

TXPG

Version	Description
1	Base Document
1.0.2	Fixed Urls
1.0.3	Added Recurrent Payment
1.0.4	EMV 3-D Secure Authentication without Redirecting to TXPG payment Page
1.0.5	Fixed EMV 3-D Secure Authentication without Redirecting to TXPG payment Page Issues, Create order edited.
1.0.6	Added Partial Reversal, small fixes
1.0.7	Added Recurrent Payment CIT
1.0.8	Added P2P Credit
1.0.9	Added Card To Card via TXPG API
1.0.10	Minor fixes
1.0.11	Minor fixes
1.0.12	Added Google Pay

Content

Introduction	4
Certificate creation.....	5
Create Order Purchase	7
Request for Quasi Cash Purchase.....	12
Preauthorization.....	14
Completion.....	18
Reversal.....	21
Partial Reversal.....	24
Refund.....	26
Get Order Details.....	29
Recurrent Payment	30
EMV 3-D Secure v2 Authentication without Redirecting to TWEC PG Page.....	36
Recurrent Payment CIT (Client initiated transaction) with 3DS authentication.....	52
P2P Credit.....	62
Card To Card via TXPG API.....	69
Google Pay™.....	76

Introduction

This document describes TXPG REST API (hereinafter, REST API) used for interaction of external systems, in particular, merchant systems and terminals, with the TranzWare/TranzAxis e-Commerce Payment Gateway software product (hereinafter, TXPG). The document describes the general principles of operation with TXPG REST API and standard scenarios of interaction with TXPG via API.

This document provides manual for Merchants support personnel.

Certificate Creation.

For security reasons merchants communicate with TWEC PG using certificates.

Each merchant receives archive file. After extraction you should run **request.bat** file.

This will generate certificate request.

There will be several fields to fill, some of them already predefined.

Country Name (2 letter code) [GE]: by default Georgian country code, just press Enter,

State or Province Name (full name) [Georgia]: by default Georgia,

Locality Name (eg, city) [Tbilisi]: by default Tbilisi (if your organization is registered in different city – enter its name)

Organization Name (eg, company) []: Name of your organization,

Organizational Unit Name (eg, section) []: Department name (for example - IT Department),

Common Name (eg, YOUR name) []: company web address without www letters (for example - libertbank.ge),

A challenge password []: - If you want to protect your certificate with password enter it, or just press Enter without entering password (If you enter password there may be problems with certificate integration into your system)

An optional company name []: -Additional field for you company name, leave it empty if you don't need it

```
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'privkey.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GE]:
State or Province Name (full name) [Georgia]:
Locality Name (eg, city) [Tbilisi]:
Organization Name (eg, company) []:Libertybank
Organizational Unit Name (eg, section) []:IT Department
Common Name (eg, YOUR name) []:libertybank.ge

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

2 files will be created after finishing this process. They will be located in the same folder where request.bat is.

client.key – This is your Private Key.

client.req – This file is your Certificate Request.

Please send client.req to the processing department of our Bank. This request would be signed and you will receive client.cer

This will be your certificate.

In case you need PEM format you can – place client.req in the folder where request was made and run **GEN_PEM.bat**. Password will be requested and we should press ENTER 3 times, then new client.req will be generated, which you may use in your application.

Create Order Operation (Purchase)

The Create Order (createOrder) operation is an initial and mandatory step when working with the order. The operation involves the merchant or terminal logins. When using the merchant login, the terminalId parameter should be additionally passed in the request.

Request Format

URL	https://{path} where {path} – relative path that is added to the URL address of the FES API Gateway application and indicates the type of the operation being executed (for details, refer to the txpg-frontend-api.yaml file and the current document)
HTTP method	POST/GET/PUT – depends on the operation type
Parameters of HTTP header	<ul style="list-style-type: none"> • Content-Type: application/json; • If HTTP Basic authentication is used: Authorization Basic <auth> where <auth> – compound user login and password represented as a string (in /: format) in the base64 encoding, where login_owner_type is the type of the login owner, i.e. TerminalSys, TerminalUser, Merchant, MultiMerchant, ConsumerOfMerch (see above). For example: Authorization: Basic VGVyYWluYWxTeXMvTG9naW4xOIBhc3N3b3JkMTIzNA== that corresponds to "TerminalSys/Login1:Password1234". • If a login of the external user is used – the session ID, for example: TXPG-User-Session-ID: 210701-17155008-0lhzjr= • Idempotence key, for example: TXPG-Idempotence-Key: 123e4567-e89b-12d3-a456-426652340000

Let's consider Default Create Order variant with order type Purchase:

SOAP

Example of the request url:

<https://libertypaytstest.lb.ge:6443/Exec>

Example of the request body:

```
<?xml version="1.0" encoding="UTF-8"?>
  <TKKPG>
    <Request>
      <Operation>CreateOrder</Operation>
      <Language>RU</Language>
      <Order>
        <OrderType>Purchase</OrderType>
        <Merchant>IA00007A</Merchant>
        <Amount>1</Amount>
        <Currency>981</Currency>
        <Description>xxxxxxx</Description>
        <ApproveURL>https://libertypaytstest.lb.ge/</ApproveURL>
        <CancelURL>https://libertypaytstest.lb.ge/</CancelURL>
        <DeclineURL>https://libertypaytstest.lb.ge/</DeclineURL>
        <AddParams>
          </AddParams>
        </Order>
      </Request>
    </TKKPG>
```

Where:

- **Operation** - Operation Type.
- **Language** - Interface language Language
- **OrderType** - Order Type. The financial transaction to be created. Specify the Purchase value to perform the purchase.
- **Merchant** - Internet shop ID.
- **Amount** - Transaction amount in the minimum units of the transaction currency.
- **Currency** - Transaction currency code (e.g. 981– Georgian Lari, 840 – US Dollar)
- **Description** - Order text description. It is displayed to the customer during payment.
- **ApproveURL** - URL to forward the customer to if the operation is approved. If the parameter is not defined, its value will be taken from the parameter of the same name specified in the merchant settings.
- **CancelURL** - URL to forward the customer to if the operation is cancelled (by clicking the Cancel button).
- **DeclineURL** - URL to forward the customer to if the payment operation is declined.

Example of the response body:

```
<TKKPG>
  <Response>
    <Operation>CreateOrder</Operation>
    <Status>00</Status>
    <Order>
      <OrderID>1718</OrderID>
      <SessionID>alaoznssnmkg</SessionID>
      <URL>https://libertypaytxtest.lb.ge/twpga</URL>
    </Order>
  </Response>
</TKKPG>
```

where:

- **Operation** - Operation Type.
- **Status** - Order status. 00 means no errors.
- **OrderID** – Order Id
- **SessionID** – Session Id or password.
- **URL** - payment page url if not provided use one written below.

After that merchant redirects client to the following URL where client enters payment data and completes payment process.

<https://libertypaytxtest.lb.ge/twpga?id=1718&password=alaoznssnmkg>

JSON

Example of the request url:

https://libertypaytxtest.lb.ge:6444/order?terminalid=41
where:

- **terminalid** – Merchant Id.

Example of Authorization Header:

Authorization: Basic VGVyYWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "order":{
    "typeRid":"TWPG-Purchase",
    "language":"ru",
    "amount":"0.1",
    "currency":"GEL",
    "surcharge":{"status":"Charge", "srcAmount":10.03 },
    "description":"test order",
    "custAttrs":[
      {"rid":"@TWPG@ApproveURL", "valAsStr":"https://libertypay.ge/Shop/out.jsp"},
      {"rid":"@TWPG@DeclineURL", "valAsStr":"https://libertypay.ge/Shop/out.jsp"},
      {"rid":"@TWPG@CancelURL", "valAsStr":"https://libertypay.ge/Shop/out.jsp"}
    ],
    "consumerDevice":{
      "browser":{
        "deviceId":"6996B7A6-24D9-4F5A-BE09-99AD420AAFFF",
        "acceptHeader":"application/json,application/jose;charset=utf-8",
        "ip":"127.0.0.1",
        "tzOffset":"-300",
        "platform":"Mozilla",
        "jsEnabled":true,
        "colorDepth":32,
        "screenW":1024,
        "screenH":800,
        "userAgent":"chrome",
        "language":"Ru-ru",
        "javaEnabled":true
      }
    }
  }
}
```

where:

- **typeRid** - Order type.
- **Language** – Language.
- **amount** - Amount . **1 equals to 1 lari.**
- **currency** – Currency.
- **description** - Additional info
- **consumerDevice** – Client Device parameters
- **surcharge** – Transaction Fee

Example of the response body:

```
{
  "order": {
    "id": 1618,
    "status": "Preparing",
    "password": "1fupcb2qtta5r",
    "url": " https://libertypayxttest.lb.ge/twpga"
  }
}
```

where:

- **id** - identifier of the created order. It should be passed in all requests related to the order.
- **status** – status of the order.
- **password** - password of the order.
- **url** - payment page url if not provided use one written below.

Next we redirect client to

<https://libertypayxttest.lb.ge/twpga?id=1618&password=1fupcb2qtta5r>

where client enters credit card data and proceeds with payment.

Request for Quasi Cash Purchase JSON

Example of the request url:

https://libertypaytxtest.lb.ge:6444/order?terminalid=41
where:

- **terminalid** – Merchant Id.

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "order":{
    "typeRid":"TWPG-QuasiCash",
    "language":"ru",
    "amount":"0.1",
    "currency":"GEL",
    "description":"test order",
    "custAttrs":[
      {"rid":"@TWPG@ApproveURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"},
      {"rid":"@TWPG@DeclineURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"},
      {"rid":"@TWPG@CancelURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"}
    ],
    "consumerDevice":{
      "browser":{
        "deviceId":"6996B7A6-24D9-4F5A-BE09-99AD420AAFFF",
        "acceptHeader":"application/json,application/jose;charset=utf-8",
        "ip":"127.0.0.1",
        "tzOffset":"-300",
        "platform":"Mozilla",
        "jsEnabled":true,
        "colorDepth":32,
        "screenW":1024,
        "screenH":800,
        "userAgent":"chorme",
        "language":"Ru-ru",
        "javaEnabled":true
      }
    }
  }
}
```

where:

- **typeRid** - Order type.
- **Language** – Language.
- **amount** - Amount . **1 equals to 1 lari.**
- **currency** – Currency.
- **description** - Additional info
- **consumerDevice** – Client Device parameters

Example of the response body:

```
{  
  "order": {  
    "id": 1618,  
    "status": "Preparing",  
    "password": "1fupcb2qtta5r",  
    "url": " https://libertypaytxtest.lb.ge/twpga"  
  }  
}
```

where:

- **id** - identifier of the created order. It should be passed in all requests related to the order.
- **status** – status of the order.
- **password** - password of the order.
- **url** - payment page url if not provided use one written below.

Next we redirect client to

<https://libertypaytxtest.lb.ge/twpga?id=1618&password=1fupcb2qtta5r>

where client enters credit card data and proceeds with payment.

Preauthorization

This transaction puts a hold on the customer account to the operation amount. Completion is required later to complete the transaction.

SOAP

Example of the request url:

<https://libertypaytxtest.lb.ge:6443/Exec>

Example of the request body:

```
<?xml version="1.0" encoding="UTF-8"?>
  <TKKPG>
    <Request>
      <Operation>CreateOrder</Operation>
      <Language>RU</Language>
      <Order>
        <OrderType>PreAuth</OrderType>
        <Merchant>IA00007A</Merchant>
        <Amount>1</Amount>
        <Currency>981</Currency>
        <Description>xxxxxxx</Description>
        <ApproveURL>https://libertypaytxtest.lb.ge/</ApproveURL>
        <CancelURL>https://libertypaytxtest.lb.ge/</CancelURL>
        <DeclineURL>https://libertypaytxtest.lb.ge/</DeclineURL>
        <AddParams>
          </AddParams>
        </Order>
      </Request>
    </TKKPG>
```

Where:

- **Operation** - Operation Type.
- **Language** - Interface language Language
- **OrderType** - Order Type. The financial transaction to be created. Specify the Purchase value to perform the purchase.
- **Merchant** - Internet shop ID.
- **Amount** - Transaction amount in the minimum units of the transaction currency.
- **Currency** - Transaction currency code (e.g. 981– Georgian Lari, 840 – US Dollar)
- **Description** - Order text description. It is displayed to the customer during payment.
- **ApproveURL** - URL to forward the customer to if the operation is approved. If the parameter is not defined, its value will be taken from the parameter of the same name specified in the merchant settings.
- **CancelURL** - URL to forward the customer to if the operation is cancelled (by clicking the Cancel button).
- **DeclineURL** - URL to forward the customer to if the payment operation is declined.

Example of the response body:

```
<TKKPG>
  <Response>
    <Operation>CreateOrder</Operation>
    <Status>00</Status>
    <Order>
      <OrderID>1718</OrderID>
      <SessionID>alaoznssnmkg</SessionID>
      <URL>https://libertypaytxtest.lb.ge/twpga</URL>
    </Order>
  </Response>
</TKKPG>
```

where:

- **Operation** - Operation Type.
- **Status** - Order status. 00 means no errors.
- **OrderID** – Order Id
- **SessionID** – Session Id or password.
- **URL** - payment page url if not provided use one written below.

After that merchant redirects client to the following URL where client enters payment data and completes payment process.

<https://libertypaytxtest.lb.ge/twpga?id=1718&password=alaoznssnmkg>

JSON

Example of the request url:

<https://libertypaytxtest.lb.ge:6444/order?terminalid=41>
where:

- **terminalid** – Merchant Id.

Example of Authorization Header:

Authorization: Basic VGVyYWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9Qlw==

Example of the request body:

```
{
  "order":{
    "typeRid":"TWPG-Preauth",
    "language":"ru",
    "amount":"0.1",
    "currency":"GEL",
    "description":"test order",
    "custAttrs":[
      {"rid":"@TWPG@ApproveURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"},
      {"rid":"@TWPG@DeclineURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"},
      {"rid":"@TWPG@CancelURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"}
    ],
    "consumerDevice":{
      "browser":{
        "deviceId":"6996B7A6-24D9-4F5A-BE09-99AD420AAFFF",
        "acceptHeader":"application/json,application/jose;charset=utf-8",
        "ip":"127.0.0.1",
        "tzOffset":"-300",
        "platform":"Mozilla",
        "jsEnabled":true,
        "colorDepth":32,
        "screenW":1024,
        "screenH":800,
        "userAgent":"chorme",
        "language":"Ru-ru",
        "javaEnabled":true
      }
    }
  }
}
```

where:

- **typeRid** - Order type.
- **Language** – Language.
- **amount** - Amount . **1 equals to 1 lari.**
- **currency** – Currency.
- **description** - Additional info
- **consumerDevice** – Client Device parameters

Example of the response body:

```
{  
  "order": {  
    "id": 1618,  
    "status": "Preparing",  
    "password": "1fupcb2qtta5r",  
    "url": "https://libertypayxttest.lb.ge/twpga"  
  }  
}
```

where:

- **id** - identifier of the created order. It should be passed in all requests related to the order.
- **status** – status of the order.
- **password** - password of the order.
- **url** - payment page url if not provided use one written below.

Next we redirect client to

<https://libertypayxttest.lb.ge/twpga?id=1618&password=1fupcb2qtta5r>

where client enters credit card data and proceeds with payment.

Completion

Finishes transaction created with Preauthorization.

SOAP

Example of the request url:

https://libertypaytxttest.lb.ge:6443/Exec

Example of the request body:

```
<?xml version="1.0" encoding="UTF-8"?>
  <TKKPG>
    <Request>
      <Operation>Completion</Operation>
      <Language>RU</Language>
      <Order>
        <Merchant>IA00007A</Merchant>
        <OrderID>1718</OrderID>
        <AddParams>
          </AddParams>
        </Order>
        <SessionID>alaoznssnmkg</SessionID>
        <Amount>1</Amount>
        <Currency>981</Currency>
        <Description>xxxxxxx</Description>
      </Request>
    </TKKPG>
```

Where:

- **Operation** - Operation Type.
- **Language** - Interface language Language
- **OrderType** - Order Type. The financial transaction to be created. Specify the Purchase value to perform the purchase.
- **Merchant** - Internet shop ID.
- **Amount** - Transaction amount in the minimum units of the transaction currency.
- **Currency** - Transaction currency code (e.g. 981– Georgian Lari, 840 – US Dollar)
- **Description** - Order text description. It is displayed to the customer during payment.

Example of the response body:

```
<TKKPG>
  <Response>
    <Operation>Completion</Operation>
    <Status>00</Status>
    <POSResponse>
      <out>
        <l name="ResponseCode" value="001"/>
        <f name="F" value=""/>
        <f name="S" value="10450433-000f2n="/>
        <f name="t" value="322601039"/>
        <f name="a" value="&C981#&R01#&r000322601039#&p230131-10475277-
000f2p="/>
        <f name="6" value="HG1udWxs"/>
      </out>
    </POSResponse>
  </Response>
</TKKPG>
```

where:

- **Operation** - Operation Type.
- **Status** - Order status. 00 means no errors.

JSON

Example of the request url:

https://libertypaytxtest.lb.ge:6444/order/1807/exec-tran

where:

- 1807 – Preauthorized Order Id.

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{"tran": {"phase": "Clearing", "amount": "0.1"}}
```

where:

- **phase** - phase type.
- **amount** - Amount . **1 equals to 1 lari.**

Example of the response body:

```
{  
  "tran": {  
    "match": {  
      "tranActionId": "230207-15013168-000fhu=",  
      "ridByPmo": "322602595"  
    }  
  }  
}
```

where:

- tranActionId - identifier of the created transaction.
- ridByPmo – internal Id.

Payment Reversal

The operation is used to reverse (cancel) the payment if the funds were put on hold in the account and the order status was set to APPROVED or PREAUTH-APPROVED.

SOAP

Example of the request url:

https://libertypaytxtest.lb.ge:6443/Exec

Example of the request body:

```
<?xml version="1.0" encoding="UTF-8"?>
<TKKPG>
  <Request>
    <Operation>Reverse</Operation>
    <Language>RU</Language>
    <Order>
      <Merchant>IA00007A</Merchant>
      <OrderID>1760</OrderID>
    </Order>
    <SessionID>154q5ykfy72d3</SessionID>
  </Request>
</TKKPG>
```

Where:

- **Operation** - Operation Type.
- **Language** - Interface language Language
- **OrderType** - Order Type. The financial transaction to be created. Specify the Purchase value to perform the purchase.
- **Merchant** - Internet shop ID.
- **OrderID** – Order Id.
- **SessionID** – Session Id.

Example of the response body:

```
<TKKPG>
  <Response>
    <Operation>Reverse</Operation>
    <Status>00</Status>
    <Order>
      <OrderID>1760</OrderID>
    </Order>
    <Reversal>
      <RespCode>001</RespCode>
    </Reversal>
  </Response>
</TKKPG>
```

where:

- **Operation** - Operation Type.
- **Status** - Order status. 00 means no errors.
- **OrderID** – Order Id.
- **RespCode** – Response Code.

JSON

Example of the request url:

<https://libertypaytxtest.lb.ge:6444/order/1803/exec-tran>

where:

- **terminalid** – Merchant Id.

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{"tran": {"voidKind": "Full", "amount": "0.1", "phase": "Single"}}
```

where:

- **voidKind** – Full – void transaction Type.
- **amount** - Amount . **1 equals to 1 lari.**
- **phase** – Phase type.

Example of the response body:

```
{
  "tran": {
    "match": {
      "tranActionId": "230207-11513433-000fge=",
      "ridByPmo": "322602545"
    }
  }
}
```

where:

- **tranActionId** - identifier of the created transaction.
- **ridByPmo** – internal Id.

Partial Reversal

The operation is used to reverse (cancel) partial payment amount.

After creating order and making payment transaction we need to get order information using advanced Get order command.

JSON

Example of the request url:

```
https://libertypaytxttest.lb.ge:6444/order/1626?password=vzfsmec0trz1
&orderDetailLevel=2&merchantDetailLevel=2&ctsDetailLevel=2&cofpDetailLevel=2&consumerDetailLevel=2&consumerTokenDetailLevel=2&tokenDetailLevel=2&tranDetailLevel=2&actionDetailLevel=2
```

Example of the response body:

```
{
.....
"trans": [
  {
    "approvalCode": "384564",
    "actionId": "230428-12095173-000hpg=",
.....
  }
}
```

From this response we need actionId.

Example of the request url:

```
https://libertypaytxttest.lb.ge:6444/order/1803/exec-tran
```

where:

- **terminalid** – Merchant Id.

Example of Authorization Header:

Authorization: Basic VGVyYWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "tran": {
    "voidKind": "Partial",
    "amount": "0.1",
    "phase": "Single",
    "links": {
      "voided": {
        "tranActionId": "230428-12095173-000hpg="
      }
    }
  }
}
```

where:

- **voidKind** – Full – void transaction Type.
- **amount** - Amount . **1 equals to 1 lari.**
- **tranActionId** – Id of the transaction
- **phase** – Phase type.

Example of the response body:

```
{
  "tran": {
    "match": {
      "tranActionId": "230207-11513433-000fge=",
      "ridByPmo": "322602545"
    }
  }
}
```

where:

- **tranActionId** - identifier of the created transaction.
- **ridByPmo** – internal Id.

Refund

Refund operation returns funds back to client from a given transaction.

SOAP

Example of the request url:

https://libertypaytxtest.lb.ge:6443/Exec

Example of the request body:

```
<?xml version="1.0" encoding="UTF-8"?>
<TKKPG>
<Request>
<Operation>Refund</Operation>
<Language>RU</Language>
<Order>
<Merchant>IA00007A</Merchant>
<OrderID>1780</OrderID>
</Order>
<SessionID>1p3363vorurpi</SessionID>
<Refund>
<Amount>1</Amount>
<Currency>981</Currency>
</Refund>
</Request>
</TKKPG>
```

Where:

- **Operation** - Operation Type.
- **Language** - Interface language Language
- **Merchant** - Internet shop ID.
- **OrderID** – Order Id.
- **SessionID** – Session Id.
- **Amount** - Transaction amount in the minimum units of the transaction currency.
- **Currency** - Transaction currency code (e.g. 981– Georgian Lari, 840 – US Dollar)

Example of the response body:

```
<TKKPG>
  <Response>
    <Operation>Refund</Operation>
    <Status>00</Status>
  </Response>
</TKKPG>
```

where:

- **Operation** - Operation Type.
- **Status** - Order status. 00 means no errors.

JSON

Example of the request url:

<https://libertypaytxtest.lb.ge:6444/order/1780/exec-tran>

where:

- 1780 – Order Id of transaction that is decided to be refunded.

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "tran": {
    "phase": "Single",
    "amount": "0.01",
    "type": "Refund"
  }
}
```

where:

- **phase** – Type of phases – use Single for the Refund operation.
- **amount** - Amount . **1 equals to 1 lari.**
- **Type** – operation type.

Example of the response body:

```
{
  "tran": {
    "approvalCode": "941228",
    "match": {
      "tranActionId": "230202-13431730-000fac=",
      "ridByPmo": "322601576"
    }
  }
}
```

where:

- **id** - identifier of the created order. It should be passed in all requests related to the order.
- **ApprovalCode** – code that is approving successful state of given transaction.
- **tranActionId** - identifier of the created transaction.
- **ridByPmo** – internal Id.

P.S. You can make any amount of refund transactions, until sum of all this refund transactions will reach original transaction amount.

Get Order Details (getOrder) Operation

To check order status we can send getOrder request:

JSON

Example of the request header:

`https://libertypaytxttest.lb.ge:6444/order/1626?password=vzfsmec0trz1`

Example of the response body:

```
{
  "order": {
    "id": 1783,
    "typeRid": "TWPG-Purchase",
    "status": "FullyPaid",
    "lastStatusLogin": "IA00007A",
    "amount": 0.1,
    "currency": "GEL",
    "createTime": "2023-02-03 13:27:26",
    "type": {
      "allowVoid": false,
      "title": "TWPG-Purchase"
    }
  }
}
```

Recurrent Payment MIT

(Merchant Initiated Transaction)

Recurrent payment allows client to save payment method and use it automatically to pay transactions. In first Create Order – amount can be set to zero.

JSON

First Create Order.

Example of the request header:

<https://libertypaytxttest.lb.ge:6444/order?terminalid=41>

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Step1:

Example of the request body:

```
{
  "order":{
    "typeRid":"TWPG-Purchase",
    "language":"ru",
    "amount":"0",
    "currency":"GEL",
    "description":"test order",
    "hppCofCapturePurposes":[
      "Cit",
      "Recurring",
      "UnspecifiedMit"
    ],
    "custAttrs": [
      {
        "rid": "@TWPG@CardHolderLogin",
        "valAsStr": "Login121212"
      },
      {
        "rid":"Custom-needregcard",
        "valAsStr":"on"
      }
    ]
  }
}
```

```
    },
    {"rid": "@TWPG@ApproveURL", "valAsStr": "https://libertypay.ge/Shop/out.jsp"},
    {"rid": "@TWPG@DeclineURL", "valAsStr": "https://libertypay.ge/Shop/out.jsp"},
    {"rid": "@TWPG@CancelURL", "valAsStr": "https://libertypay.ge/Shop/out.jsp"}
  ],
  "consumerDevice": {
    "browser": {
      "deviceId": "6996B7A6-24D9-4F5A-BE09-99AD420AAFFF",
      "acceptHeader": "application/json,application/jose;charset=utf-8",
      "ip": "127.0.0.1",
      "tzOffset": "-300",
      "platform": "Mozilla",
      "jsEnabled": true,
      "colorDepth": 32,
      "screenW": 1024,
      "screenH": 800,
      "userAgent": "chorme",
      "language": "Ru-ru",
      "javaEnabled": true
    }
  }
}
```

Example of the response body:

```
{
  "order": {
    "id": 1939,
    "status": "Preparing",
    "password": "1ekphcnoq2sm5"
  }
}
```

Redirect client to TWPGA payment page to make first payment.

<https://libertypaytxtest.lb.ge/twpga?id=1939&password=1ekphcnoq2sm5>

After client finishes payment we need to Get Order Details. Use Id and password from previous response.

Step2:

Example of the request header:

<https://libertypaytxttest.lb.ge:6444/order/1939?merchantId=41&password=1ekphcnoq2sm5&orderDetailLevel=2&merchantDetailLevel=2&ctsDetailLevel=2&cofpDetailLevel=2&consumerDetailLevel=2&consumerTokenDetailLevel=2&tokenDetailLevel=2&tranDetailLevel=2&actionDetailLevel=2>

Example of the response body:

```
{  
  .....  
  "storedTokens": [  
    {  
      "id": 1365  
    }  
  ],  
  .....
```

From here we need only storedToken ID.

Next Step is to make another Create Order, this time without forcing client to enter payment data.

Step3:

Example of the request header:

<https://libertypaytxttest.lb.ge:6444/order?terminalid=41>

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{  
  "order":{  
    "typeRid":"MIT",  
    "language":"ru",  
    "amount":"0.1",  
    "currency":"GEL",  
    "description":"test order",  
    "hppCofCapturePurposes":[  
      "Cit",  
      "Recurring",  
      "UnspecifiedMit"  
    ]  
  }  
}
```



```
],
"custAttrs": [
  {
    "rid": "@TWPG@CardHolderLogin",
    "valAsStr": "Login121212"
  },
  {
    "rid": "Custom-needregcard",
    "valAsStr": "on"
  },
  {"rid": "@TWPG@ApproveURL", "valAsStr": "https://libertypay.ge/Shop/out.jsp"},
  {"rid": "@TWPG@DeclineURL", "valAsStr": "https://libertypay.ge/Shop/out.jsp"},
  {"rid": "@TWPG@CancelURL", "valAsStr": "https://libertypay.ge/Shop/out.jsp"}
],
"consumerDevice": {
  "browser": {
    "deviceId": "6996B7A6-24D9-4F5A-BE09-99AD420AAFFF",
    "acceptHeader": "application/json,application/jose;charset=utf-8",
    "ip": "127.0.0.1",
    "tzOffset": "-300",
    "platform": "Mozilla",
    "jsEnabled": true,
    "colorDepth": 32,
    "screenW": 1024,
    "screenH": 800,
    "userAgent": "chorme",
    "language": "Ru-ru",
    "javaEnabled": true
  }
}
}
```

Example of the response body:

```
{
  "order": {
    "id": 1941,
    "status": "Preparing",
    "password": "1rh0yo4se89rn"
  }
}
```

Now we need to set Source Token for that order.

Step4:

Example of the request header:

<https://libertypaytxttest.lb.ge:6444/order/1941/set-src-token?password=1rh0yo4se89rn>

Example of the request body:

```
{ "order": {
  "initiationEnvKind": "Server"
},
  "token": {
    "storedId": 1365
  }
}
```

Example of the response body:

```
{
  "order": {
    "status": "Preparing",
    "cvv2AuthStatus": "NotRequired",
    "tdsV1AuthStatus": "IneligibleOrder",
    "tdsV2AuthStatus": "IneligibleOrder",
    "otpAutStatus": "IneligibleOrder",
    "srcToken": {
      "id": 1367,
      "paymentMethod": "Card",
      "role": "Src",
      "status": "Active",
      "regTime": "2023-03-17 16:09:02",
      "displayName": "412571*****9128",
      "card": {
        "expiration": "0326",
        "brand": "Visa",
        "currency": "GEL"
      }
    }
  }
}
```

Step5:

Last step is to execute our new transaction.

Example of the request header:

<https://libertypaytxttest.lb.ge:6444/order/1941/exec-tran?password=1rh0yo4se89rn>

Example of the request body:

```
{
  "tran": {
    "phase": "Single",
    "amount": "0.1",
    "conditions": {
      "cofUsage": "Recurring"
    }
  }
}
```

Example of the response body:

```
{
  "tran": {
    "approvalCode": "378493",
    "match": {
      "tranActionId": "230317-11093925-000gh8=",
      "ridByPmo": "322608998"
    }
  }
}
```

In case of order decline merchant may get such type of error:

```
{"errorCode":"PmoDecline","errorDescription":"Transaction declined by PMO: Auth Response: 6 - WITHOUT PIN EXCLUDECOUNT LIMIT(F)"}
```

P.S. In case of zero amount status of transaction always will remain as “Preparing”, instead of “FullyPaid”.

EMV 3-D Secure Authentication without Redirecting to TXPG payment Page

Making transaction without visiting TXPG payment page

SOAP

Creating Purchase Order

Example of the request url:

<https://libertypaytxtest.lb.ge:6443/Exec>

Example of the request body:

```
<?xml version="1.0" encoding="UTF-8"?>
  <TKKPG>
    <Request>
      <Operation>CreateOrder</Operation>
      <Language>RU</Language>
      <Order>
        <OrderType>Purchase</OrderType>
        <Merchant>IA00007A</Merchant>
        <Amount>1</Amount>
        <Currency>981</Currency>
        <Description>xxxxxxx</Description>
        <ApproveURL>https://libertypaytxtest.lb.ge/</ApproveURL>
        <CancelURL>https://libertypaytxtest.lb.ge/</CancelURL>
        <DeclineURL>https://libertypaytxtest.lb.ge/</DeclineURL>
        <AddParams>
          </AddParams>
        </Order>
      </Request>
    </TKKPG>
```

Where:

- **Operation** - Operation Type.
- **Language** - Interface language Language
- **OrderType** - Order Type. The financial transaction to be created. Specify the Purchase value to perform the purchase.
- **Merchant** - Internet shop ID.
- **Amount** - Transaction amount in the minimum units of the transaction currency.
- **Currency** - Transaction currency code (e.g. 981– Georgian Lari, 840 – US Dollar)
- **Description** - Order text description. It is displayed to the customer during payment.

- **ApproveURL** - URL to forward the customer to if the operation is approved. If the parameter is not defined, its value will be taken from the parameter of the same name specified in the merchant settings.
- **CancelURL** - URL to forward the customer to if the operation is cancelled (by clicking the Cancel button).
- **DeclineURL** - URL to forward the customer to if the payment operation is declined.

Example of the response body:

```
<TKKPG>
  <Response>
    <Operation>CreateOrder</Operation>
    <Status>00</Status>
    <Order>
      <OrderID> 1809</OrderID>
      <SessionID> 6lv5g8sv063y</SessionID>
      <URL> https://libertypaytxtest.lb.ge/twpga</URL>
    </Order>
  </Response>
</TKKPG>
```

where:

- **Operation** - Operation Type.
- **Status** - Order status. 00 means no errors.
- **OrderID** – Order Id
- **SessionID** – Session Id or password.
- **URL** - payment page url if not provided use one written below.

Next step Process3DSAuth – First**Example of the request url:**

https://libertypaytxttest.lb.ge:6443/Exec

Example of the request body:

```
<?xml version="1.0" encoding="UTF-8"?>
  <TKKPG>
    <Request>
      <Operation>Process3DSAuth</Operation>
      <Step>FIRST</Step>
      <Language>RU</Language>
      <Order>
        <OrderID>1809</OrderID>
        <Merchant>IA00007A</Merchant>
      </Order>
      <SessionID>6lv5g8sv063y</SessionID>
      <PAN>4125710020009128</PAN>
      <PAN2>4125710020009128</PAN2>
      <ExpDate>0326</ExpDate>
      <CardHolderName>Jhon Doe</CardHolderName>
      <CVV2>582</CVV2>
      <AReqDetails>
        <deviceChannel></deviceChannel>
      </AReqDetails>
    </Request>
  </TKKPG>
```

Where:

- **Operation** - Operation Type.
- **Step** – Step Type.
- **Language** - Interface language Language
- **OrderID**- Order Id.
- **Merchant** - Internet shop ID.
- **SessionID** – Session Id or password.
- **PAN** – Card PAN.
- **ExpDate** – Expiration Date
- **CardHolderName** – Name on the Card.
- **CVV2**- CVV.

Example of the response body:

```
<TKKPG>
  <Response>
    <Operation>Process3DSAuth</Operation>
    <Status>00</Status>
    <NextStep>AREQ</NextStep>
  </Response>
</TKKPG>
```

where:

- **Operation** - Operation Type.
- **Status** - Order status. 00 means no errors.
- **NextStep** – Next Step Name.

Next step Process3DSAuth – AREQ**Example of the request url:**

https://libertypaytxtest.lb.ge:6443/Exec

Example of the request body:

```
<?xml version="1.0" encoding="UTF-8"?>
<TKKPG>
<Request>
<Operation>Process3DSAuth</Operation>
<Step>AREQ</Step>
<Order>
<OrderID>1809</OrderID>
<Merchant>IA00007A</Merchant>
</Order>
<SessionID>6lv5g8sv063y</SessionID>
<PAN>4125710020009128</PAN>
<PAN2>4125710020009128</PAN2>
<ExpDate>0326</ExpDate>
<CardHolderName>Jhon Doe</CardHolderName>
<CVV2>582</CVV2>
<AReqDetails>
<notificationURL>http://your_URL</notificationURL>
<messageCategory>01</messageCategory>
<browserAcceptHeader>application/json,application/jose;charset=utf-8</browserAcceptHeader>
```

```
<browserJavaEnabled>false</browserJavaEnabled>
<browserJavascriptEnabled>true</browserJavascriptEnabled>
<browserLanguage>en-US</browserLanguage>
<browserColorDepth>24</browserColorDepth>
<browserScreenHeight>768</browserScreenHeight>
<browserScreenWidth>1024</browserScreenWidth>
<browserTZ>-240</browserTZ>
<browserUserAgent>Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/91.0.4472.101 Safari/537.36 Edg/91.0.864.48</browserUserAgent>
<threeRIInd>05</threeRIInd>
<threeDSCompInd>U</threeDSCompInd>
<threeDSRequestorAuthenticationInd>01</threeDSRequestorAuthenticationInd>
<threeDSRequestorChallengeInd>03</threeDSRequestorChallengeInd>
<transType>01</transType>
<deviceChannel>02</deviceChannel>
</AReqDetails>
</Request>
</TKKPG>
```

Where:

- **Operation** - Operation Type.
- **Step** – Step Type.
- **OrderID**- Order Id.
- **Merchant** - Internet shop ID.
- **SessionID** – Session Id or password.
- **PAN** – Card PAN.
- **ExpDate** – Expiration Date
- **CardHolderName** – Name on the Card.
- **CVV2**- CVV.
- **AreqDetails** – Additional Details including browser Data and 3DS settings...

Example of the response body:

```
<TKKPG>
  <Response>
    <Operation>Process3DSAuth</Operation>
    <Status>00</Status>
    <NextStep>CRES</NextStep>
    <Refinement>
      <RefinementType>CREQ</RefinementType>
      <AcsURL>https://acstest.libertybank.ge</AcsURL>

      <CReq>eyJtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmVyc2lvbiI6IjluMi4wIiwiaWwiYWVzVHJhbnNJRCI6ImE2ZT
      AwZTkwlTRjODUtNGZhNS1hYzNkLWU5NWQ4YjdmZjU1ZSIsInRocmVIRFNTZXJ2ZXJUcmFuc0EiEiNTc2ODUz
      MzEtNjA5NS00ZDE3LTg2MzctNzU2YjJjMDhjY2QzIn0=</CReq>
    </Refinement>
    <Ares>{"threeDSServerTransID":"57685331-6095-4d17-8637-756b2c08ccd3","acsTransID":"a6e00e90-
    4c85-4fa5-ac3d-e95d8b7ff55e","dsTransID":"46ba2166-2885-48f6-9c32-
    e796a3bae717","messageType":"Ares","messageVersion":"2.2.0","messageExtension":{"name":"MesExt1"
    ,"id":"ID1","criticalityIndicator":false,"data":{"valueOne":"value"}}, "dsReferenceNumber":"DSRefNum1","a
    csReferenceNumber":"3DS_LOA_ACS_COPL_020200_00327","acsChallengeMandated":"N","acsOperatorID"
    ::"10055116","acsURL":"https://acstest.libertybank.ge","authenticationType":"02","transStatus":"C"}</Ares
    >
  </Response>
</TKKPG>
```

where:

- **Operation** - Operation Type.
- **Status** - Order status. 00 means no errors.
- **NextStep** – Next Step Name.
- **AcsURL** – URL to 3DS page.
- **CReq** – Creq hashed string required for 3DS.
- **Ares** – additional Ares data...

Next step - Merchants opens Iframe in clients browser and sends POST request with creq inside form

Form example:

```
<form id="formCReq" name="formCReq" action="https://acstest.libertybank.ge" method="post">
  <input type="hidden" id="creq" name="creq" value="
  eyJtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmVyc2lvbiI6IjluMi4wIiwiaWwiYWVzVHJhbnNJRCI6ImE2ZT
  AwZTkwlTRjODUtNGZhNS1hYzNkLWU5NWQ4YjdmZjU1ZSIsInRocmVIRFNTZXJ2ZXJUcmFuc0EiEiNTc2ODUz
  MzEtNjA5NS00ZDE3LTg2MzctNzU2YjJjMDhjY2QzIn0=" />
</form>
```

After that ACS will open 3d authorization page inside Iframe.
Client enters OTP, then ACS sends POST response to the notificationURL which we sent in Areq step: Inside that POST response there will be cres which you need for further steps.

Example:

```
cres = eyJ0aHJIZURTU2VydmVyVHJhbnNJRCI6ImNiYTkyYTRiLWRmOTQtNDhkMCO4NGY3LTImYjE2ZTE3MTVhNCiSlmFjc1RyYW5zSUQiOiJyWYxNTAxMi1iMjI2LTRiZDQtYTIyZC0wMzcwMjA3Y2lyMWIiLCJtZXNzYWdlVHlwZSI6IksNSZXMlLCJtZXNzYWdlVmVyc2lvbil6IjluMS4wliwidHJhbnNTdGF0dXMiOiJZIn0
```

cres must be captured by Merchant and transferred to TXPG as a parameter.

Example of the request url:

```
https://libertypaytstest.lb.ge:6443/Exec
```

Example Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<TKKPG>
<Request>
<Operation>Process3DSAuth</Operation>
<Step>CRES</Step>
<Order>
<OrderID>15425181</OrderID>
<Merchant> IA00007A</Merchant>
</Order>
<SessionID>EBDCFB9EA0524A688A23779792A644F7</SessionID>
<PAN>4125706004391888</PAN>
<PAN2>4125706004391888</PAN2>
<ExpDate>2211</ExpDate>
<CardHolderName>Jhon Doe</CardHolderName>
<CVV2>814</CVV2>
<Cres>eyJ0aHJIZURTU2VydmVyVHJhbnNJRCI6ImNiYTkyYTRiLWRmOTQtNDhkMCO4NGY3LTImYjE2ZTE3MTVhNCiSlmFjc1RyYW5zSUQiOiJyWYxNTAxMi1iMjI2LTRiZDQtYTIyZC0wMzcwMjA3Y2lyMWIiLCJtZXNzYWdlVHlwZSI6IksNSZXMlLCJtZXNzYWdlVmVyc2lvbil6IjluMS4wliwidHJhbnNTdGF0dXMiOiJZIn0</Cres>
</Request>
</TKKPG>
```

TXPG verifies the signature of the received PARES and the 3-D Secure authentication result. If the order payment is allowed according to the TXPG rules, TXPG sends the transaction of the respective type to the authorization system. Depending on the authorization system response, TXPG assigns the APPROVED/ DECLINED status to the order.

TXPG sends the operation execution result and XMLOut in the response to the ProcessPARES operation. TXPG response format:

Example Response:

```
<TKKPG><Response><Operation>Process3DSAuth</Operation><Result><XMLOut><Message
date="21/09/2021
11:54:58"><Language>GE</Language><ResponseCode>001</ResponseCode><Version>1.0</Version><Curr
ency>981</Currency><PAN>412570XXXXX1888</PAN><SessionId>EBDCFB9EA0524A688A23779792A644F7</S
essionId><TotalAmountScr>0.01</TotalAmountScr><CurrencyISOAlpha>GEL</CurrencyISOAlpha><TranD
ateTime>21/09/2021
11:54:58</TranDateTime><ThreeDSVerification>Y</ThreeDSVerification><ApprovalCodeScr>703528</
ApprovalCodeScr><AcqFeeScr>0.00</AcqFeeScr><CurrencyScr>Lari</CurrencyScr><AuthorizationResp
onseCode>01</AuthorizationResponseCode><OrderDescription>Descr 123124
</OrderDescription><PurchaseAmountScr>0.01</PurchaseAmountScr><ThreeDSVerificaion>Y</ThreeDS
Verificaion><TransactionType>Purchase</TransactionType><MerchantTranID>4c246316-811c-4571-
a904-
0e253d3e3a19</MerchantTranID><OrderStatusScr>დადასტურებულია</OrderStatusScr><TotalAmount>1
</TotalAmount><RezultOperation>შედეგი</RezultOperation><ThreeDSVars><PrefixParam><PurchaseIf
NotEnrolled>true</PurchaseIfNotEnrolled><TypeCard>3DS1_TERA</TypeCard></PrefixParam><AnswerV
ars><eci>05</eci><ThreeDSVersion>2</ThreeDSVersion><CalculatedCAVV>4c246316811c4571a9040e253
d3e3a19000000000020108030096000000008660096000000000</CalculatedCAVV><ThreeDSVerification
>Y</ThreeDSVerification><ThreeDSVerificaion>Y</ThreeDSVerificaion><xid>4c246316-811c-4571-
a904-
0e253d3e3a19</xid><CAVV>AAIBCAMAlgAAAAAIzgCWAAAAAAA=</CAVV><HexCAVV>4c246316811c4571a9040e25
3d3e3a190000000000020108030096000000008660096000000000</HexCAVV></AnswerVars></ThreeDSVars
><ShopName>Demo Site
</ShopName><SessionID>EBDCFB9EA0524A688A23779792A644F7</SessionID><OrderID>15425181</OrderID
><ApprovalCode>703528
A</ApprovalCode><FeeScr>0.00</FeeScr><PAN2>412570XXXXX1888</PAN2><OrderStatus>APPROVED</Ord
erStatus><Brand>3DS1_TERA</Brand><RRN>00008660097</RRN><PurchaseAmount>1</PurchaseAmount><T
ranId>8660097</TranId></Message></XMLOut></Result><Status>00</Status><AuthenticationValue>AA
IBCAMAlgAAAAAIzgCWAAAAAAA=</AuthenticationValue><ECI>05</ECI></Response></TKKPG>
```

Merchant informs the customer of the operation execution result.

JSON

Creating Purchase Order

Example of the request url:

https://libertypaytxtest.lb.ge:6444/order?terminalid=41
where:

- **terminalid** – Merchant Id.

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "order":{
    "typeRid":"TWPG-Purchase",
    "language":"ru",
    "amount":"0.1",
    "currency":"GEL",
    "description":"test order",
    {"rid":"@TWPG@ApproveURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"},
    {"rid":"@TWPG@DeclineURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"},
    {"rid":"@TWPG@CancelURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"}
  }
  "consumerDevice":{
    "browser":{
      "deviceId":"6996B7A6-24D9-4F5A-BE09-99AD420AAFFF",
      "acceptHeader":"application/json,application/jose;charset=utf-8",
      "ip":"127.0.0.1",
      "tzOffset":"-300",
      "platform":"Mozilla",
      "jsEnabled":true,
      "colorDepth":32,
      "screenW":1024,
      "screenH":800,
      "userAgent":"chrome",
      "language":"Ru-ru",
      "javaEnabled":true
    }
  }
}
```

where:

- **typeRid** - Order type.
- **Language** – Language.
- **amount** - Amount . **1 equals to 1 lari.**
- **currency** – Currency.
- **description** - Additional info
- **consumerDevice** – Client Device parameters

Example of the response body:

```
{  
  "order": {  
    "id": 1810,  
    "status": "Preparing",  
    "password": "1jtdfr528wm3p",  
    "url": "https://libertypaytxtest.lb.ge/twpga"  
  }  
}
```

where:

- **id** - identifier of the created order. It should be passed in all requests related to the order.
- **status** – status of the order.
- **password** - password of the order.
- **url** - payment page url if not provided use <https://libertypaytxtest.lb.ge/twpga>.

Creating Source Token

Example of the request url:

https://libertypaytxttest.lb.ge:6444/order/1810/set-src-token?password=1jtdfr528wm3p

where:

- **order** – Order Id.
- **password** - Session Id or password.

Example of Authorization Header:

Authorization: Basic VGVyYWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "token": {
    "card": {
      "panBlock": {
        "data": "4125710020009128"
      },
      "expiration": "0326",
      "entryMode": "ECommerce"
    }
  }
}
```

where:

- **data** – Card PAN.
- **expiration** – Expiration Data.
- **entryMode** - EntryMode Type.

Example of the response body:

```
{
  "order": {
    "status": "Preparing",
    "cvv2AuthStatus": "Required",
    "tdsV1AuthStatus": "IneligibleOrder",
    "tdsV2AuthStatus": "Required",
    "otpAutStatus": "IneligibleOrder",
    "srcToken": {
      "id": 1248,
      "paymentMethod": "Card",
      "role": "Src",
      "status": "Active",
      "regTime": "2023-02-08 15:03:50",
    }
  }
}
```

```
"displayName": "412571*****9128",
"card": {
  "expiration": "0326",
  "brand": "Visa",
  "currency": "GEL"
}
},
"tdsData": {
  "aReq": {
    "deviceChannel": "02",
    "messageVersion": "2.2.0",
    "messageCategory": "01",
    "messageType": "AReq",
    "threeDSComplnd": "U",
    "threeDSRequestorAuthenticationInd": "01",
    "threeDSRequestorID": "IA00007A",
    "threeDSRequestorName": "IA00007A",
    "threeDSRequestorURL": "https://libertybank.ge/",
    "threeDSServerRefNumber": "DSRefNum1",
    "threeDSServerURL": "http://192.168.20.16:1111",
    "threeDSServerTransID": "8641f571-50ed-46f3-8dd4-54977f9d3f02",
    "threeDSServerOperatorID": "DSRefNum1",
    "browserAcceptHeader": "application/json,application/jose;charset=utf-8",
    "browserIP": "127.0.0.1",
    "browserJavaEnabled": true,
    "browserJavascriptEnabled": true,
    "browserLanguage": "Ru-ru",
    "browserColorDepth": "32",
    "browserScreenHeight": "800",
    "browserScreenWidth": "1024",
    "browserTZ": "-300",
    "browserUserAgent": "chorme",
    "cardExpiryDate": "2603",
    "acquirerBIN": "LB_11.2",
    "acquirerMerchantID": "IA00007A",
    "merchantCountryCode": "268",
    "merchantName": "IA00007A",
    "mcc": "5065",
    "purchaseAmount": "10",
    "purchaseCurrency": "981",
    "purchaseExponent": "2",
    "purchaseDate": "20230208095836",
    "transType": "01"
  }
}
}
```

Next Merchant makes **AREQ Request**

Example of the request url:

`https://libertypaytxtest.lb.ge:6444/order/1810/tds-areq?password=1jtdfr528wm3p`

where:

- **order** – Order Id.
- **password** - Session Id or password.
- **threeDSserverTransID** – Id that came with create source token response

Example of Authorization Header:

Authorization: Basic VGVyYWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "aReq": {
    "acquirerBIN": "400551",
    "acquirerMerchantID": "IA00007A",
    "browserAcceptHeader": "test",
    "browserColorDepth": "1",
    "browserJavaEnabled": "false",
    "browserJavascriptEnabled": "false",
    "browserLanguage": "ru_RU",
    "browserScreenHeight": "10",
    "browserScreenWidth": "10",
    "browserTZ": "3000",
    "browserUserAgent": "test",
    "cardExpiryDate": "2603",
    "deviceChannel": "02",
    "mcc": 3000,
    "merchantCountryCode": "268",
    "merchantName": "IA00007A",
    "messageCategory": "01",
    "messageType": "AReq",
    "messageVersion": "2.2.0",
    "purchaseAmount": "10",
    "purchaseCurrency": "981",
    "purchaseDate": "20210311043956",
    "purchaseExponent": "2",
    "threeDSCompInd": "U",
    "threeDSRequestorAuthenticationInd": "01",
    "threeDSRequestorChallengeInd": "04",
    "threeDSRequestorID": "RBC62866",
    "threeDSRequestorName": "RCB test requestor",
```



```
"threeDSRequestorURL": "http://ya.ru",
"threeDSOperatorID": "10067463",
"threeDSRefNumber": "3DS_LOA_SER_COPL_020200_00313",
"threeDSTransID": "8641f571-50ed-46f3-8dd4-54977f9d3f02",
"threeDSURL": "http://10.7.3.143:3000",
"transType": "01",
"notificationURL": "http://test.com/return"
}
}
```

Example of the response body:

```
{
  "aRes": {
    "acsReferenceNumber": "3DS_LOA_ACS_COPL_020200_00327",
    "acsChallengeMandated": "N",
    "acsOperatorID": "10055116",
    "acsURL": "https://acstest.libertybank.ge",
    "authenticationType": "02",
    "dsReferenceNumber": "DSRefNum1",
    "transStatus": "C",
    "threeDSRefNumber": "8641f571-50ed-46f3-8dd4-54977f9d3f02",
    "acsTransID": "1ee8e5bc-432e-4406-b0cf-2d165ee6c3a6",
    "dsTransID": "01a27b57-0e34-4c4d-a5e3-706a715f75cf",
    "messageType": "ARes",
    "messageVersion": "2.2.0",
    "messageExtension": [
      {
        "name": "MesExt1",
        "id": "ID1",
        "criticalityIndicator": false,
        "data": {
          "valueOne": "value"
        }
      }
    ]
  },
  "challengeRequired": true,
  "cReq":
  "eyJhY3NUcmFuc0EiIjoiMmVlOGU1YmMtNDMyZS00NDA2LWlwY2YtMmQxNjVlZTZjM2E2IiwidGhyZWVEU1NlcnZlclRyYW5zSUQiOiIiYmM1OTc2Ny05ZmRmLTQxOWQtOGU0MS1jNzk3YWlzZGE3ZmUiLCJjaGFsbGVuZ2VXaW5kb3dTaXplIjoiMmDUiLCJtZXNzYWdlVHlwZSI6IkNSZXElLCJtZXNzYWdlVmVyc2lvbiI6IjluMi4wLn0",
  "order": {
    "tdsV2AuthStatus": "Required",
    "status": "Preparing"
  }
}
```


Example of the request body:

```
{
  "tran":{
    "phase":"Single",
    "amount": "0.1",
    "authentication":{
      "cvv2Block":
{      "data":"582"      }
    }
  }
}
```

where:

- **data** – CVV.

Example of the response body:

```
{
  "tran": {
    "approvalCode": "384449",
    "match": {
      "tranActionId": "230424-10054137-000hlp=",
      "ridByPmo": "322615867"
    }
  }
}
```

Recurrent Payment CIT

(Client Initiated Transaction) with 3DS authentication

First step - Create Order – amount can be set to zero.

JSON

First Create Order.

Example of the request header:

<https://libertypaytxtest.lb.ge:6444/order?terminalid=41>

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "order":{
    "typeRid":"CIT-3DS",
    "language":"ru",
    "amount":"0",
    "currency":"GEL",
    "description":"test order",
    "hppCofCapturePurposes":[
      "Cit"
    ],
    "custAttrs": [
      {
        "rid": "@TWPG@CardHolderLogin",
        "valAsStr": "Login121212"
      },
      {
        "rid":"Custom-needregcard",
        "valAsStr":"on"
      },
      {"rid":"@TWPG@ApproveURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"},
      {"rid":"@TWPG@DeclineURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"},
      {"rid":"@TWPG@CancelURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"}
    ]
  }
}
```

```
  ],
  "consumerDevice":{
    "browser":{
      "deviceId":"6996B7A6-24D9-4F5A-BE09-99AD420AAFFF",
      "acceptHeader":"application/json,application/jose;charset=utf-8",
      "ip":"127.0.0.1",
      "tzOffset":"-300",
      "platform":"Mozilla",
      "jsEnabled":true,
      "colorDepth":32,
      "screenW":1024,
      "screenH":800,
      "userAgent":"chrome",
      "language":"Ru-ru",
      "javaEnabled":true
    }
  }
}
```

Example of the response body:

```
{
  "order": {
    "id": 4320,
    "password": "k9tg0ggoq7tk",
    "status": "Preparing",
    "cvv2AuthStatus": "Required"
  }
}
```

Redirect client to TWPGA payment page to make first payment.

<https://libertypaytxtest.lb.ge/twpga?id=4320&password=k9tg0ggoq7tk>

After client finishes payment we need to Get Order Details. Use Id and password from previous response.

Example of the request header:

<https://libertypaytxtest.lb.ge:6444/order/4320?merchantId=122&password=k9tg0ggoq7tk&orderDetailLevel=2&merchantDetailLevel=2&ctsDetailLevel=2&cofpDetailLevel=2&consumerDetailLevel=2&consumerTokenDetailLevel=2&tokenDetailLevel=2&tranDetailLevel=2&actionDetailLevel=2>

Example of the response body:

```
{
  .....
  "storedTokens": [
    {
      "id": 2973
    }
  ],
  .....
}
```

From here we need only storedToken ID.

Next Step is to make another Create Order, this time without forcing client to enter payment data.

Example of the request header:

<https://libertypaytxttest.lb.ge:6444/order?terminalid=122>

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "order":{
    "typeRid":"CIT-3DS",
    "language":"ru",
    "amount":"0.2",
    "currency":"GEL",
    "description":"test order",
    "consumerDevice":{
      "browser":{
        "deviceId":"6996B7A6-24D9-4F5A-BE09-99AD420AAFFF",
        "acceptHeader":"application/json,application/jose;charset=utf-8",
        "ip":"127.0.0.1",
        "tzOffset":"-300",
        "platform":"Mozilla",
        "jsEnabled":true,
        "colorDepth":32,
        "screenW":1024,
        "screenH":800,
        "userAgent":"chorme",
        "language":"Ru-ru",
        "javaEnabled":true
      }
    }
  }
}
```

Example of the response body:

```
{
  "order": {
    "id": 4323,
    "password": "1dzhwawu49tsx",
    "status": "Preparing",
    "cvv2AuthStatus": "Required"
  }
}
```

Now we need to set Source Token for that order.

Example of the request header:

<https://libertypaytxttest.lb.ge:6444/order/4323/set-src-token?password=1dzhwawu49tsx>

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "order": {
    "initiationEnvKind": "Browser"
  },
  "token": {
    "storedId": 2973
  }
}
```

Example of the response body:

```
{
  "order": {
    "status": "Preparing",
    "cvv2AuthStatus": "Required",
    "tdsV1AuthStatus": "IneligibleOrder",
    "tdsV2AuthStatus": "Required",
    "otpAutStatus": "IneligibleOrder",
    "srcToken": {
      "id": 2976,
      "paymentMethod": "Card",
      "role": "Src",
      "status": "Active",
      "regTime": "2023-05-19 15:08:47",
      "displayName": "412571*****9128",
      "card": {
        "expiration": "0326",
        "brand": "Visa",
        "currency": "GEL",
        "country": "GEO"
      }
    }
  },
  "tdsData": {
    "aReq": {
      "threeDSSTransID": "90f05392-c3d2-4ea8-a4df-4236efa31a9c",
      "messageType": "AReq",
      "messageVersion": "2.2.0",
      "threeDSCompInd": "U",
      "threeDSRequestorAuthenticationInd": "01",
      "threeDSRequestorDecMaxTime": "600",
      "threeDSRequestorDecReqInd": "Y",
      "threeDSRequestorID": "IA00GPI",
      "threeDSRequestorName": "GPI ONLY RECCURING",
      "threeDSRequestorPriorAuthenticationInfo": {
        "threeDSReqPriorAuthData": "dd29cc03-2349-4502-971e-7b669ffa08b3",
        "threeDSReqPriorAuthMethod": "02",
        "threeDSReqPriorAuthTimestamp": "202305191053",
        "threeDSReqPriorRef": "1ffb6d3f-7d38-4c1d-aff3-06ebf22ea739"
      },
      "threeDSRequestorURL": "https://libertybank.ge/",
      "threeDSRequestorOperatorID": "DSRefNum",
    }
  }
}
```



```
"threeDSServerRefNumber": "DSRefNum",
"threeDSServerURL": "https://3dstestsrv.libertypay.ge:9744",
"acquirerBIN": "LB_Terminal",
"acquirerMerchantID": "IA00GPI",
"browserAcceptHeader": "application/json,application/jose;charset=utf-8",
"browserIP": "127.0.0.1",
"browserJavaEnabled": true,
"browserJavascriptEnabled": true,
"browserLanguage": "Ru-ru",
"browserColorDepth": "32",
"browserScreenHeight": "800",
"browserScreenWidth": "1024",
"browserTZ": "-300",
"browserUserAgent": "chorme",
"cardExpiryDate": "2603",
"deviceChannel": "02",
"mcc": "5814",
"merchantCountryCode": "268",
"merchantName": "GPI ONLY RECCURING",
"messageCategory": "01",
"purchaseAmount": "20",
"purchaseCurrency": "981",
"purchaseExponent": "2",
"purchaseDate": "20230519110837",
"transType": "01"
}
}
}
```

Next step is Authentication request for 3DS where we use areq response part from previous request response.

Example of the request header:

<https://libertypaytxttest.lb.ge:6444/order/4323/exec-tran?password=1dzhwawu49tsx>

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "aReq": {
    "threeDSServerTransID": "90f05392-c3d2-4ea8-a4df-4236efa31a9c",
    "messageType": "AReq",
    "messageVersion": "2.2.0",
    "threeDSCompInd": "U",
    "threeDSRequestorAuthenticationInd": "01",
    "threeDSRequestorDecMaxTime": "600",
    "threeDSRequestorDecReqInd": "Y",
    "threeDSRequestorID": "IA00GPI",
    "threeDSRequestorName": "GPI ONLY RECCURING",
    "threeDSRequestorPriorAuthenticationInfo": {
      "threeDSReqPriorAuthData": "dd29cc03-2349-4502-971e-7b669ffa08b3",
      "threeDSReqPriorAuthMethod": "02",
      "threeDSReqPriorAuthTimestamp": "202305191053",
      "threeDSReqPriorRef": "1ffb6d3f-7d38-4c1d-aff3-06ebf22ea739"
    },
    "threeDSRequestorURL": "https://libertybank.ge/",
    "threeDSServerOperatorID": "DSRefNum",
    "threeDSServerRefNumber": "DSRefNum",
    "threeDSServerURL": "https://3dstestsrv.libertypay.ge:9744",
    "acquirerBIN": "LB_Terminal",
    "acquirerMerchantID": "IA00GPI",
    "browserAcceptHeader": "application/json,application/jose;charset=utf-8",
    "browserIP": "127.0.0.1",
    "browserJavaEnabled": true,
    "browserJavascriptEnabled": true,
    "browserLanguage": "Ru-ru",
    "browserColorDepth": "32",
    "browserScreenHeight": "800",
    "browserScreenWidth": "1024",
    "browserTZ": "-300",
    "browserUserAgent": "chorme",
    "cardExpiryDate": "2603",
    "deviceChannel": "02",
    "mcc": "5814",
    "merchantCountryCode": "268",
    "merchantName": "GPI ONLY RECCURING",
    "messageCategory": "01",
    "purchaseAmount": "20",
    "purchaseCurrency": "981",
```

```
"purchaseExponent": "2",
"purchaseDate": "20230519110837",
"transType": "01",
"browserAcceptHeader": "application/json,application/jose;charset=utf-8",
"browserJavaEnabled": false,
"browserJavascriptEnabled": true,
"browserLanguage": "en-US",
"browserColorDepth": "24",
"browserScreenHeight": "949",
"browserScreenWidth": "1922",
"browserTZ": "-240",
"browserUserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36",
"notificationURL": "http://test.com/return"
}
}
```

Example of the response body:

```
{
  "aRes": {
    "threeDSServerTransID": "90f05392-c3d2-4ea8-a4df-4236efa31a9c",
    "dsReferenceNumber": "DSRefNum",
    "dsTransID": "72d2d64e-d94a-48a6-9a82-47e42cc3e4af",
    "messageExtension": [
      {
        "name": "MesExt1",
        "id": "ID1",
        "criticalityIndicator": false,
        "data": {
          "valueOne": "value"
        }
      }
    ],
    "messageType": "ARes",
    "messageVersion": "2.2.0",
    "acsChallengeMandated": "N",
    "acsOperatorID": "10055116",
    "acsReferenceNumber": "3DS_LOA_ACS_COPL_020200_00327",
  }
}
```

```

    "acsTransID": "0dedb4bc-434a-4df9-a0f9-38bd59ef0936",
    "acsURL": "https://acstest.libertybank.ge",
    "authenticationType": "02",
    "transStatus": "C"
  },
  "challengeRequired": true,
  "cReq":
"eyJhY3NUcmFuc0EljoiMGRlZGI0YmMtNDM0YS00ZGY5LWUwZjktMzhiZDU5ZWYwOTM2liwidGhyZ
WVEU1NlcnZlclRyYW5zSUQiOiI5MGYwNTM5Mi1jM2QyLTRIYTgtYTRkZi00MjM2ZWZhMzFhOWMiL
CjJaGFsbGVuZ2VXaW5kb3dTaXplIjoiMDUiLCJtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmVyc2lv
bil6ljlulMi4wln0",
  "order": {
    "tdsV2AuthStatus": "Required",
    "status": "Preparing"
  }
}

```

From this response Merchant takes cReq, opens Iframe in clients browser and sends POST request with creq inside form

```

<form id="formCReq" name="formCReq" action="https://acstest.libertybank.ge" method="post">
  <input type="hidden" id="creq" name="creq"
value="eyJhY3NUcmFuc0EljoiMGRlZGI0YmMtNDM0YS00ZGY5LWUwZjktMzhiZDU5ZWYwOTM2liwi
dGhyZWVEU1NlcnZlclRyYW5zSUQiOiI5MGYwNTM5Mi1jM2QyLTRIYTgtYTRkZi00MjM2ZWZhMzFhO
WMiLCJJaGFsbGVuZ2VXaW5kb3dTaXplIjoiMDUiLCJtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmV
yc2lvbil6ljlulMi4wln0" />
</form>

```

ACS page is opening inside Iframe. Clients enters OTP, then ACS sends POST response to the notificationURL which we sent in Areq step: Inside that POST response there will be cres

```

cres = eyJ0aHJlZURTU2VydmlvYVJhbnNRCi6lMnNiYTkyYTRiLWRmOTQtNDhkMCM0NGY3LTImYjE2ZTE3MTVh
NCi1mFjc1RyYW5zSUQiOiI5MGYwNTM5Mi1jM2QyLTRIYTgtYTRkZi00MjM2ZWZhMzFhOWMiLCJtZXNzYWdlVHlwZ
SI6IkNSZXMlLCJtZXNzYWdlVmVyc2lvbil6ljlulMS4wliwidHJhbnNTdGF0dXMiOiJlZiIn0

```

After obtaining cres, merchant proceeds to the final step.

EXEC Tran

Example of the request url:

<https://libertypaytxtest.lb.ge:6444/order/4323/exec-tran>

where:

- **order** – Order Id.
- **password** - Session Id or password.

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "tran": {
    "phase": "Single",
    "amount": "0.2",
    "authentication": {
      "cvv2Block":
        { "data": "" }
    },
    "conditions": {
      "cofUsage": "Recurring"
    }
  }
}
```

where:

- **data** – cvv.

Example of the response body:

```
{
  "tran": {
    "approvalCode": "388683",
    "match": {
      "tranActionId": "230519-11135194-000qm2=",
      "ridByPmo": "322625492"
    }
  }
}
```

P2P Credit

P2P Credit using PAN

First step - Create Order .

JSON

Example of the request header:

<https://libertypaytxttest.lb.ge:6444/order?terminalid=41>

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9Qlw==

Example of the request body:

```
{
  "order":{
    "typeRid":"P2P-Credit",
    "language":"ru",
    "amount":"1.00",
    "currency":"GEL",
    "description":"P2P-Credit",
    "custAttrs":[
      {"rid":"@TWPG@ApproveURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"},
      {"rid":"@TWPG@DeclineURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"},
      {"rid":"@TWPG@CancelURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"}
    ],
    "consumerDevice":{
      "browser":{
        "deviceId":"6996B7A6-24D9-4F5A-BE09-99AD420AAFFF",
        "acceptHeader":"application/json,application/jose;charset=utf-8",
        "ip":"127.0.0.1",
        "tzOffset":"-300",
        "platform":"Mozilla",
        "jsEnabled":true,
        "colorDepth":32,
        "screenW":1024,
        "screenH":800,
        "userAgent":"chrome",
```

```
        "language": "Ru-ru",
        "javaEnabled": true
    }
}
},
"dstToken": {
  "card": {
    "panBlock": {
      "data": "4125710020009128"
    },
    "entryMode": "ECommerce",
    "expiration": "0326"
  }
}
}
```

Example of the response body:

```
{
  "order": {
    "id": 2283,
    "password": "9m6743uhaykg",
    "status": "Preparing",
    "cvv2AuthStatus": "IneligibleOrder"
  }
}
```

Next step – exec tran.

Example of the request header:

<https://libertypaytxttest.lb.ge:6444/order/2283/exec-tran>

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "tran": {
    "type": "Credit",
    "phase": "Single",
    "amount": "2.00"
  }
}
```

Example of the response body:

```
{
  "tran": {
    "approvalCode": "388713",
    "match": {
      "tranActionId": "230523-14314532-000imn=",
      "ridByPmo": "322625966"
    }
  }
}
```

P2P Credit using stored token

First step - Create Order – may be zero amount.

Example of the request header:

<https://libertypaytxttest.lb.ge:6444/order?terminalid=41>

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "order":{
    "typeRid":"TWPG-Purchase",
    "language":"ru",
    "amount":"0",
    "currency":"GEL",
    "description":"test order",
    "hppCofCapturePurposes":[
      "Cit",
      "Recurring",
      "UnspecifiedMit"
    ],
    "custAttrs": [
      {
        "rid": "@TWPG@CardHolderLogin",
        "valAsStr": "Login121212"
      },
      {
        "rid":"Custom-needregcard",
        "valAsStr":"on"
      },
      {"rid":"@TWPG@ApproveURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"},
      {"rid":"@TWPG@DeclineURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"},
      {"rid":"@TWPG@CancelURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"}
    ],
    "consumerDevice":{
      "browser":{
        "deviceId":"6996B7A6-24D9-4F5A-BE09-99AD420AAFFF",
        "acceptHeader":"application/json,application/jose;charset=utf-8",
        "ip":"127.0.0.1",
        "tzOffset":"-300",
        "platform":"Mozilla",
        "jsEnabled":true,
        "colorDepth":32,
        "screenW":1024,
        "screenH":800,
        "userAgent":"chrome",
        "language":"Ru-ru",
        "javaEnabled":true
      }
    }
  }
}
```

Example of the response body:

```
{
  "order": {
    "id": 1939,
    "status": "Preparing",
    "password": "1ekphcnoq2sm5"
  }
}
```

Redirect client to TWPGA payment page to make first payment.

<https://libertypaytxtest.lb.ge/twpga?id=1939&password=1ekphcnoq2sm5>

After client finishes payment we need to Get Order Details. Use Id and password from previous response.

Example of the request header:

<https://libertypaytxtest.lb.ge:6444/order/1939?merchantId=41&password=1ekphcnoq2sm5&orderDetailLevel=2&merchantDetailLevel=2&ctsDetailLevel=2&cofpDetailLevel=2&consumerDetailLevel=2&consumerTokenDetailLevel=2&tokenDetailLevel=2&tranDetailLevel=2&actionDetailLevel=2>

Example of the response body:

```
{
  .....
  "storedTokens": [
    {
      "id": 1365
    }
  ],
  .....
}
```

From here we need only storedToken ID.

Next Step is to make another Create Order with typeRid P2P-Credit.

Example of the request header:

<https://libertypaytxtest.lb.ge:6444/order?terminalid=41>

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "order":{
    "typeRid":"P2P-Credit",
    "language":"ru",
    "amount":"2.00",
    "currency":"GEL",
    "description":"P2P-Credit",
    "custAttrs":[
      {"rid":"@TWPG@ApproveURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"},
      {"rid":"@TWPG@DeclineURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"},
      {"rid":"@TWPG@CancelURL","valAsStr":"https://libertypay.ge/Shop/out.jsp"}
    ],
    "consumerDevice":{
      "browser":{
        "deviceId":"6996B7A6-24D9-4F5A-BE09-99AD420AAFFF",
        "acceptHeader":"application/json,application/jose;charset=utf-8",
        "ip":"127.0.0.1",
        "tzOffset":"-300",
        "platform":"Mozilla",
        "jsEnabled":true,
        "colorDepth":32,
        "screenW":1024,
        "screenH":800,
        "userAgent":"chorme",
        "language":"Ru-ru",
        "javaEnabled":true
      }
    }
  },
  "dstToken":{
    "storedId": 1365
  }
}
```

Example of the response body:

```
{
  "order": {
    "id": 2283,
    "password": "9m6743uhaykg",
    "status": "Preparing",
    "cvv2AuthStatus": "IneligibleOrder"
  }
}
```

Next step – exec tran.

Example of the request header:

<https://libertypaytxttest.lb.ge:6444/order/2283/exec-tran>

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "tran": {
    "type": "Credit",
    "phase": "Single",
    "amount": "2.00"
  }
}
```

Example of the response body:

```
{
  "tran": {
    "approvalCode": "388713",
    "match": {
      "tranActionId": "230523-14314532-000imn=",
      "ridByPmo": "322625966"
    }
  }
}
```

Card To Card Transfer via TXPG API

JSON

Creating Single-Transfer Order

Example of the request url:

<https://libertypaytxtest.lb.ge:6444/order?terminalid=241>

where:

- **terminalid** – Merchant Id.

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUFQMIBUUKFOU0ZFUjoxMjNpb3BJT1Aj

Example of the request body:

```
{
  "order": {
    "typeRid": "Single-Transfer",
    "language": "ru",
    "amount": "25.00",
    "currency": "GEL",
    "hppRedirectUrl": "https://bank.com",
    "description": "Lololo"
  },
  "dstToken": {
    "card": {
      "panBlock": {
        "data": "4125706004381848"
      },
      "entryMode": "ECommerce"
    }
  }
}
```

where:

- **typeRid** - Order type.
- **Language** – Language.
- **amount** - Amount.

- **currency** – Currency.
- **description** - Additional info
- **hppRedirectUrl** – redirect url
- **dstToken** – Destination Card for transaction transfer

Example of the response body:

```
{  
  "order": {  
    "id": 9574,  
    "hppUrl": "https://libertypaytxtest.lb.ge/twpga",  
    "password": "3g5udgtebyez",  
    "status": "Preparing",  
    "cvv2AuthStatus": "Required"  
  }  
}
```

where:

- **id** - identifier of the created order. It should be passed in all requests related to the order.
- **status** – status of the order.
- **password** - password of the order.
- **hppUrl** - payment page url if not provided use <https://libertypaytxtest.lb.ge/twpga>.
- **cvv2AuthStatus** – cvv authorization status

Creating Source Token**Example of the request url:**

<https://libertypaytxtest.lb.ge:6444/order/9574/set-src-token?password=3g5udgtebyez>

where:

- **order** – Order Id.
- **password** - Session Id or password.

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUFQMIBUUKFOU0ZFUjoxMjNpb3BJT1Aj

Example of the request body:

```
{
  "token": {
    "card": {
      "panBlock": {
        "data": "5324343004638435"
      },
      "entryMode": "ECommerce",
      "expiration": "1124"
    }
  }
}
```

where:

- **token** – Source Card for transaction transfer.
- **Data** – Card Pan number
- **expiration** – Expiration Data.
- **entryMode** - EntryMode Type.

Example of the response body:

```
{
  "order": {
    "status": "Preparing",
    "cvv2AuthStatus": "Required",
    "tdsV1AuthStatus": "IneligibleOrder",
    "tdsV2AuthStatus": "Required",
    "otpAutStatus": "NotSupportedIss",
    "srcToken": {
      "id": 7004,
      "paymentMethod": "Card",
      "role": "Src",
      "status": "Active",
      "regTime": "2024-02-21 14:41:07",
      "displayName": "532434*****8435",
      "card": {
        "expiration": "1124",
        "brand": "Mastercard",
        "currency": "GEL",
        "country": "GEO"
      }
    }
  },
  "tdsData": {
```

```
"aReq": {
  "threeDSServerTransID": "d3f350c8-84af-497d-b121-8bdc8492efa8",
  "messageType": "AReq",
  "messageVersion": "2.2.0",
  "threeDSCompInd": "U",
  "threeDSRequestorAuthenticationInd": "01",
  "threeDSRequestorID": "IA00007A",
  "threeDSRequestorName": "IATRANSFER",
  "threeDSRequestorURL": "https://libertybank.ge/",
  "threeDSServerOperatorID": "DSRefNum",
  "threeDSServerRefNumber": "DSRefNum",
  "threeDSServerURL": "https://3dstestsrv.libertypay.ge:9844",
  "acquirerBIN": "LB_Terminal",
  "acquirerMerchantID": "777777",
  "cardExpiryDate": "2411",
  "deviceChannel": "02",
  "mcc": "2842",
  "merchantCountryCode": "268",
  "merchantName": "IATRANSFER",
  "messageCategory": "01",
  "purchaseAmount": "2500",
  "purchaseCurrency": "981",
  "purchaseExponent": "2",
  "purchaseDate": "20240221104048"
}
}
```

Next Merchant makes **AREQ Request**

Example of the request url:

<https://libertypaytxtest.lb.ge:6444/order/9569/tds-areq?password=1m77s8jnhff86>

where:

- **order** – Order Id.
- **password** - Session Id or password.
- **threeDSServerTransID** – Id that came with create source token response

Example of Authorization Header:

Authorization: Basic VGVyYWwWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "aReq": {
    "threeDSServerTransID": "d3f350c8-84af-497d-b121-8bdc8492efa8",
    "messageType": "AReq",
    "messageVersion": "2.2.0",
    "threeDSComplnd": "U",
    "threeDSRequestorAuthenticationInd": "01",
    "threeDSRequestorID": "IA00007A",
    "threeDSRequestorName": "IATRANSFER",
    "threeDSRequestorURL": "https://libertybank.ge/",
    "threeDSServerOperatorID": "DSRefNum",
    "threeDSServerRefNumber": "DSRefNum",
    "threeDSServerURL": "https://3dstestsrv.libertypay.ge:9844",
    "acquirerBIN": "LB_Terminal",
    "acquirerMerchantID": "777777",
    "cardExpiryDate": "2411",
    "deviceChannel": "02",
    "mcc": "2842",
    "merchantCountryCode": "268",
    "merchantName": "IATRANSFER",
    "messageCategory": "01",
    "purchaseAmount": "2500",
    "purchaseCurrency": "981",
    "purchaseExponent": "2",
    "purchaseDate": "20240221104048",
    "transType": "01",
    "browserAcceptHeader": "application/json,application/jose;charset=utf-8",
    "browserIP": "127.0.0.1",
    "browserJavaEnabled": true,
    "browserJavascriptEnabled": true,
    "browserLanguage": "Ru-ru",
    "browserColorDepth": "32",
    "browserScreenHeight": "800",
    "browserScreenWidth": "1024",
    "browserTZ": "-300",
    "browserUserAgent": "chorme",
    "notificationURL": "http://test.com/return"
  }
}
```

Example of the response body:

```
{
  "aRes": {
    "threeDSServerTransID": "d3f350c8-84af-497d-b121-8bdc8492efa8",
    "dsReferenceNumber": "DSRefNum",
    "dsTransID": "3c699b01-0403-4b58-b9d6-d6c5170ebf1a",
    "messageExtension": [
      {
        "name": "MesExt1",
        "id": "ID1",
        "criticalityIndicator": false,
        "data": {
          "valueOne": "value"
        }
      }
    ],
    "messageType": "ARes",
    "messageVersion": "2.2.0",
    "acsChallengeMandated": "N",
    "acsOperatorID": "10055116",
    "acsReferenceNumber": "3DS_LOA_ACS_COPL_020200_00327",
    "acsTransID": "c0c93257-b6dc-45be-80db-021a5cb1576b",
    "acsURL": "https://acstest.libertybank.ge",
    "authenticationType": "02",
    "transStatus": "C"
  },
  "challengeRequired": true,
  "cReq":
  "eyJhY3NUcmFuc0IEljoiyZBjOTMyNTctYjZkYy00NWJILTgwZGltMDIxYTVjYjE1NzZiliwidGhyZWVEU1NlcnZlclRyYW5zSUQiOiJkM2YzNTBjOC04NGFmLTQ5N2QtYjEyMS04YmRjODQ5MmVmYTgiLCJjaGFsbGVuZ2VXaW5kb3dTaXplIjoimDUiLCJtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmVyc2lubi6l6ljuMi4wln0",
  "order": {
    "tdsV2AuthStatus": "ChallengeInProgress",
    "status": "Preparing"
  }
}
```

From this response Merchant takes cReq, opens Iframe in clients browser and sends POST request with creq inside form

```
<form id="formCReq" name="formCReq" action="https://acstest.libertybank.ge" method="post">
  <input type="hidden" id="creq" name="creq"
value="eyJhY3NUcmFuc0IEljoiyZBjOTMyNTctYjZkYy00NWJILTgwZGltMDIxYTVjYjE1NzZiliwidGhyZWVEU1NlcnZlclRyYW5zSUQiOiJkM2YzNTBjOC04NGFmLTQ5N2QtYjEyMS04YmRjODQ5MmVmYTgiLCJjaGFsbGVuZ2VXaW5kb3dTaXplIjoimDUiLCJtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmVyc2lubi6l6ljuMi4wln0" />
</form>
```

ACS page is opening inside Iframe. Clients enters OTP, then ACS sends POST response to the notificationURL which we sent in Areq step: Inside that POST response there will be cres
cres = eyJ0aHJlZURTU2VydmVvVHJhbnNJRCI6ImNiYTkyYTRiLWVhbnRmOTQtdNDhkMCM04NGY3LTImYjE2ZTE3MTVhbnRlcmFjc1RyYW5zSUQiOiJjYXNNTAxMi1iMjI2LTRiZDQtdYTIyZC0wMzcwMjA3Y2lyMWIiLCJtZXNzYWdlVHlwZSI6IktzXMiLCJtZXNzYWdlVmVyc2lvbiI6IjIuMS4wIiwidHJhbnNTdGF0dXMiOiJlZn0

After obtaining cres, merchant proceeds to the final step.

EXEC Tran

Example of the request url:

<https://libertypaytxtest.lb.ge:6444/order/1810/exec-tran?password=1jtdfr528wm3p>

where:

- **order** – Order Id.
- **password** - Session Id or password.

Example of Authorization Header:

Authorization: Basic VGVybWluYWxTeXMvSUEwMDAxMDE6MTIzaW9wSU9QIw==

Example of the request body:

```
{
  "tran": {
    "phase": "Single",
    "amount": "25.00",
    "authentication": {
      "cvv2Block": {
        "data": "007"
      }
    }
  }
}
```

where: **data** – cvv.

Example of the response body:

```
{
  "tran": {
    "approvalCode": "384449",
    "match": {
      "tranActionId": "230424-10054137-000hlp=",
      "ridByPmo": "322615867"
    }
  }
}
```

Google Pay™

Google Pay is a wallet provider which means that Google Pay transfers card data securely to LIBERTY and we make regular card transactions from that, using one of the regular acquirers.

To enable Google Pay, simply go to the settings page for Google Pay and activate it. Note that when you use Google Pay you must adhere to Google's Terms of Service. Once you have enabled it, we will automatically show the Google Pay button in your payment window for applicable payments.

LIBERTY supports both "PAN only" and "Cryptogram 3DS" methods of authorization. If 3D-secure is needed for the transaction, LIBERTY will automatically take care of that. Since some 3D-secure authentications require the billing address of the customer, we recommend that you supply it to LIBERTY when you create the payment.

LIBERTY supports VISA and MasterCard transactions from Google Pay.

You can use Google Pay for single authorizations or for recurring billing authorizations.

Google Pay is free to use and we recommend enabling it to increase your conversion rate.