sample code: http://simplepartner.hu/download.php?target=v21autosdk
Documentation:
Hungarian version: http://simplepartner.hu/download.php?target=v2autodochu
English version: http://simplepartner.hu/download.php?target=v2autodocen

2 SimplePay systems

SimplePay is made up of two payment systems that are completely separated from each other. One of the systems can be used for transaction tests during the development, while the other for the live transactions. The two systems are not connected to each other in any way.

2.1 Sandbox, test system in cases where OTP Webshop is used

It can only be used to initiate test transactions.
Only the test cards found on the payment page can be used to initiate transactions.

2.1.1 URL's for SOAP requests

Until now in OTP Bank's VPOS service environment
https://www.otpbankdirekt.hu/mwaccesspublic/mwaccess

In the SimplePay sandbox service environment:
https://sandbox.simplepay.hu/mw/mw/pspHU

2.1.2 URLs in case of payment page redirection

Until now in OTP Bank's VPOS service environment
https://www.otpbankdirekt.hu/webshop/do/webShopVasarlasInditas

In the SimplePay sandbox service environment
https://sandbox.simplepay.hu/pay/pay/webshop/do/webShopVasarlasInditas

2.1.3 Test VPOS key

Test transactions can be initiated using the **test** VPOS used in OTP Banking service.

VPOS key: #02299991
VPOS currency: HUF

VPOS key: #02299992
VPOS currency: EUR

VPOS key: #02299993
VPOS currency: USD

2.1.4 VPOS name

It is identical to the name of the key. It can be used as POS ID in the followings when initiating transactions:
<posid>**#02299991**</posid>

No additional settings are necessary in the sandbox system.

2.2 Live payment system when using the OTP Webshop

It can only be used to initiate live transactions.
Separate merchant accounts can be created for more currencies and for more websites under a single contract (merchant).
The VPOS used in the OTP Bank system belongs to such SimplePay merchant accounts.

The name of the merchant account (Merchant identifier or MERCHANT) can be found on the merchant admin interface.
The value of MERCHANT is in the **MWxxxxxx** format. When transactions are managed, this needs to be used as the POS ID without the MW prefix.

2.2.1 URL's for SOAP requests

In OTP Bank's VPOS service environment
https://www.otpbankdirekt.hu/mwaccesspublic/mwaccess

SimplePay in a live service environment
https://securepay.simplepay.hu/mw/mw/pspHU

2.2.2 URLs in case of payment page redirection

In OTP Bank's VPOS service environment
https://www.otpbankdirekt.hu/webshop/do/webShopVasarlasInditas

SimplePay in a live service environment
https://securepay.simplepay.hu/pay/pay/webshop/do/webShopVasarlasInditas

2.2.3 The content of the VPOS key

The live merchant VPOS key applied for the OTP Bank can be used in the live system or a new key can be generated in the SimplePay system. It is the decision of the merchant which one it wishes to use. The two solutions are technically equivalent.

When using the OTP VPOS key
In this case, the contents of the merchant's former VPOS key (**123456.pubKey**) must be setup in our system. In the SimplePay merchant admin system (https: //admin.simplepay.hu/admin/), the content of the public key can be entered in the field "**Public key**" on the "**Account manager/Technical data**" page.

Using a new VPOS key
In this case, a new key can be generated in the SimplePay system. In the SimplePay merchant admin system (https://admin.simplepay.hu/admin/), the key can be generated on the "**Account manager/Technical data**" page.
A new key can be generated by clicking on the "**Key generation**" button, whose private elements can be downloaded and the public element will be setup in the system automatically.

2.2.4 VPOS name

It is the same as the name of the SimplePay merchant account without the MW prefix (MERCHANT). Its value can be used as POS ID in the following.

For example, if the value of the MERCHANT is **MW123456**, it can be used without the MW prefix (**123456**) in the followings as a POS ID: <posid>**123456**</posid>

3 Transaction types

3.1 Three-party payments

No further settings are necessary apart from the above for enabling the payment service to continue operating in the current way in the SimplePay system.

Attention must, however, be paid to the fact that in contrast with earlier requirements, strong customer authentication, effective September 30, 2020, will require extended transaction data to be sent.

3.2 Three-player payment transactions with strong customer authentication (SCA, 3DS).

The above mentioned Webshop 5.1 version receives, and transmits to the card issuing bank, the data required for strong customer authentication.

The data must be sent in the "**isAreq**" field. The location and content of the **isAreq** field are **base64**-encoded data described in the **JSON** architecture. The description of the JSON structure and the fields' data content **are to be found in section 5.1.1. in the above mentioned Webshop 5.1 documentation.**

SimplePay passes on the content of the isAreq field when a payment is initiated. The bank also takes these into consideration when authorising a transaction, which is why sending these and the authenticity of the data are crucial to the success of the transaction!

The data are checked by the SimplePay system for format but their content is validated only by the card issuing bank. The following aspects are reviewed by the SimplePay system in the format check.

In case the trading system sends data in the traditional way or the **isAreq** also contains the same field, the content of the **isAreq** will be forwarded. For example, such data could be the **isMailAddress** parameter and the email parameter within the **isAreq**. If both are sent then **isAreq/email** is evaluated.

Is the content of the isAreq field base64-encoded – if there is such a field?
If it is not, the error message is this: **HIBASBASE64FORMATUM**

If there is an isAreq field, is its content a valid JSON string after base64-decoding?
If it is not, the error message is this: **HIBASJSONFORMATUM**

3.2.1 The minimum necessary data for a successful 3DS process

Not all merchants may have all of the data listed in the **isAreq** JSON string available when starting a transaction, however, sending the following is essential for a successful transaction to take place.

billAddrCity: city
billAddrCountry: country
billAddrLine1: address
billAddrPostCode: postal code
billAddrState: county
email: the buyer's e-mail address

In case the data required for any of these fields is not available, the entire field needs to be removed from the JSON string; in other words, to avoid any misunderstanding, only items to which the merchant's system can add real content to, are worth transmitting to the SimplePay system.
Particular attention must be paid to making sure that data transmitted in the JSON structure remains at the required level, i.e. the "**billAddrCity**" should always be under "**cardholder**", regardless of what other elements have been removed from the JSON structure.

The following JSON structure is associated with the above minimum necessary data:

```
{
   "cardholder":{
      "billAddrCity":"billAddrCity",
      "billAddrCountry":"222",
      "billAddrLine1":"billAddrLine1",
      "billAddrPostCode":"billAddrPostCode",
      "billAddrState":"111",
      "email":"aaa-bbb@example.com",
   },
}
```

3.2.2 Recommended data for a successful 3DS process

Transmitting the following data is an option but if they are available it is recommended that they be transferred in the **isAreq**.

threeDSReqAuthMethod: the cardholder's way of registration on the merchant's side
possible values:
01: guest
02: ID registered with the merchant
05: third party ID in the merchant system (Facebook, Google account, etc.)

billAddrLine2: second line of address
billAddrLine3: third line of address
shipAddrCity: city of delivery
shipAddrCountry: country of delivery
shipAddrLine1: delivery address
shipAddrLine2: delivery address second line
shipAddrLine3: delivery address third line
shipAddrPostCode: postal code of delivery
shipAddrState: county of deliver
mobilePhone, or homePhone, or workPhone: any phone number of the customer
cc: country code
subscriber: telephone number

Particular attention must be paid to making sure that data transmitted in the JSON structure remains at the required level, i.e. in the sample below "**cc**" is to be below "**cardholder**" / "**mobilePhone**".

JSON structure supplemented with optional data:

```
{
   "threeDSRequestor":{
      "threeDSRequestorAuthenticationInfo":{
         "threeDSReqAuthMethod":"02"
      }
   },
   "cardholder":{
      "billAddrCity":"billAddrCity",
      "billAddrCountry":"222",
      "billAddrLine1":"billAddrLine1",
      "billAddrPostCode":"billAddrPostCode",
      "billAddrState":"111",
      "email":"aaa-bbb@example.com",
      "mobilePhone":{
         "cc":"cc",
         "subscriber":"subs"
      },
      "shipAddrCity":"shipAddrCity",
      "shipAddrCountry":"111",
      "shipAddrLine1":"shipAddrLine1",
      "shipAddrPostCode":"shipAddrPostCode",
      "shipAddrState":"222"
   },
}
```

3.2.3 Further optional data

Any other data in the JSON described in the documentation can be sent optionally. They may be of relevance if the merchant can send them with valid content.

3.2.4 Data Transfer Declaration

The merchant transfers the order/customer data to a third party in the **isAreq** content, therefore the **customer must** explicitly **accept the data transfer declaration**.

The data transfer declaration and the way it is to be positioned are to be found in the SimplePay API v2 documentation in "**Data Transfer Declaration**".

3.2.5 Unknown buyer data

If the buyer's data are not known in the merchant system, they can be given by the buyer on the payment page as a consequence of the following variables.

Such variables are already available in the Webshop 5.0 version.

```
<isMailAddressNeeded>true</isMailAddressNeeded>
<isNameNeeded>true</isNameNeeded>
<isCountryNeeded>true</isCountryNeeded>
<isCountyNeeded>true</isCountyNeeded>
<isZipcodeNeeded>true</isZipcodeNeeded>
<isStreetNeeded>true</isStreetNeeded>
```

3.3 Two-party payments for card registrations

The card registration and the payment with registered card functions can still be used in the SimplePay system, too.

In SimplePay, this API can still be used to store a card and the stored card can be used for further transactions.

The tokens that belong to the cards stored in the OTP system can still be used via the SimplePay system. In this case, the SimplePay VPOS and the earlier OTP VPOS will be merged and transactions can still be initiated with the tokens registered earlier.

The card is always registered in a three-party registration fee. After that can be the registered cared debited in a two-party way.

3.4 Three-party payments with recurring card registration.

Recurring payments are transactions that the merchant initiates at regular intervals without debiting the registered card without the presence of the customer. Recurring can be used, for example, for regularly recurring collections, subscription management, and so on.

Recurring payments require a three-party card registration payment (**INITIATINGWEBSHOPPAYMENT**), after which the card provided during registration can be debited in a two-party way (**WEBSHOPPAYMENTTWOPARTIES**).

The current interface requires the following variables during customer or card registration during **INITIATINGWEBSHOPPAYMENT.**
isConsumerRegistrationNeeded = true
isConsumerRegistrationId = abc123

3.4.1 isRecurringNeeded

In addition to these known elements, henceforth it is necessary to send the **isRecurringNeeded** parameter, as a result of which the SimplePay system will perform the given customer registration or card registration as a recurring registration.
isRecurringNeeded = true

3.5 Two-party recurring payments

The card used during recurring registration can be debited by the merchant's system without the customer's presence, for example on a monthly, weekly, etc. basis in the case of regular debits.

For these payments, the current **WEBSHOPPAYMENTTWOPARTIES** requests can still be used. In the requests, the ID specified in the **isConsumerRegistrationId** field at the time of registration must still be sent.

Moreover, the following elements must be added to the request.

3.5.1 isAreq

In addition to the data described for three-party payments, in this case the **cardholderName** field also belongs to the required data, as there is no payment page where the customer can enter this.

```
{
   "cardholder":{
      "billAddrCity":"billAddrCity",
      "billAddrCountry":"222",
      "billAddrLine1":"billAddrLine1",
      "billAddrPostCode":"billAddrPostCode",
      "billAddrState":"111",
      "email":"aaa-bbb@example.com",
      "cardholderName":"",
   },
}
```

3.5.2 isType

This field is used to indicate whether or not the customer is present during the transaction.

Its value can be one of the following

- **CIT** (Customer Initiated Transaction) a transaction initiated in the presence of the customer
- **MIT** (Merchant Initiated Transaction) a transaction initiated without the presence of the buyer
- **REC** (Recurring) is a transaction that is initiated **regularly** without the presence of the customer

In the case of a two-party payment after **recurring** registration, this value can only be **REC**, as the transaction is initiated by the merchant with some regularity without the presence of the customer.

3.5.3 isRecurringNeeded

In addition to these known elements, henceforth it is necessary to send the following parameter, as a result of which the SimplePay system will send the transaction to the card issuing bank as a recurring payment.
isRecurringNeeded = true

3.6 Three-party payments with oneclick card registration

In the case of Oneclick card registration, the registered card is saved by the merchant as a convenience service, which can be debited with the customer's presence for subsequent payments.

The current interface **requires** the following variables during customer or card registration during **INITIATINGWEBSHOPPAYMENT.**
isConsumerRegistrationNeeded = true
isConsumerRegistrationId = abc123

The points mentioned above are not different from the previous registration option. However, the card provided later during registration can only be debited in the presence of the customer with a **WEBSHOPPAYMENTTWOPARTIES** request.

3.7 Two-party oneclick payments

The card used during the oneclick registration can be further charged by the merchant's system in the customer's presence.

For these payments, the current **WEBSHOPPAYMENTTWOPARTIES** requests can still be used. In the requests, the ID specified in the **isConsumerRegistrationId** field at the time of registration must still be sent.

Moreover, the following elements must be added to the request.

3.7.1 isAreq

A detailed description of **isAreq** can be found above at the three-party payments. In addition to the data described for three-party payments, in this case the **cardholderName** field also belongs to the required data, as there is no payment page where the customer can enter this.

```
{
   "cardholder":{
      "billAddrCity":"billAddrCity",
      "billAddrCountry":"222",
      "billAddrLine1":"billAddrLine1",
      "billAddrPostCode":"billAddrPostCode",
      "billAddrState":"111",
      "email":"aaa-bbb@example.com",
      "cardholderName":"",
   },
}
```

3.7.2 isType

This field is used to indicate whether or not the customer is present during the transaction.
Its value can be one of the following

- **CIT** (Customer Initiated Transaction) a transaction initiated in the presence of the customer
- **MIT** (Merchant Initiated Transaction) a transaction initiated without the presence of the buyer
- **REC** (Recurring) is a transaction that is initiated **regularly** without the presence of the customer

In the case of a three-party payment after **onclick** registration, this value can be **CIT** or **MIT**, as the transaction is initiated by the merchant with or without the presence of the customer, nevertheless, it is not a regular but an occasional purchase.

An example for **CIT** is when the debit is initiated immediately upon the interaction of the present customer. Although the payment transaction is technically performed by a

WEBSHOPPAYMENTTWOPARTIES request, during which there is no payment page, due to the presence of the customer, this can actually be interpreted as a three-party payment.

An example for **MIT** s when a customer's interaction only stores payment information and initiates it at a later time after a check that is relevant to the merchant, such as some merchant, or by bundling at night.

3.7.3 isBrowser

In the case of a **CIT** transaction, the parameters of the customer's **browser**. This piece of data is also part of **3DS** authentication, which is forwarded to the card issuing bank.
The location and content of the **isBrowser** field are **base64**-encoded data described in the **JSON** architecture.

details of the **browser** array

- **accept**: Value of accept http header
- **agent**: Value of user-Agent http header
- **ip**: IP of browser source
- **java**: whether the browser supports java applets; in javascript: *navigator.javaEnabled()*
- **lang**: browser language; in javascript: *navigator.language*
- **color**: color depth of browser; in javascript: *screen.colorDepth*
- **height** = browser screen height; in javascript: *screen.height*
- **width** = browser screen width; in javascript: *screen.width*
- **tz** = browser time zone; in javascript: *new Date().getTimezoneOffset()*

JSON pattern


```
{
"accept":"text\/html,application\/xhtml+xml,application\/xml;q=0.9,image\/webp,image\/apng,*\/*;q=0.8,application\/signed-exchange;v=b3;q=0.9",
  "agent":"Mozilla\/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit\/537.36 (KHTML, like Gecko) Chrome\/83.0.4103.106 Safari\/537.36",
  "ip":"94.199.53.96",
  "java":false,
  "lang":"hu-HU",
  "color":24,
  "height":1920,
  "width":1080,
  "tz":-120
}
```


3.7.4 redirectUrl

If **3DS authentication by the bank is unsuccessful**, authorisation is **not** yet performed.

The merchant's system needs to be prepared for when during the 3DS process a card issuing bank may require the cardholder to interactively identify himself/herself (**challenge**) during **CIT** transaction. Identification requires customer interaction. In this case, in response to a **WEBSHOPPAYMENTTWOPARTIES** request the SimplePay system returns a URL (**redirectUrl**) to the merchant to which it may redirect the customer.

Because the customer is present during the transaction (**isType = CIT**), the merchant has the option to redirect the customer to the provided URL.

As a result of the above, the merchant system must be prepared to receive the **redirectUrl** field in a synchronous response to a **WEBSHOPPAYMENTTWOPARTIES** request.

3.7.5 challenge

If the merchant's system redirects the customer to the URL received in the **"redirectUrl"** field, the customer is at a place where additional customer authentication required by the card issuing bank is performed.

3.7.6 isBackUrl

The challenge process has to be done in a browser, at the end of which the SimplePay system redirects the customer back to the merchant website.

In order to be able to specify at the end of the potential challenge process where to redirect the customer, it is necessary to enter the URL in the **WEBSHOPPAYMENTTWOPARTIES** request.

It is necessary to send the **isBackUrl** field in the **WEBSHOPPAYMENTTWOPARTIES** request to make the redirection possible to the merchant's website in case of a challenge.

The content of this field is the same as the **isBackUrl** field for three-party payments.

The **isBackUrl** is optional. If not sent, the information about the outcome of the transaction will be announced on the SimplePay page at the end of the challenge process and authorisation.

3.8 Two-party payments by sending card details

By using the function, payments where the customer's card is requested on the merchant's website can still be initiated.

Due to the PCI-DSS compliance of the SimplePay system, merchant side compliance is also necessary in cases when it requests card details from the customer and forwards them in the two-party method.
In order to initiate transactions in the live system, the merchant is required to produce its own PCI-DSS **AOC** (Attestation of Compliance) to SimplePay.

For these payments, the current **WEBSHOPPAYMENTTWOPARTIES** requests can still be used. In the requests the card data must still be sent as before and the current **WEBSHOPPAYMENTTWOPARTIES** requests have to be amended with the following elements.

3.8.1 isAreq

A detailed description of **isAreq** can be found above at the three-party payments. In addition to the data described for three-party payments, in this case the **cardholderName** field also belongs to the required data, as there is no payment page where the customer can enter this.

```
{
   "cardholder":{
      "billAddrCity":"billAddrCity",
      "billAddrCountry":"222",
      "billAddrLine1":"billAddrLine1",
      "billAddrPostCode":"billAddrPostCode",
      "billAddrState":"111",
      "email":"aaa-bbb@example.com",
      "cardholderName":"",
```

```
   },
}
```

3.8.2 isType

This field is used to indicate whether or not the customer is present during the transaction.

Its value can be one of the following

- **CIT** (Customer Initiated Transaction) a transaction initiated in the presence of the customer
- **MIT** (Merchant Initiated Transaction) a transaction initiated without the presence of the buyer
- **REC** (Recurring) is a transaction that is initiated **regularly** without the presence of the customer

In this case, all three types can occur.

If the **customer** was **present on the merchant's side** when the transaction was initiated, it is a **three-party** payment when this value can only be **CIT**.

If the **customer was not present** on the merchant's side when the transaction was initiated, and this is a one time debit, then it is a **two-party payment** when this value can only be **MIT**.

If the **customer was not present** on the merchant's side when the transaction was initiated and it is a **recurring** debit, then it is a **two-party payment** when this value can only be **REC**.

3.8.3 isBrowser

In the case of a **CIT** transaction, the parameters of the customer's **browser**. This piece of data is also part of **3DS** authentication, which is forwarded to the card issuing bank.
The location and content of the **isBrowser** field are **base64**-encoded data described in the **JSON** architecture.

details of the **browser** array

- **accept**: Value of accept http header
- **agent**: Value of user-Agent http header
- **ip**: IP of browser source
- **java**: whether the browser supports java applets; in javascript: *navigator.javaEnabled()*
- **lang**: browser language; in javascript: *navigator.language*
- **color**: color depth of browser; in javascript: *screen.colorDepth*
- **height** = browser screen height; in javascript: *screen.height*
- **width** = browser screen width; in javascript: *screen.width*
- **tz** = browser time zone; in javascript: *new Date().getTimezoneOffset()*

JSON pattern

```
{
"accept":"text\/html,application\/xhtml+xml,application\/xml;q=0.9,image\/webp,image\/apng,*\/*;q=0.8,application\/signed-exchange;v=b3;q=0.9",
  "agent":"Mozilla\/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit\/537.36 (KHTML, like Gecko) Chrome\/83.0.4103.106 Safari\/537.36",
  "ip":"94.199.53.96",
  "java":false,
  "lang":"hu-HU",
  "color":24,
  "height":1920,
  "width":1080,
  "tz":-120
```

```
}
```

3.8.4 redirectUrl

If **3DS authentication by the bank is unsuccessful**, authorisation is **not** yet performed.

The merchant's system needs to be prepared for when during the 3DS process a card issuing bank may require the cardholder to interactively identify himself/herself (**challenge**) during **CIT** transaction. Identification requires customer interaction. In this case, in response to a **WEBSHOPPAYMENTTWOPARTIES** request the SimplePay system returns a URL (**redirectUrl**) to the merchant to which it may redirect the customer.

Because the customer is present during the transaction (isType = CIT), the merchant has the option to redirect the customer to the provided URL.

As a result of the above, the merchant system must be prepared to receive the **redirectUrl** field in a synchronous response to a **WEBSHOPPAYMENTTWOPARTIES** request.

3.8.5 challenge

If the merchant's system redirects the customer to the URL received in the **"redirectUrl"** field, the customer is at a place where additional customer authentication required by the card issuing bank is performed.

3.8.6 isBackUrl

The challenge process has to be done in a browser, at the end of which the SimplePay system redirects the customer back to the merchant website.

In order to be able to specify at the end of the potential challenge process where to redirect the customer, it is necessary to enter the URL in the **WEBSHOPPAYMENTTWOPARTIES** request.

It is necessary to send the **isBackUrl** field in the **WEBSHOPPAYMENTTWOPARTIES** request to make the redirection possible to the merchant's website in case of a challenge.

The content of this field is the same as the **isBackUrl** field for three-party payments.

The **isBackUrl** is optional. If not sent, the information about the outcome of the transaction will be announced on the SimplePay page at the end of the challenge process and authorisation.

4 Settlement

The SimplePay system can provide the formats and delivery paths used in the OTP Webshop, too.

4.1 Transactional analytics file format

The SimplePay system can currently provide the following OTP settlement formats.

- CSVD (in case of accepting HUF, EUR and USD)
- K01D (in case of accepting HUF)
- K04D (in case of accepting EUR and USD)
- K05D (in case of accepting HUF)

In addition to this, the proprietary settlement file format of SimplePay can also be selected (in case of accepting HUF, EUR and USD), a sample of which can be provided as an attachment to the settlement-related documentation below.

Hungarian version: http://simplepartner.hu/download.php?target=merchantguidehu
English version: http://simplepartner.hu/download.php?target=merchantguideen

4.2 New CSV transaction analytics format

In 2020 H2 OTP Bank will standardise the settlement analytics formats and provide the data in a single format which will be different from the above. The SimplePay system will also be enabled to use the new analytics format, and in parallel with the Bank it will be capable of generating analytics files in the new format as well.

At the same time, the previous formats listed above will also continue to be available in the SimplePay system. In the cases where the merchant side business logic or the automated data processing system is based on previous formats, the system will continue to provide transaction data in unchanged form.

4.3 Transactional analytics delivery method

The SimplePay system can send the analytics files to the merchant using the following methods.

- e-mail
- OTP SFTP server (in a new folder on the remote.otpbank.hu:17626 server)
- SimplePay's own SFTP server
- Hypex

5 Testing and deployment

Since the merchant will initiate transactions in the SimplePay system after the migration, the live system will be used in accordance with the deployment rules of SimplePay.

The following testing points cannot be interpreted for the OTP Webshop, therefore testing these is either not necessary or not possible.

- IPN reception
- OneClick payments and cardSecret
- unique tokens in case of recurring payments

Once the SimplePay-side testing of the merchant system is successful, the live system will be deployed.

5.1 Three-party payments

The pre-deployment testing rules of **three-party payments** can be found in the chapter titled "**Testing**" in the SimplePay development documentation.
Hungarian version: http://simplepartner.hu/download.php?target=v21dochu
English version: http://simplepartner.hu/download.php?target=v21docen

5.2 Two-party payments for card registrations

Two-party payments with card registration require further testing. The pre-deployment testing rules can be found in the chapter titled "**Testing**" in the SimplePay development documentation.
Hungarian version: http://simplepartner.hu/download.php?target=v21cardstoragedochu
English version: http://simplepartner.hu/download.php?target=v21cardstoragedocen

5.3 Two-party payments by sending card details

The pre-deployment testing rules of two-party payments using card data sending can be found in the chapter titled "**Testing**" in the SimplePay development documentation.
Hungarian version: http://simplepartner.hu/download.php?target=v2autodochu
English version: http://simplepartner.hu/download.php?target=v2autodocen

6 Recommended modifications

Due to the added services, the technical possibilities of SimplePay over the previous deployment requirements of the OTP Webshop, or the changes in the regulatory environment, it is recommended to rethink the existing payment process.

6.1 Displaying transaction identifiers

Where the transaction identifier is not visible to the customer, it is recommended to display it so as to allow a more seamless management of customer queries.

6.2 Displaying information

In merchant systems where no information is provided in case of failed or cancelled payments, it is recommended to display information to the customer about the outcome of the payment. In the below documentation of three-party payments, the chapter titled "**Information depending on the result of the transaction**" contains relevant information.

Hungarian version: http://simplepartner.hu/download.php?target=v21dochu

English version: http://simplepartner.hu/download.php?target=v21docen

6.3 Accepting the Data Transfer Declaration

Before any personal data concerning the customer is transferred, the customer must accept the Data Transfer Declaration. This is currently recommended but with strong customer authentication going live on September 30, 2020, it will be a mandatory requirement upon data transfer.

The placement and the text of the declaration can be found in the chapter titled "**Data transfer declaration**" in the documentation of three-party payments.

Hungarian version: http://simplepartner.hu/download.php?target=v21dochu

English version: http://simplepartner.hu/download.php?target=v21docen

6.4 Accepting the Card storage declaration

Prior to the card registration transaction, the customer must be informed, whereby the customer must explicitly accept the card registration statement. This statement does not replace the data transfer declaration, but is complementary to it in the event of a card registration transaction.

The documentation is available at the URL below:

Hungarian version: http://simplepartner.hu/download.php?target=v21cardstoragedochu

English version: http://simplepartner.hu/download.php?target=v21cardstoragedocen

For **Oneclick** transactions, the declaration can be found in chapter "**Oneclick Card Registration Statement**".
For **Recurring** transactions, the declaration can be found in chapter "**Recurring Card registration statement**".

6.5 Payment on a mobile device

In the case payment is performed in a mobile application using webview, we suggest that you take the recommendations of **Annex I** to the below mentioned documentation into consideration.
This section discusses the issue of social login options through certain (older) webview components that have been disabled by Google and Facebook for security reasons.
However, a number of other problems may be attributed to the obsolete component, so it is recommended that you use other components described in the Annex.
Hungarian version: http://simplepartner.hu/download.php?target=v21dochu
English version: http://simplepartner.hu/download.php?target=v21docen

7 SimplePay API v2

Migration may be carried out using the SimplePay API v2, too.

In this case, the merchant will technically not initiate the transactions in the SimplePay system using the OTP Webshop interface but with the help of its proprietary SimplePay API.
The advantage of this is that it complies with the PSD2 requirements that will soon come into effect, the current regulations related to card companies, as well as all PCi-DSS requirements.
Contrary to this, no additional development will be made on the OTP Webshop service, which means that it will continue operating at its current level of technology and services.

The entire SimplePay API v2 documentation and sample code are available at the address specified in chapter 1.

7.1 Charging cards previously registered with OTP in the case of API v2

Cards previously registered in OTP VPOS can be used by API v2 also.

The following documentation on the technical implementation can be found in the chapter "**Loading cards stored in the OTP Webshop service**".

In addition, the merchant's **current SimplePay** account must be linked to the merchant's **previous OTP VPOS** service in the SimplePay admin system. This opens up an active passage between the two services. and the previously stored card will be available in the current system.
In order for this to happen, indicate your need to do so when concluding the contract.

This solution **cannot be tested in sandbox**, while it requires a real and existing OTP bank card registration that is only available in a live system.

Hungarian version: http://simplepartner.hu/download.php?target=v21cardstoragedochu
English version: http://simplepartner.hu/download.php?target=v21cardstoragedocen

8 Support

For further information or technical support, please contact us at itsupport@otpmobil.com.

For faster processing, please provide us the data identifying the problem or your inquiry.

Transaction
If you have questions regarding transactions, please provide us the **SimplePay** identifier of the payment. The identifier is composed of nine digits.

Merchant account
Regarding technical configurations, please provide the identifier of the merchant's account within the SimplePay system. The identifier is the account's MERCHANT value.

Payment system
The system your inquiry relates to. **Sandbox system** exclusively for tests, and **Live system** for live payment transactions.

Deployment
In case of deployment tests, please contact us via **itsupport@otpmobil.com**, providing the following:
- under which contracted domain name the testable payment was developed
- which account is in use (MERCHANT)
- where we can access the testable system

9 Annexes

I. EMV 3D Secure

Legislative and supervisory bodies of the European Union and of the Member State, the relevant European Union Directive (Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015; **PSD2**), or by adapting it to national legislation, a so-called Mandatory **Strong Customer Authentication (SCA)** is required with effect from **14 September 2019** for the acceptance of all cash substitute currency issued within the European Economic Area.

Due to negotiations with market players, MNB (The Central Bank of Hungary) allows an extended transition period of 12 months from the above date for domestic operators, so the new deadline is 30 September 2020.

Domestic legal background

Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015 has been adapted to the following legislation:

- Act CCXXXVII of 2013 on Credit Institutions and Financial Enterprises
- Act LXXXV of 2009 on the Pursuit of the Business of Payment Services
- Act CCXXXV of 2013 on. Certain Payment Providers

The PSD2 modifications were included, among others, in Act CXLV of 2017 published in the 184th Hungarian Bulletin of 2017 and the Delegated Regulation No. 2018/389 of the European Commission.
With regard to bank card acceptance, the regulation also covers acceptance over the Internet (so-called VPOS). In accordance with PSD2 compliance, and to make card acceptance over the Internet safer, the cardholder's enhanced security check, the so-called EMV 3D Secure 2.0 service, must be introduced and fully implemented by 14 September 2020.

3D Secure in practice

OTP Mobil Kft. fully implements the technology related to the Decree. **If bank card payment is used** in a so-called three-player way, **according to this document,** so the merchant system does not use extra service elements based on card storage, **then the merchant has no further action to make**.

At the same time, the provision of data from the merchant system **required by the** EMV 3D Secure 2.0 **standard is essential for the implementation of 3D Secure.** To do this, the merchant system needs to set the parameters for each transaction initiated with the data specified in the description of the "**start**" request in this documentation.

You can find information about the requirement of the EMV 3D Secure standard at the following link:
https://www.emvco.com/emv-technoligies/3d-secure/

What is EMV 3D Secure 2.0 service, and why is it necessary?

Convenient online purchases have become widespread in recent years, initiated even from mobile devices over the Internet. Along with this, the number and volume of fraud, computer or internet abuse, data theft has increased.

Not only regulators but also card companies and banks are constantly working on new, more efficient solutions to ensure safe and reliable credit card payments. 3D Secure 1.0, currently used for online shopping, allows the payer's identification when making online purchases from a browser. This option is not available for purchases initiated from smart devices, and is only available for mobile phone payments when initiating a payment through a browser (not an application).

EMV 3D Secure 2.0 offers even more secure client authentication and can be used not only for purchases made from browser-driven interfaces, but also for in-app purchases, and for payments over mobile phones and other smart devices. EMV 3D Secure 2.0 relies on customer authentication methods such as biometrics (such as fingerprints or face recognition) or one-time passwords. The transaction transfers more data to the issuing banks than it currently does, enabling the customer to be qualified more thoroughly, and to detect any fraud or abuse quickly and effectively.

What technical preparation does the implementation of 3D Secure 2.0 require?

The law requires that under 3D Secure 2.0, with a few exceptions, cardholders must be identified using strong customer authentication when initiating transactions, using bank cards issued within the European Economic Area, at payment acceptors operating within the European Economic Area. For proper authentication, both the card-issuing bank and the acquiring bank (or the payment acceptor) must use a 3D Secure solution (known as Acquirer Solution MPI or Merchant Plug-in in 3D Secure 1.0) that allows the buyer to be identified.

The card acceptance process via the Internet, when developed and operated on the basis of this documentation, is the standard three-player model, which means that the cardholder will be redirected from the web shop interface to the SimplePay web interface provided by OTP

Mobil Kft. The cardholder will enter the required card information on this payment page, the transaction will take place on this interface, and then the customer will be redirected to the web shop.

When using this payment method, the bank is responsible for the secure management and transmission of the bank card information (between the payment interface and the bank). No technical activities are needed for this on the merchant's side when the merchant sends the purchase information required for the 3D Secure process.