

Introduction

API Factoring is executed on the HTTPS protocol based on JSON queries.

The documentation consists of 3 main parts:

- Authorization description, API methods and error codes.
- Test manual.
- Manual on execution of the basic elements.

Authorization

Basic URL addresses

1. To interact with Revo services, two base addresses are used:
 - a. <https://r.revoplus.ru/> - the production address of the service.
 - b. <https://demo.revoplus.ru/> - the demo address of the service.
2. BASE_URL - the variable indicating the base address.

BASE_URL = <https://r.revoplus.ru/>

BASE_URL = <https://demo.revoplus.ru/>

The connection shall be made only via HTTPS protocol - when trying to connect via HTTP, 404 error will occur.

Authorization parameters

1. On Revo's side a unique store identifier and a secret key are generated, which are transferred to the partner:
 - store_id - unique store identifier. For one partner several unique identifiers can be generated, usually not less than 2. STORE_ID1 is used for Registration and Limit methods. STORE_ID2 - for all other methods.
 - secret_key - a secret key, used when generating the electronic digital signature for authentication of query parameters to protect the form from being launched by third parties. Key length from 8 bytes. Encryption algorithm SHA1.
2. For authorization, the partner sends POST query, using [digital signature](#) signature and unique store identifier store_id.
3. Examples of URL queries can be found in [API Methods](#) section.

Example of parameters

```
secret_key = "098f6bcd4621d373cade4e832627b4f6"
```

```
STORE_ID1 = 12
```

```
STORE_ID2 = 13
```

The principle of digital signature forming

A secret key `secret_key` is added to the data string in json format. SHA1 algorithm is used to the received string, and as a result a digital signature signature is formed.

Please note, that the length will always be 40 symbols according to SHA1 when forming the signature.

Algorithm of digital signature forming

```
require 'digest/sha1'
```

```
secret_key = '9fff8c602b08b00323567be0001480f6'
```

```
data = "{\"order_id\": \"FACTPRECHR152632\", \"amount\": \"8300.00\"}"
```

```
SIGNATURE = Digest::SHA1.hexdigest(data + secret_key)
```

The result of the encryption in the above example will be the string "cbfb21630cd585f59c3a50fc3365d8c26b97cd4e".

API Methods

Registration

The method returns a reference to the iFrame in order to get the limit. Upon completion of the form to the address specified in callback [urljson response](#) with the result of the decision on client's limit is being sent.

Depending on the information on the user in the Revo system, the form will have a different number of steps (you need to enter the primary phone for this) - see more on [REVO iFrame description](#).

```
POST BASE_URL/factoring/v1/limit/auth?store_id=STORE_ID1&signature=SIGNATURE
```

If the client has already filled in the personal data on the partner's website, it shall be sent in the query for the autocomplete of the corresponding form fields.

Parameters

callback_url string	URL for Revo response on decision for the client.
redirect_url string	URL for redirecting after clicking the button/link "Return to the online store." in the Revo form.
current_order object	The object containing information about the order.
order_id string	Unique order number. Maximum 255 characters. The unique random string can be used.
primary_phone string, <i>optional</i>	Client's phone number of 10 digits (without the country code).
primary_email string, <i>optional</i>	Client's email.
person object, <i>optional</i>	The object containing information about the client.
first_name string, <i>optional</i>	Client's name.
surname string, <i>optional</i>	Client's surname.
patronymic string, <i>optional</i>	Client's patronymic.
birth_date object, <i>optional</i>	Client's birth date in dd.mm.yyyy format.

The example of query in json format

```
{
  "callback_url": "https://shop.ru/revo/decision",
  "redirect_url": "https://shop.ru/revo/redirect",
```

```
"primary_phone": "9268180621",
"primary_email": "ivan@gmail.com",
"current_order":
{
  "order_id": "R001233"
},
"person":
{
  "first_name": "Peter",
  "surname": "Chernyshev",
  "patronymic": "Alexandrovich",
  "birth_date": "15.01.1975"
}
}
```

Response Parameters

status Response code.
integer

message A short text description of the response.
string

iframe_url Reference to the generated iFrame.
string

The example of the response for successful authentication.

```
{
  "status": 0,
  "message": "Payload valid",
  "iframe_url": "https://r.revoplus.ru/form/v1/af45ef12f4233f"
}
```

callback parameters

order_id Unique order number. Maximum 255 characters.
string

decision Decision on instalments. In case of positive decision - value **approved** (the order is waiting to be finalized). In case of negative decision - **declined**.
string

amount Amount in roubles with kopecks.

float

mobile_phone Client's phone number of 10 digits (without the country code).
string

email Client's phone number of 10 digits (without the country code).
string

loan_id The unique order number in Revo system.
integer

Example of callback in case of successful completion of the form:

```
{
  "order_id": "32423",
  "decision": "approved",
  "amount": 5000.00,
  "mobile_phone": "89262341793",
  "email": "ivan@gmail.com"
}
```

If the 'decision' is equal to 'declined' the 'amount' value will be zero.

Limit

The method for obtaining the client's limit amount by his phone's number. In case of new customers, it is not possible to obtain information about the limit by phone number only.

```
POST
BASE_URL/api/external/v1/client/limit?store_id=STORE_ID1&signature=SIGNATURE
```

Parameters

client The object containing information about the client.
object

mobile_phone Client's phone number of 10 digits (without the country code).
string

Response Parameters

status
integer Response
code.

message
string A short text description of the response.

client
object The object containing information about the client.

mobile_phone
string Client's phone number of 10 digits (without the country
code).

limit_amount
string The limit of funds, available to the client, in roubles with
kopecks.

status
string User status. Possible values:
active - the partial payments service to the amount of
limit_amount;
inactive - the partial payments service is not available to
the user;
new - a new user for which partial payment service to the
amount of **limit_amount** is available.

The example of query in json format

```
{
  "client":
  {
    "mobile_phone": "9031234567"
  }
}
```

Example of response, when the client is found in database

```
{
  "meta":
  {
    "status": 0,
    "message": "Payload valid"
  },
  "client":
  {
    "mobile_phone": "9031234567",
    "limit_amount": "9500.00",
    "status": "active"
  }
}
```

```
}
```

The example of response, when the client is found in the database, but the loan issuance is not possible

```
{
  "meta":
  {
    "status": 0,
    "message": "Payload valid"
  },
  "client":
  {
    "mobile_phone": "9031234567",
    "limit_amount": "6700.00",
    "status": "inactive"
  }
}
```

Example of response, when the client is not found in database

```
{
  "meta":
  {
    "status": 0,
    "message": "Payload valid"
  },
  "client":
  {
    "mobile_phone": "9031234567",
    "limit_amount": "0.00",
    "status": "new"
  }
}
```

Checkout

The method returns a reference to the iFrame for registration of the client's order. Upon completion of the form to the address specified in callback [url](#) [json response](#) with the result of the registration is being sent. In case of successful registration, funds in the amount are held on the client's account in the Revo system.

Depending on the information on the user in the Revo system, the form will have a different number of steps (you need to enter the primary phone for this) - see on [REVO iFrame description](#).

For business models where the client needs to arrange prepayment, the method supports 2 ways of prepayment:

- If the prepayment is made prior to calling the iFrame, then the information about it shall be transferred to `prepayment_amount`.
- If the prepayment is to be made after the iFrame is called, the corresponding setting on the Revo side is made. Parameter `skip_result_page` shall be set as true and address of prepayment page, to which the client will be redirected upon the completion of registration in iFrame shall be transferred to callback url.

If the client has already filled in the personal data on the partner's website, it shall be sent in the query for the autocomplete of the corresponding form fields.

POST

`BASE_URL/factoring/v1/precheck/auth?store_id=STORE_ID2&signature=SIGNATURE`

Parameters

callback_url
string

URL for Revo response on decision for the client.

redirect_url
string

URL for redirecting after clicking the button/link "Return to the online store." in the Revo form.

current_order
object

The object containing information about the order.

order_id
string

Unique order number. Maximum 255 characters. For example, order numbering in partner's system can be used.

valid_till
String, optional

The period when the order is considered valid (the period of funds holding). When the period is expired, the order is cancelled. Format: `dd.mm.yyyy hh:mm:ss+hh:mm`, where after "+" time zone related to GMT is specified. The default is 24 hours.

term
integer, optional

Period of installments in months.

amount
float

Order amount in roubles with kopecks.

prepayment_amount
float, optional

The amount of prepayment already paid by the client in roubles with kopecks.

primary_phone
string, optional

Client's phone number of 10 digits (without the country code).

primary_email
string, *optional*

Client's email.

person
object, *optional*

The object containing information about the client.

first_name
string, *optional*

Client's name.

surname
string, *optional*

Client's surname.

patronymic
string, *optional*

Client's patronymic.

birth_date
string, *optional*

Client's birth date in **dd.mm.yyyy** format.

cart_items
object, *optional*

The object containing block of information about the order.

sku
string, *optional*

Stock keeping unit.

name
string

Product name.

price
float

Product price.

sale_price
float, *optional*

Product price with the discount (if any).

quantity
integer

Product quantity.

brand

Product brand.

string, optional

skip_result_page

bool, optional

The flag that determines whether the page with the result of registration in the iFrame will be displayed.

The default - **false**.

true - upon the successful completion of registration immediate redirection by **redirect_url** is executed.

false - upon successful completion of registration, a window with the result will be displayed..

additional_data

object, optional

Object for transferring the block with additional information about the order.

name

string, optional

Field name.

value

string, optional

Field value.

The example of query in json format

```
{
  "callback_url": "https://shop.ru/revo/decision",
  "redirect_url": "https://shop.ru/revo/redirect",
  "primary_phone": "9268180621",
  "primary_email": "ivan@gmail.com",
  "current_order":
  {
    "order_id": "R001233",
    "valid_till": "21.07.2018 12:08:01+03:00",
    "term": 3,
    "amount": 59499.00,
    "prepayment_amount": 1000.00
  },
  "person":
  {
    "first_name": "Peter",
    "surname": "Chernyshev",
    "patronymic": "Alexandrovich",
    "birth_date": "15.01.1975"
  },
  "cart_items":
  [{
    "sku": "1231",
    "name": "Samsung Note 8",
```

```
"price": 55999,
"quantity": 1,
"brand": "Samsung"
},
{
  "sku": "23543",
  "name": "Brand case",
  "price": 3500,
  "sale_price": 2999,
  "quantity": 1,
  "brand": "Samsung"
}],
"skip_result_page": true,
"additional_data":
[
  {
    "name": "Color",
    "value": "Black"
  },
  {
    "name": "Size",
    "value": "Large"
  }
]
```

Response Parameters

status Response code.
integer

message A short text description of the response.
string

iframe_url Reference to the generated iFrame.
string

The example of the response in case of successful authentication.

```
{
  "status": 0,
  "message": "Payload valid",
  "iframe_url":
  "https://revo.ru/factoring/v1/form/6976174c5b6a1bb089d15b80e0a6afc62d4283fe"
}
```

Callback parameters

order_id Unique order number. Maximum

string 255 characters.

decision
string Decision on instalments. In case of positive decision - value **approved**. In case of negative decision - **declined**.

amount
float Amount to be paid in parts in roubles with kopecks.

prepayment_amount
float, *optional* The prepayment amount in roubles with kopecks.

total_amount
float, *optional* Total order amount, including prepayment.

term
integer Period of installments in months.

client
object The object containing information about the client.

primary_phone
string Client's phone number of 10 digits (without the country code).

primary_email
string, *optional* Client's email.

full_name
string Surname, name, patronymic with spaces.

first_name
string Client's name.

surname
string Client's surname.

patronymic Client's patronymic.

string

schedule
object The object containing information about the payments schedule.

date
string Payment date
in **dd.mm.yyyy** format.

amount
float Payment amount in
roubles with kopecks.

monthly_overpayment
float The monthly overpayment in roubles and kopecks.

If the 'decision' is equal to 'declined' the 'amount' value will be zero, and 'schedule' will be an empty block.

Example of callback in case of successful registration of the product:

```
{
  "order_id": "R107356",
  "decision": "approved",
  "amount": 6700.00,
  "prepayment_amount": 1000.00,
  "total_amount": 7700.00,
  "term": 3,
  "client":
  {
    "primary_phone": "8880010203"
    "email": "ivan@gmail.com",
    "full_name": "Ivanov Ivan Ivanovich",
    "first_name": "Ivan",
    "surname": "Ivanov",
    "patronymic": "Ivanovich"
  },
  "schedule":
  [
    {
      "date": "01.01.2018",
      "amount": 2933.33
    },
    {
      "date": "01.02.2018",
      "amount": 2933.33
    },
    {
      "date": "01.03.2018",
      "amount": 2933.33
    }
  ]
}
```

```
}
```

Schedule

The method returns information on the available preliminary payment schedules for the specified basket amount.

```
POST BASE_URL/factoring/v1/schedule?store_id=STORE_ID2&signature=SIGNATURE
```

Parameters

amount Amount to be paid in parts in roubles with kopecks.
float

The example of query in json format

```
{  
  "amount": 5000.00  
}
```

Response Parameters

message A short text description of the response.
string

payment_schedule Block of objects, containing information on preliminary payment schedules.
object

total The total amount of the instalment plan including the overpayment.
float

monthly_payment Approximate value of the monthly payment including the overpayment.
float

monthly_overpayment The monthly overpayment

float

in roubles and kopecks.

term
int

Period of installments in months.

payment_dates
object

The object containing information about the payments schedule.

date
string

Payment date in **dd.mm.yyyy** format.

amount
float

Payment amount in roubles with kopecks.

The example of response, when two payment schedules are available: for 3 and 6 months.

```
{
  "status": 0,
  "message": "Payload valid",
  "payment_schedule": [
    {
      "total": 7000.01,
      "monthly_payment": 2334,
      "monthly_overpayment": 666.67,
      "term": 3,
      "payment_dates": [
        {
          "date": "11.06.2018",
          "amount": 2334.00
        },
        {
          "date": "09.07.2018",
          "amount": 2334.00
        },
        {
          "date": "09.08.2018",
          "amount": 2332.01
        }
      ]
    },
    {
      "total": 6500,
      "monthly_payment": 1100,
      "monthly_overpayment": 250,
      "term": 6,
      "payment_dates": [
        {
          "date": "11.06.2018",
```

```
    "amount": 1100.00
  },
  {
    "date": "09.07.2018",
    "amount": 1100.00
  },
  {
    "date": "09.08.2018",
    "amount": 1100.00
  },
  {
    "date": "10.09.2018",
    "amount": 1100.00
  },
  {
    "date": "09.10.2018",
    "amount": 1100.00
  },
  {
    "date": "09.11.2018",
    "amount": 1000.00
  }
]
```

Status

The method returns information on the status of the order.

```
POST BASE_URL/factoring/v1/status?store_id=STORE_ID2&signature=SIGNATURE
```

Parameters

order_id Unique order number. Maximum 255 characters.
string

The example of query in json format

```
{
  "order_id": "R107356"
}
```

Response Parameters

status Respons
integer e code.
r

message string	A short text description of the response.
current_order object	The object containing information about the order.
order_id string	Unique order number. Maximum 255 characters.
expired bool	The flag, reflecting the order validity status (funds holding). For valid orders - false . Becomes true with maturity valid_till .
status string	Information on order status. Possible values: pending, hold, finished, canceled, declined, refunded .
decision string	Information on limit status. In case the limit is approved - approved . In case the limit is declined - declined .
amount float	The amount to be paid in parts by the client in roubles with kopecks.
term integer	Period of installments in months.

Status and Decision values

Decision	Status	Description
null	pending	The client haven't completed the registration in the form. The decision on the limit has not been taken.
approved	hold	The limit is approved, the funds are held, order is waiting to be finalized.

approved **finished** Order is finalized. In case of the subsequent change **expired**, or in case of partial refund the value **status** remains unchanged.

approved **canceled** The order is cancelled. In case of the subsequent change **expired** the value **status** remains unchanged.

approved **expired** The limit is approved, the deadline for ordering has expired.

approved **refunded** A full refund has been made for the order. In case of partial refund **status** will remain **finished**.

approved **declined** The limit is approved, the partial payments service is not available to the client (for example, the order amount exceeds the limit).

declined **declined** The limit is declined due to Revo politics.

The example of the response, in case the client has completed the registration in iFrame and is awaiting the response on the order from the Partner

```
{
  "status": 0,
  "message": "Payload valid",
  "current_order":
  {
    "order_id": "FACTPRECHR00004768",
    "expired": false,
    "status": "hold",
    "decision": "approved",
    "amount": 4999.0,
    "term": 3
  }
}
```

The example of the response, in case the client completed the registration in iFrame, but the partner has cancelled the order

```
{
  "status": 0,
  "message": "Payload valid",
  "current_order":
  {
    "order_id": "FACTPRECHR00004768",
    "expired": true,
    "status": "canceled",
    "decision": "approved",
    "amount": 4999,
    "term": 3
  }
}
```

The example of the response, in case the client completed the registration in iFrame, partner has confirmed the order.

```
{
  "status": 0,
  "message": "Payload valid",
  "current_order":
  {
    "order_id": "FACTR00004755",
    "expired": false,
    "status": "finished",
    "decision": "approved",
    "amount": 1000,
    "term": 3
  }
}
```

The example of the response, in case of refusal due to Revo politics.

```
{
  "status": 0,
  "message": "Payload valid",
  "current_order":
  {
    "order_id": "FACTPRECHR00004721",
    "expired": true,
    "status": "declined",
    "decision": "declined",
    "amount": 6498,
    "term": null
  }
}
```

The example of the response, in case the holding period for the order has expired. The order is cancelled.

```
{
  "status": 0,
  "message": "Payload valid",
  "current_order":
  {
    "order_id": "FACTPRECHR141531",
    "expired": true,
    "status": "expired",
    "decision": "approved",
    "amount": 9000,
    "term": 3
  }
}
```

The example of the response, in case the client successfully completed the registration in iFrame, partner has confirmed the order. The full refund has been executed.

```
{
  "status": 0,
  "message": "Payload valid",
  "current_order":
  {
    "order_id": "FACTPRECHR00004714",
    "expired": true,
    "status": "refunded",
    "decision": "approved",
    "amount": 734.51,
  }
}
```

```
"term": 3
}
}
```

Change

Method for changing the amount of already created order.

```
POST
BASE_URL/factoring/v1/precheck/change?store_id=STORE_ID2&signature=SIGNATURE
```

Parameters

order_id
string
Unique order number.
Maximum 255 characters.

amount
float
Amount in roubles with kopecks.

cart_items
object
The object containing block of information about the order.

sku
string, *optional*
Stock keeping unit.

name
string
Product name.

price
float
Product price.

sale_price
float, *optional*
Product price with the discount (if any).

quantity
integer
Product quantity. If a non-integer number is transmitted, it is not displayed in the form.

brand
string, *optional*

Product brand.

The example of query in json format

```
{
  "order_id": "R107356",
  "amount": 59999.00,
  "cart_items":
  [{
    "sku": "1231",
    "name": "Samsung Note 8",
    "price": 55999,
    "quantity": 1
  }]
}
```

Response parameters

status Response code.
integer

message A short text description of the response.
string

schedule The object containing information about the payments schedule.
object

date Payment date in **dd.mm.yyyy** format.
string

amount Payment amount in roubles with kopecks.
float

The example of the response, in case of successful change of the order:

```
{
  "status": 0,
  "message": "Payload valid",
  "schedule":
  [
    {
      "date": "01.01.2018",
      "amount": 2933.33
    },
    {
      "date": "01.02.2018",
      "amount": 2933.33
    }
  ]
}
```

```
}  
{  
  "date": "01.03.2018",  
  "amount": 2933.33  
}  
}]  
}
```

Cancel

Method for order cancellation. Upon cancellation, the previously held client's funds will be unblocked.

Parameters

order_id Unique order number. Maximum 255 characters.
string

The example of query in json format

```
{  
  "order_id": "R107356"  
}
```

Response Parameters

status Response code.
integer

message A short text description of the response.
string

The example of the response in case the file is successfully uploaded

```
{  
  "status": 0,  
  "message": "Payload valid"  
}
```

Finish

Method for finalizing the order by transferring a sales contract to be serviced by Revo. The query shall be sent with the content type multipart/form-data. Two keys shall be specified in the query. The first key with the body name, which shall specify the body of the json query. The second key with the check name, where the file is attached (fiscal document). Signature is formed according to the basic principle, without the second key.

When trying to finalize the order with expired 'valid_till', the 'cancel' method will be executed.

```
POST
BASE_URL/factoring/v1/precheck/finish?store_id=STORE_ID2&signature=SIGNATURE
```

Parameters

order_id Unique order number. Maximum 255 characters.
string

amount Amount in roubles with kopecks.
float

check_number The number of the fiscal document in the partner system (for example, the check number).
string

The example of query in json format

```
{
  "order_id": "R107356",
  "amount": 6700.00,
  "check_number": "ZDDS3123F"
}
```

Response Parameters

status Response code.
integer

message A short text description of the response.
string

The example of the response in case the file is successfully uploaded

```
{
  "status": 0,
  "message": "Payload valid"
}
```

Return

Method of performing the procedure for the full or partial return of the order. Only already finalized order can be returned. If the order has not yet been finalized, instead of returning it, it shall be

cancelled by using the [Cancel](#) method. A partial refund can be made no earlier than on the next day after the finalization. When the funds are returned to the client's account, they are credited in total, including overpayments, if the client has already made payments to pay off the instalments.

```
POST BASE_URL/factoring/v1/return?store_id=STORE_ID2&signature=SIGNATURE
```

Parameters

order_id Unique order number. Maximum 255 characters.
string

amount Returned amount in roubles with kopecks. The return can be either total or partial.
float

The example of query in json format

```
{  
  "order_id": "R001233",  
  "amount": 2010.00  
}
```

Response Parameters

status Response code.
integer

message A short text description of the response.
string

The example of the response in case of successful processing the query for return

```
{  
  "status": 0,  
  "message": "Payload valid"  
}
```

The example of the response in case of unsuccessful processing the query for return

```
{  
  "status": 10,  
  "message": "JSON decode error"  
}
```

Errors codes

Code	Message	Comments
0	Payload valid	Everything ok.
10	JSON decode error	Incorrect json query.
20	Order <code>order_id</code> missing	<code>order_id</code> not specified.
21	Wrong order <code>order_id</code> format	Wrong <code>order_id</code> format
22	Order exists	The order with the given <code>order_id</code> already exists and is finalized.
23	Order expired	The order with the given <code>order_id</code> expired.
24	Order with specified id not found	The order with the given <code>order_id</code> not found.
30	Wrong order <code>order_sum</code> format	Wrong <code>order_sum</code> format.
32	Order amount is different from the amount specified before	The order amount specified during order finalization is different than the amount, for which the order is placed. Finalization not executed.
33	Order amount is outside of <code>tariff_limits</code>	The order amount is outside the range set in the partner's tariff. The order not created.
35	Order prepayment amount is wrong	The prepayment <code>prepayment_amount</code> value exceeds <code>amount</code> .
40	Order <code>callback_url</code> missing	<code>Callback_url</code> not specified.
41	Order <code>redirect_url</code> missing	<code>redirect_url</code> not specified.
50	Store id is missing	<code>store_id</code> not specified.

Code	Message	Comments
51	Store not found	Store with the id <code>store_id</code> not found.
60	<code>Signature</code> missing	Digital signature <code>signature</code> missing.
61	<code>Signature</code> wrong	The given digital signature <code>signature</code> is incorrect.
62	Error saving file	Error while file saving.
70	Phone number is different	Phone number differs from the one, given in the order.
71	Client has not enough limit	Client has not enough limit to execute partial payment of the order.
80	Unable to finish - order is already finished/canceled	Unable to finish the order - the order with given <code>order_id</code> is already finished or cancelled.
81	Unable to cancel - order is already finished/canceled	Unable to cancel the order - the order with given <code>order_id</code> is already finished or cancelled.
82	Unable to change - order is already finished/canceled	Unable to change the order - the order with given <code>order_id</code> is already finished or cancelled.
90	Cart items are missing	Unable to change the order - the information on the basket not transferred.
100	At the moment the server cannot process your request	In all other cases.

Testing and debugging

Integration testing and debugging are executed on the demo server (<https://demo.revoplus.ru>). When entering the telephone number in the questionnaire, it is recommended to use the non existing operator prefix 888, so that sms messages are not sent to real people. It is not allowed to use this prefix on the production server.

All verification codes and pin codes 8888.

- In order to obtain the denial when using [Registration](#) it is necessary to specify a phone that starts with 88821.
- In order to obtain the approval when using [Registration](#) it is necessary to specify a phone that starts with 888, except 88821.
- For testing on the production server, the code is set in the settings of the partner's store. It can be, for example 7777, instead of 8888.
- In the questionnaire there is the verification based on surname, name and patronymic + birth date or passport number to coincide with the existing client, so during testing it is necessary to enter various data into these fields.

REVO iFrame description

Client's registration

1-2. Completing the questionnaire.

3. Transfer to the phone number verification screen with the code from the SMS message.

3.1 If the client made an error when entering the phone number, it is possible to move to the step with the phone number entering.

4.1 Upon entering the correct sms code, the window with the result "The registration was successful" and information about the available limit is displayed.

4.2 Or the window with the result "Unfortunately, 'Partial payment' is not available to you." is displayed.

When calling iframe Revo, the first screen (1) displays fields for entering personal data:

- Surname
- Name
- Patronymic
- Birth date
- Mobile phone number
- Email
- Passport series and number

Validation:

- Surname, name, patronymic - only the data in Cyrillic is accepted. Data type string
- Birth date - is given in dd.mm.yyyy format, data type date.
- Mobile phone number - it is necessary to enter in 10-digit format, i.e. (888)1231212. Data type string
- Email - the field for entering email, data type string.

- Passport series and number - the length must be exactly 10 characters, data type string.

When the iframe form is called, it is possible to auto-complete fields through the body json query.

Client's authorization

1. Clicking the Login button on the first page of the form.
 - 2.1 Entering a mobile phone number and pressing the "Send SMS" button.
 - 2.1 If the mobile number does not exist in the Revo database, the client proceeds to the window with the information message "We haven't found you with the specified phone number". Clicking on the "Register in 1 minute" button leads to the first step of the form.
 - 3.1 If the mobile phone number exists in the Revo database, the client proceeds to the Mobile Phone Confirmation screen with the code from the SMS message.
 - 3.2 If the client made a mistake while entering the phone number, then it is possible to proceed to the step with entering the phone number.
 - 4.1 Upon entering the correct sms code, the window with the result "The registration was successful" and information about the available limit is displayed.
 - 4.2 Or the window with the information "Unfortunately, 'Partial payment' is not available to you." is displayed.

By clicking on the FAQ button, a list of frequently asked questions is displayed in the upper right corner of the form.

When authorizing a client, a 30-minute session is created, during which repeated calls to the client immediately open the final screen of any form. If the client hasn't completed the registration in the form after the expiration of 'valid_till', then error message will be issued "Unfortunately, the reservation time of the order has expired."

When authorizing a client, a 30-minute session is created, during which repeated calls to the client immediately open the final screen of any form.

If the client hasn't completed the registration in the form after the expiration of 'valid_till', then error message will be issued "Unfortunately, the reservation time of the order has expired."

Purchase formalities

Completing the questionnaire.

- 2.1 Transfer to the phone number verification screen with the code from the SMS message.
- 2.2 If the client made a mistake while entering the phone number, then it is possible to proceed to the step with entering the phone number.
- 3.1 If the prepayment is not required to place the order, then transfer to the screen where the client shall select one of the possible payment periods for the given amount of the order is executed.

3.1 If it is necessary to arrange the prepayment to place the order, then a transfer to the screen where the client shall select one of the possible payment periods for the given amount of the order, as well as indicate the desired amount of the advance payment is executed.

4.1 Upon the the successful registration of the order without prepayment, the result window with the information "Registration completed successfully" and the button for returning to the store are displayed.

4.2 Or the window with the information "Unfortunately, 'Partial payment' is not available to you." is displayed.

4.3 Upon the successful registration of the order with a prepayment, the result window with the information "The registration was successful" and the button for switching to payment by the card is displayed.

4.4 Or the window with the information "Unfortunately, 'Partial payment' is not available to you." is displayed.

Presentation on site

Recommendations on the presentation of the service Partial payment on the partner site is presented in the presentation. Below the instructions for implementing individual Revo elements from the presentation can be found.

Calling iFrame

By clicking on the "Register in 1 minute" button, "?" on the right of the inscription "or 150 R/month" and "Payment" with the selected "Partial payment" method it is necessary to call iFrame Revo for registration and ordering, respectively. For this, you need to receive the link to iFrame and transfer it to REVO plugin js method using the [Registration](#) or [Checkout](#) method.

iframe_url – The address of the opened iFrame, mandatory parameter. target_selector – selector of the element, inside which the iFrame shall be inserted.

Then the provided Revo plugin js works (implemented as a REVO module), which inserts `<iframe src=iframe_url />` and manipulates this iFrame.

The plugin is available at: `https://{BASE_URI}/javascripts/iframe/v2/revoiframe.js` and it can be added to the online store page.

```
<script src="https://{BASE_URI}/javascripts/iframe/v2/revoiframe.js"></script>
```

This plugin provides also the possibility of receiving events: closing the form - onClose, loading the form - and making a decision on the application - onResult.

```
REVO.Form.show(iframe_url, target_selector);
```

In case, it is necessary to display the iFrame on a separate page, `iframe_url` shall be opened directly. The background setting for this page is on the Revo side.

```
<script src="https://{BASE_URI}/javascripts/iframe/v2/revoiframe.js"></script>
```

```
REVO.Form.onClose(function () { alert('closed'); });
REVO.Form.onLoad(function () { console.log('frame loaded'); });
REVO.Form.onResult(function() { console.log('result'); });
```

Displaying the available limit

If the client has already passed the registration and received a limit, then the funds available to him for the Partial payment can be displayed using the Limit method.

Testing and debugging

Test parameters

Test mobile numbers to be entered into [iframe](#):

8881XXXXXX - All approved (second registration = re-ordering)

88821XXXXX - completely declined

88822XXXXX - only the loan is approved

Phone number verification code - 8888

These parameters are intended for demo service only.

Test cases

Detailed description of calling the form for client's registration.

Registration method is used to call the form.

Parameters required to call the form:

- Base service URL for testing and debugging
- store_id
- signature

Description of the formation of a query to receive an iframe call reference:

- Body json of the query is entered into the data variable.
- We change data into json format
- Key name is entered into the secret_key variable.

- Signature is formed into signature variable.
- Body json of the query is entered via the puts command to send the query through the client

Data to send the query into REST CLIENT:

POST URL is

used: `https://demo.revoup.ru/factoring/v1/limit/auth?store_id=72&signature=347e8cff27d30b5200c8b32def4365ebbf4270d0`

Body

JSON:{"callback_url":"https://shop.ru/revo/decision","redirect_url":"https://shop.ru/revo/redirect","primary_phone":"9268180621","primary_email":"ivan@gmail.com","current_order":{"order_id":"R001233"}}

The example of signature forming on ruby:

```
data = {
  "callback_url": "https://shop.ru/revo/decision",
  "redirect_url": "https://shop.ru/revo/redirect",
  "primary_phone": "9268180621",
  "primary_email": "ivan@gmail.com",
  "current_order":
  {
    "order_id": "R001233"
  }
}
=> {:callback_url=>"https://shop.ru/revo/decision",
:redirect_url=>"https://shop.ru/revo/redirect", :primary_phone=>"9268180621",
:primary_email=>"ivan@gmail.com", :current_order=>{:order_id=>"R001233"}}

data = data.to_json
=>
"{\"callback_url\": \"https://shop.ru/revo/decision\", \"redirect_url\": \"https://shop.ru/revo/redirect\", \"primary_phone\": \"9268180621\", \"primary_email\": \"ivan@gmail.com\", \"current_order\": {\"order_id\": \"R001233\"}}"

secret_key = "9fff8c602b08b00323567be0001480f6"

signature = Digest::SHA1.hexdigest(secret_key + data)
=> "347e8cff27d30b5200c8b32def4365ebbf4270d0"

puts data
{"callback_url":"https://shop.ru/revo/decision","redirect_url":"https://shop.ru/revo/redirect","primary_phone":"9268180621","primary_email":"ivan@gmail.com","current_order":{"order_id":"R001233"}}
```

Detailed description of sending the query of order finalization

[Finish](#) method is used to finalize the order

Parameters required to finalize the order:

- Base URL for testing and debugging
- store_id

- signature

Description of forming the query on order finalization:

- Body json of the query is entered into the data variable.
- We change data into json format
- Key name is entered into the secret_key variable.
- Signature is formed into signature variable.
- Body json of the query is entered via the puts command to send the query through the client

Data to send the query into REST CLIENT:

POST URL isu=

used: https://demo.revoup.ru/factoring/v1/precheck/finish?store_id=72&signature=70189f8a4f413fcb01c8933cae50f4341fe8fdee

Body JSON c key =

body:{"order_id":"FACTPRECHR00005384","amount":4999.0,"check_number":"sdfhk"}

Attachment of fiscal document with key=check.

The example of sending a query through API CLIENT:

The example of signature forming on ruby:

```
data = {order_id: "FACTPRECHR00005384", amount: 4999.0, check_number: "sdfhk"}
=> {:order_id=>"FACTPRECHR00005384", :amount=>4999.0, :check_number=>"sdfhk"}
data = data.to_json
=>
"{\"order_id\": \"FACTPRECHR00005384\", \"amount\": 4999.0, \"check_number\": \"sdfhk\"}"
secret_key = "9fff8c602b08b00323567be0001480f6"
=> "9fff8c602b08b00323567be0001480f6"
signature = Digest::SHA1.hexdigest(data + secret_key)
=> "70189f8a4f413fcb01c8933cae50f4341fe8fdee"
2.3.3 :012 > puts data
{"order_id":"FACTPRECHR00005384","amount":4999.0,"check_number":"sdfhk"}
```

The example of query through REST client POSTMAN:

Features