

# 1. BASICS:

Store system communicates with **Payment gateway** using web-services.

## 1.1. Services

There are four web-services provided for access from Store system to Payment gateway:

1. **order** - managing payment orders
2. **status** - retrieving orders processing status
3. **check** - retrieving mail notifications
4. **ticket** - retrieving information about issued tickets

Service	Test location	Production location
order	https://tws.egopay.ru/order/v2/	https://ws.egopay.ru/order/v2/
status	https://tws.egopay.ru/status/v2/	https://ws.egopay.ru/status/v2/
check	https://tws.egopay.ru/check/v1/	https://ws.egopay.ru/check/v1/
ticket	https://tws.egopay.ru/ticket/v1/	https://ws.egopay.ru/ticket/v1/

Payment gateway conforms to the following industry standards and protocols:

- Message Envelope: **SOAP 1.1**
- Message Transfer: **HTTP 1.1**
- Encryption: **TLS 1.0** or **TLS 1.2**
- Web Service Description Language: **WSDL 1.1**
- Data Transfer: **XML 1.0**
- Encoding: **UTF-8**

## 1.2. Identification

Basic HTTP authorisation is required for any request. The Store system will be provided with test and production access credentials - **login and password** or **certificate**. Access to Payment gateway services can be limited by the list of ip addresses on the basis of security factor. Store is also provided with **shop\_id** - special unique Store identifier in Payment gateway system for request performance.

**Note: shop\_id is not equal to Acquirer terminals (MID/TID).** **Shop\_id** is used to perform operations with the order, while **Acquirer terminals** are used to complete authorization.

## 1.3 Types and global rules

**Float** type parameters values must be **separated with dot** (".") symbol. This rule is especially relevant to **amount** parameter in which major monetary unit (e.g. rubles) and minor monetary unit (e.g. kopecks) must be separated by dot.

**Datetime** parameter values must be in **UTC** timezone format.

Blank space is also a symbol.

The **general timeout** for responses is **30 seconds**. **Average services method call response** time equate **2-3 seconds**.

In case of SOAP fault the **faultstring** field will contain an error code (p. 8.1). Store system will receive SOAP fault to incorrect by scheme or by values request.

Example:

```
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>Server</faultcode>
      <faultstring>Internal Server Error</faultstring>
      <detail />
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

## 1.4 Schemes

There are 2 basic schemes of card data entry:

1. **payment-gateway-side**: all card PCI-DSS sensitive data entering on Payment gateway side, and can't be provided in full form into Store system.
2. **store-system-side**: for PCI-DSS certified merchants have an option to enter card data on own side and provide it from Store system to Payment gateway.

For **payment-gateway-side** Payment gateway support only **re-direct** mechanic. On order registration request (in **register** method response) Store system will receive re-direct url to perform Customer browser re-direct to this url. This url will land Customer to secure Payment gateway page and

card data entry will be made on secured Payment gateway side.

For **store-system-side** Payment gateway support 2 mechanics:

1. **browser-to-host**: Store system on own payment page have to embed special Payment gateway code and call it to send card data from Customer web browser to Payment gateway host.
2. **host-to-host**: Store system collect card data from Customer browser to its own host itself and transfer from it to Payment gateway.

## 1.5 Order and operations on it

**Order** is Payment gateway primary entry, and represents settlement acts between Store and Customer in context of their deal.

Acts are:

1. When the Customer make a **Payment** to Store.
2. When the Store make a **Refund** to Customer.

The **Payment** may have the following stages:

1. Authorization - Customer's bank holds amount of money on Customer's account. After this Payment gateway **mark order as payed**.
2. Charge - Customer's bank charges amount of money Customer's account. This stage may be reached only in process of settlement between Store and Acquirer.

The Store system initiate **operations** on order by calling proper **order** service methods. Payment gateway supports:

1. **register** - to initiate the Payment;
2. **cancel** - to interrupt the Payment **before** authorization stage;
3. **reject** - to interrupt the Payment **after** authorization stage and **before** order confirmation (p. 1.6);
4. **confirm** - as described in p. 1.6;
5. **refund** - to make refund to Customer.

## 1.6 Confirmation

Confirmation is the process of informing that Payment gateway have to start settlement initiation process between Store and Acquirer. Before the order is confirmed, Payment gateway system holds money authorization on Customer bank account without charging of it. After sending confirmation, capture process begins and Payment gateway schedules payment settlement process.

Payment gateway system supports 2 basic configuration of confirmation:

1. **auto** - Payment gateway system will schedule settlement process right after the order is payed, this is default configuration.
2. **manual** - Payment gateway system will schedule settlement process **only** after receiving **confirm** request from Store system.

Store has a 2 options for setup Payment gateway reaction on expiry time of confirmation awaiting:

1. **confirm-on-expire** - will force automatic scheduling confirmation for all unconfirmed orders.
2. **cancel-on-expire** - will cancel all all unconfirmed orders.

**Note:** Store may configure expiration period, ask Support Team for this.

## 1.7 Order processing

A set of activities in order processing depends on

1. Card entry scheme (p. 1.4)
2. **3d-secure** enable status
3. **fraudcheck** enable status

and consists of the:

1. (optional, only in **payment-gateway-side** scheme) Card data entry on Payment gateway secure page (avg. 1-3 min)
2. (optional, only for **3d-secure** enabled) Payment gateway request Customer's bank for enrolment Customer's card in **3d-secure** program (avg. 2-10 secs)
3. (optional, only for **3d-secure** enabled) Customer passing **3d-secure** authentication procedures, as usual Customer receive and enter secure code on Customer's bank page (avg. 1-2 min)
4. (optional, only for **fraudcheck** enabled) Fraud management procedures (avg. 1-2 sec)
5. Payment gateway request Customer's bank for authorization of Payment (avg. 2-5 sec).

**Note:** Store system can set Customer browser **re-direct** pages settings, to assign web pages to which Payment gateway will initiate Customer browser **re-direct** after successful or unsuccessful order processing completion. If Store request has no **re-direct** pages settings, Customer will be **re-directed** to Store **home page**, registered in Payment gateway database.

**Tip:** If Store system want to use **timelimit** to limit total time of processing, it has to appreciate above information and total overhead from Payment gateway internal processes (avg. 2-3 sec).

**Tip:** Store system can **limit** the list of cards for performing order payment by: - setting **default payment document type**; - using **allowed cards** filters (Ask Support team to create **card profile(s)**). Fraud control and marketing action are the main purposes for using this feature(s).

In case of **payment-gateway-side** card data entry scheme the following **features** are available:

1. **standard interface** Payment page. **Standard** pages are **automatically adapted** according to the browser or device window width: less then 800 pixels - mobile, more then 800 pixels - standard interface. Russian **language** of graphical user interface is set by **default**. Ask Support team to adjust others available translations (p 8.6).
2. standard **iframe** Payment page. **Iframe** page can be used only in case of **SSL-certificate** presence on Store web site. Default iframe page size - 700x500 pixels. Ask Support team to assign **iframe** page to Store **shop\_id**.

3. Payment page(s) **customization** - ask our Support team to provide **standard templates** for changes performing. Blocks **#footer**, **#header**, **css** can be changed. Pictures can be added. All **customized** payment pages **should be** checked, approved, named, uploaded by Support team. To manage several customized pages opening Store system should use Showcase parameter. Order **comment** can be added on Payment page.

## 1.8 Customer notification about complied authorization

According to Payment Systems standards the Customer must be notified about complied payment. Notification responsibility can be divided among Store and Payment gateway, or laid on one the participants of this process.

Payment gateway supports the following methods of Customer notification:

1. via **mail** - letter will contain the mail information about successful authorization (date, merchant name, sum etc.) This method can be used only if Store system transfer description/customer/email parameter. Optionally, the format of mails and the name of mailing server can be changed according to Store business demands.
2. via **browser page** - successful/unsuccessful authorization data will be shown in Customer web page after payment completion. This method can be used only if card date input is performing on payment-gateway-side.

If Store decides to perform Customer notification by its own, Payment gateway can **deactivate notification**.

To set up one of the chosen variants Store must ask Support Team for it.

## 1.9 Obtaining processing status and results

Payment gateway supports the following types of Store system informing about order processing status and results:

1. **push** - Payment gateway call Store system service. It means that Store system must support proper notification contract, described in separate document (Notification service description). Ask Support Team for it.
2. **pull** - Store system call Payment gateway service. It is described in this document (p. 3). Store system decides on its own the frequency and periodicity of order status requests.

**Tips:** Combined variant assumes that Store system will use both push notification and pull request types. This approach is the most preferable as in case of unavailability of Store system or Payment gateway notification service, Store system can request for order status via different methods of status service or vice versa.

## 1.10 Obtaining payment mail notification

Optionally, Store system can receive notification results in raw, informative format by using **check** service:

1. **push** - Payment gateway services will send the copy of every generated notification letter on the email address specified by the Store. Ask Support Team to add your or your company's assigned email address to mailing list.
2. **pull** - Store system can request for all or selected notification mails by using **check** service. Response is provided in xml format.

More information about **check** service can be found in separate document (Mail notification retrieve). Ask Support Team for it.

## 1.11 Liability levels

1. **reliable** - Store provide Payment gateway with data for access to goods/services management system. While processing the order payment, Payment gateway will request for availability of certain goods/services, and in case of positive response it will send request for receipt formation for future services or goods delivery.
2. **not reliable** - Store system independently controls order cart formation, payment status, goods/services delivery.

**Note:** Store system doesn't obtain exclusive right to choose one or another Payment gateway liability level. It is approved by Store together with the Acquirer.

# 2. ORDER REGISTRATION:

Order registration in PG can be performed with **several** methods of **service order**. **All** of these methods contain **basic information** about the **order**, **customer** and **postdata** block, but differs in included functionality. The following methods are available:

1. register\_simple - contains **only** basic parameters for order registration;
2. register\_online - contains basic parameters, cart description block for reliable type of integration and recurring block of parameters for performing recurring payments. **Usually** is used for **recurring scheme** of charges.
3. register\_offline - contains basic parameters, cart description block for reliable type of integration and allows to register orders for further offline payment. **Usually** is used for **offline types** of payments. Is described in **Offline payments functional description**.
4. register - contains basic parameters, cart description block for reliable type of integration, offline payments functional and block of parameters for performing combined type of payments. **Usually** is used for **reliable type of integration**, and **combined type of sale**.

**Note:** The choice of one method or another **depends on** the **business needs** of OS. **Recurring payments** functional, **combined sales** functional and **reliable type of integration** functional are described in separate documents. Ask Support Team for it.

**Note:** This document contains **only** register\_simple request as the basic variant of request for performing order registration in PG system. If it doesn't respond OS business needs ask Support Team to provide the certain document from **mentioned list**.

In case of successful request processing by Payment gateway OS will receive data for performing Customer browser re-direct to PG secure page.

Response parameters:

Path	Mandatory(M) / Optional(O)	Originated by Payment gateway(A) / by Store system(S)	Type	Length	Description	Example
retval	M	A	complex		PG technical	

retval	M	A	complex	parameter
retval/session	O	A	string	min 1 max 128 Unique payment session identifier
retval/paycode	O	A	string	max 14 Offline payment code
retval/redirect_url	O	A	string	Re-direct url

Example:

```

<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <registerResponse xmlns="http://www.sirena-travel.ru">
      <retval>
        <session>c857e452e31349d7b2e9a4b1738e9453</session>
        <paycode>01112223334440</paycode>
        <redirect_url>https://sandbox.egopay.ru/payments/request</redirect_url>
      </retval>
    </registerResponse>
  </soap-env:Body>
</soap-env:Envelope>

```

In case if the response timeout is reached, the Store system can resend order registration request or send one of the requests of status service, get status response and then repeat order registration request.

**Note:** Incorrect request will lead to SOAP fault in response (p. 1.3).

If order **registration** is completed successfully, Payment gateway set order status to **registered**. When the order is **registered**, Payment gateway begins order processing and changes order status to **in\_progress**. After, order status changes depends on authorization results:

1. **decline** - to **not\_authorized**
2. **accept** - to **authorized**

After **authorized** status is obtained, Payment gateway changes over order status depends on confirmation configuration (p. 1.6):

1. **auto** - order status changes to **acknowledged**, and no manual confirmation required.
2. **manual** - order status changes to **not\_acknowledged**, and order is ready to manual confirmation.

## 2.1 Register\_simple method

Service: **order**

Method: **register\_simple**

Request parameters:

Path	Mandatory(M) / Optional(O)	Type	Length	Description	Example
<b>order</b>	M	complex		Payment order	
<b>order/shop_id</b>	M	number (integer)		Store identifier	123
<b>order/number</b>	M	string	max 64	Payment order number May contain <b>unicode</b> content Case insensitive - will be converted to upper-case Can be equal to Store web order number	100500AA
<b>cost</b>	M	complex		Total cost of the order that must be charged	
<b>cost/amount</b>	M	float		Amount	100 100.00
<b>cost/currency</b>	M	string		Currency code	RUB
<b>customer</b>	O	complex		Information about Customer	
<b>customer/id</b>	O	string	min 1 max 64	Customer identifier	1
<b>customer/name</b>	O	string	max 128	Customer full name	Vasya Pupkin Вася Пупкин
<b>customer/phone</b>	O	string	max 15	Customer phone number	+79011230405
<b>customer/email</b>	O	string	max 256	Customer email address	vasya@gmail.com
<b>description</b>	O	complex		Payment order description	
<b>description/timelimit</b>	O	datetime		Time frames for Customer to perform payment If no value for timelimit provided, 15 minutes by default	2016-01-01T00:00:00+00:00

<b>description/shopref</b>	O	simple	min 1 max 64	Store system additional reference to <b>operation</b> in payment order	
<b>description/descr</b>	O	simple	min 1 max 64		additional information about the order - is <b>not used</b> in current version
<b>description/paytype</b>	O	string	min 3	Type of payment (p. 8.11) If no value is passed, PG services choose <b>card</b> type by default.	card
<b>postdata</b>	O	complex		Secure Payment gateway page settings. Possible variants of <b>value/name</b> values are described below this table.	
<b>postdata/PostEntry</b>	O	complex			
<b>postdata/PostEntry/value</b>	M	string		Secure Payment gateway page settings	
<b>postdata/PostEntry/name</b>	M	string		Secure Payment gateway page settings	

Postdata **value/name** values:

Key	Value (example)	Description
Showcase	template name or payment session type	Is used to open certain <b>customized</b> Payment page. Total list of <b>payment session types</b> is in p 8.7.
Language	en	Is used to display the correct user interface language.
Comment	your text here	Is used to add comments on Payment page.
ChosenCardType	VI	Is used to choose default payment document type.
ChosenPayMode	moto/applepay/androidpay	Card data input mode
AllowedCards	new profile name	Is used to limit the list of cards for performing payment
ReturnURLOk	https://abc124.ru/order/success/	Is used to denote the re-direct url in case of successful authorization process.
ReturnURLFault	https://abc124.ru/order/fail/	Is used to denote the re-direct url in case of unsuccessful authorization process.
ReceiptURL	https://abc124.ru/order/	Is used to denote the re-direct url after authorization process completion where the final status is displayed. Usually is used in <b>host-to-host</b> type of integration.

**Note:** The main difference between **ReturnURLOk/ReturnURLFault** and **ReceiptURL** is that **ReceiptURL** parameter has **no indicator** of order processing status. This means that Store system should define order status itself (to display it to Customer) using Payment gateway services information.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap-env:Body>
    <register_simple>
      <order>
        <shop_id>111</shop_id>
        <number>12345</number>
      </order>
      <cost>
        <currency>RUB</currency>
        <amount>1350</amount>
      </cost>
      <customer>
        <phone>+79859998877</phone>
        <name>Vasya Pupkin</name>
        <email>vasya@mail.ru</email>
      </customer>
      <description>
        <timelimit>2015-12-21T23:23:23</timelimit>
        <paytype>card</paytype>
      </description>
      <postdata>
        <PostEntry>
          <name>Language</name>
          <value>ru</value>
        </PostEntry>
        <PostEntry>
          <name>ReturnURLOk</name>
          <value>http://abc.ru</value>
        </PostEntry>
        <PostEntry>
          <name>ReturnURLFault</name>
          <value>http://abc.ru</value>
        </PostEntry>
      </postdata>
    </register_simple>
  </soap-env:Body>
</soap-env:Envelope>
```

# 3. ORDER STATUS RETRIEVE

At any time after the order was registered, Store system can retrieve order status. This functional is represented in **order** service and **status** service which include the following methods:

1. get\_status
2. get\_by\_order
3. get\_by\_order\_period
4. get\_by\_payment\_period

The description of all methods can be found below.

**Note:** There is no difference between **get\_status** and **get\_by\_order** methods responses. If OS system affords requesting only one service it use only **get\_status** method of **order** service, if several - any of methods of **status** service.

## 3.1 Get\_status method

Service: **order**

Method: **get\_status**

Request parameters:

Path	Mandatory(M) / Optional(O)	Type	Length	Description	Example
<b>order</b>	M	complex			123
<b>order/shop_id</b>	M	number (integer)		Store identifier	123
<b>order/number</b>	M	string	max 64	Payment order number in Store system	100500AA

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<soap-env:Envelope
xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap-env:Body>
    <get_status>
      <order>
        <shop_id>111</shop_id>
        <number>12345</number>
      </order>
    </get_status>
  </soap-env:Body>
</soap-env:Envelope>
```

Response parameters:

Path	Mandatory(M) / Optional(O)	Originated by Payment gateway(A) / by Store system(S)	Type	Length	Description	Example
<b>retval</b>	M	A	complex		PG technical parameter	
<b>retval/status</b>	M	A	string	min 6	Order status (p. 8.3)	acknowledged
<b>retval/order</b>	M	A	complex		Payment order	12345
<b>retval/order/shop_id</b>	M	S	number (integer)		Store identifier	123
<b>retval/order/number</b>	M	S	string	max 128	Payment order number	100500AA
<b>retval/shopref</b>	O	S	simple	min 1 max 64	Store system additional reference to operation in payment order	
<b>retval/payments</b>	O	A	complex		Information about all payments within order	
<b>retval/payments/Payment</b>	O	A	complex		Payment information	
<b>retval/payments/Payment/authorg</b>	M	A	string	max 64	Acquirer code	ucs
<b>retval/payments/Payment/authcode</b>	M	A	string	max 64	Approval code	123456

retval/payments/Payment/salepoint	O	A	string	min 1 max 64	e-cash and bill payment salepoint	22204
retval/payments/Payment/clearing	O	A	complex		Type of money settlements. Travel specific parameter.	tch
retval/payments/Payment/amount	M	S	complex		Payment amount	
retval/payments/Payment/amount/currency	M	S	string	max 3	Currency code	RUB
retval/payments/Payment/amount/amount	M	S	float		Authorized amount	100 100.00
retval/payments/Payment/doc	O	A	complex		Customer card information	
retval/payments/Payment/doc/holder	O	A	string	min 2 max 64	Card holder name	Vasya Pupkin
retval/payments/Payment/doc/code	M	A	string	max 2	Payment system card code (p. 8.12)	VI
retval/payments/Payment/doc/number	M	A	string	min 13 max 19	Primary account number (PAN) of cards	510000xx0032
retval/payments/Payment/date	M	A	datetime		Authorization date and time	2016-04-13T12:28:43
retval/payments/Payment/type	M	S	string	max 64	Payment type (p. 8.11)	card
retval/payments/Payment/id	M	A	string	max 12	Unique payment identifier. Is used to complete refund/partial refund request	616112123344
retval/error	M	A	complex		Authorization attempt error description	
retval/error/category	M	A	string	max 64	Error code (p. 8.2)	system
retval/error/code	M	A	string	max 64	Error code (p. 8.2)	ok

Example:

```

<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  <soap-env:Body>
    <get_statusResponse xmlns="http://www.sirena-travel.ru">
      <retval>
        <status>acknowledged</status>
        <order>
          <shop_id>123</shop_id>
          <number>12345</number>
        </order>
        <shopref xsi:nil="true" />
        <payments>
          <Payment>
            <authorg>ucs</authorg>
            <authcode>123456</authcode>
            <clearing>card</clearing>
            <amount>
              <amount>1350.00</amount>
              <currency>RUB</currency>
            </amount>
            <doc>
              <holder>Vasya Pupkin</holder>
              <code>CA</code>
              <number>510000*0008</number>
            </doc>
            <date>2016-07-27T09:46:18</date>
            <type>card</type>
            <id>620909280094</id>
          </Payment>
        </payments>
      </error>
    </get_statusResponse>
  </soap-env:Body>
</soap-env:Envelope>

```

```

        <category>system</category>
        <code>ok</code>
    </error>
</retval>
</get_statusResponse>
</soap-env:Body>
</soap-env:Envelope>

```

## 3.2 Get\_by\_order method

Service: **status**

Method: **get\_by\_order**

Request parameters:

Path	Mandatory(M) / Optional(O)	Type	Length	Description	Example
<b>order</b>	M	complex			123
<b>order/shop_id</b>	M	number (integer)		Store identifier	123
<b>order/number</b>	M	string	max 64	Payment order number in Store system	100500AA

Example:

```

<?xml version="1.0" encoding="utf-8"?>
<soap-env:Envelope
xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap-env:Body>
    <get_by_order>
      <order>
        <shop_id>111</shop_id>
        <number>12345</number>
      </order>
    </get_by_order>
  </soap-env:Body>
</soap-env:Envelope>

```

Response parameters:

Path	Mandatory(M) / Optional(O)	Originated by Payment gateway(A) / by Store system(S)	Type	Length	Description	Example
<b>retval</b>	M	A	complex		PG technical parameter	
<b>retval/status</b>	M	A	string	min 6	Order status (p. 8.3)	acknowledged
<b>retval/order</b>	M	A	complex		Payment order	
<b>retval/order/shop_id</b>	M	S	number (integer)		Store identifier	123
<b>retval/order/number</b>	M	S	string	max 128	Payment order number	100500AA
<b>retval/shopref</b>	O	S	simple	min 1 max 64	Store system additional reference to operation in payment order	
<b>retval/payments</b>	O	A	complex		Information about all payments within order	
<b>retval/payments/Payment</b>	O	A	complex		Payment information	
<b>retval/payments/Payment/authorg</b>	M	A	string	max 64	Acquirer code	ucs
<b>retval/payments/Payment/authcode</b>	M	A	string	max 64	Approval code	123456
<b>retval/payments/Payment/salepoint</b>	O	A	string	min 1 max 64	e-cash and bill payment salepoint	22204
<b>retval/payments/Payment/clearing</b>	O	A	complex		Type of money settlements. Travel specific parameter.	tch

retval/payments/Payment/amount	M	S	complex			Payment amount	
retval/payments/Payment/amount/currency	M	S	string	max 3		Currency code	RUB
retval/payments/Payment/amount/amount	M	S	float			Authorized amount	100 100.00
retval/payments/Payment/doc	O	A	complex			Customer card information	
retval/payments/Payment/doc/holder	O	A	string	min 2 max 64		Card holder name	Vasya Pupkin
retval/payments/Payment/doc/code	M	A	string	max 2		Payment system card code (p. 8.12)	VI
retval/payments/Payment/doc/number	M	A	string	min 13 max 19		Primary account number (PAN) of cards	510000xx0032
retval/payments/Payment/date	M	A	datetime			Authorization date and time	2016-04-13T12:28:43
retval/payments/Payment/type	M	S	string	max 64		Payment type (p. 8.11)	card
retval/payments/Payment/id	M	A	string	max 12		Unique payment identifier. Is used to complete refund/partial refund request	616112123344
retval/error	M	A	complex			Authorization attempt error description	
retval/error/category	M	A	string	max 64		Error code (p. 8.2)	system
retval/error/code	M	A	string	max 64		Error code (p. 8.2)	ok
items	O	S	complex			Denotes the list of order items. Travel specific parameters.	

Example:

```

<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <get_by_orderResponse xmlns="http://www.sirena-travel.ru">
      <retval>
        <payments>
          <Payment>
            <amount>
              <amount>1350.00</amount>
              <currency>RUB</currency>
            </amount>
            <date>2016-07-27T09:46:18</date>
            <clearing>card</clearing>
            <authcode>123456</authcode>
            <id>620909280094</id>
            <doc>
              <exp_date>2020-12-01T00:00:00</exp_date>
              <code>CA</code>
              <number>510000*0008</number>
              <token />
              <holder>Pupkin Vasya</holder>
            </doc>
            <type>card</type>
            <authorg>ucs</authorg>
          </Payment>
        </payments>
        <error>
          <code>ok</code>
          <category>system</category>
        </error>
        <order>
          <number>12345</number>
          <shop_id>111</shop_id>
        </order>
        <shopref />
        <status>acknowledged</status>
      </retval>
    </get_by_orderResponse>
  </soap-env:Body>
</soap-env:Envelope>

```

```

</get_by_orderResponse>
</soap-env:Body>
</soap-env:Envelope>

```

**Note:** Incorrect request will lead to SOAP fault in response (p. 1.3).

**get\_by\_order\_period / get\_by\_payment\_period** methods are used to get information about the orders registration or payments completion within requested timeframe respectively.

Service: **status**

Methods: **get\_by\_order\_period, get\_by\_payment\_period**

Request parameters:

Path	Mandatory(M) / Optional(O)	Type	Length	Description	Example
<b>shop_id</b>	M	number (integer)		Store identifier	123
<b>start</b>	M	datetime		date and time of the beginning of creation/payment period	2016-01-01T00:00:00+00:00
<b>stop</b>	M	datetime		date and time of the end of creation/payment period	2016-01-01T02:00:00+00:00

Example:

```

<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:egop="http://egopay.ru">
<soapenv:Header/>
<soapenv:Body>
  <egop:get_by_order_period>
    <egop:shop_id>22222</egop:shop_id>
    <egop:start>2016-01-23T10:42:00.000Z</egop:start>
    <egop:stop>2016-01-23T11:42:00.000Z</egop:stop>
  </egop:get_by_order_period>
</soapenv:Body>
</soapenv:Envelope>

```

**Max period value – 2 hours.**

Response parameters: include the list of orders, each of that consists of items described above in get\_by\_order method (see response parameters for get\_by\_order request).

**Note:** Incorrect request will lead to SOAP fault in response (p. 1.3).

All described methods of status service are used in 2 cases:

1. to retrieve orderstatus;
2. to get error code in case if no response returned on confirm/refund/reject requests.

## 4. ORDER CONFIRMATION:

After successful order processing, order status changes to **not\_acknowledged**.

Method usage rules and regulations:

1. **amount** always must be equal or less authorized amount, else Store system will receive **wrong\_amount** failure (p. 8.1).
2. **amount** may be less then authorized amount **only** if Store opened **partial confirmation** feature. Ask Support Team for it.
3. Store system may send **more than one confirm request** for one order, and if **amount** in this requests **will be equal** then Store system will receive positive response for every request.
4. If Store system will send confirm requests where **amount will not be equal** then Payment gateway will save **amount of the first processed request**, and for all another confirm requests with **another amount** Store system will receive **already\_processed** failure (p. 8.1).

**Note:** In scenario when Store system did not receive response it may resend it. Payment gateway use **shopref** parameter (is generated for each new request by Store system) to identify is Payment gateway process this request or not. If case when Payment gateway identify that request already processed, Store system will receive **already\_processed** failure (p. 8.1).

Service: **order**

Method: **confirm**

Request parameters:

Path	Mandatory(M) / Optional(O)	Type	Length	Description	Example
<b>order</b>	M	complex			
<b>order/shop_id</b>	M	number (integer)		Store identifier	123
<b>order/number</b>	M	string	max 64	Payment order number in Store system	100500AA
<b>cost</b>	M	complex		Information about the amount that must be confirmed	
<b>cost/amount</b>	M	float		Amount	100 equals 100.00

cost/currency	M	string		Currency code	RUB
shopref	O	simple	min 1 max 64	Store system additional reference to <b>operation</b> in payment order	
items	O	complex		Denotes the list of order items. Is used only for travel services sales	
tickets	O	complex		Denotes the list of issued tickets. Travel specific parameters.	

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:egop="http://egopay.ru">
<soapenv:Header />
<soapenv:Body>
  <egop:confirm>
    <egop:order>
      <egop:shop_id>22222</egop:shop_id>
      <egop:number>123-123</egop:number>
    </egop:order>
    <egop:cost>
      <egop:currency>RUB</egop:currency>
      <egop:amount>100</egop:amount>
    </egop:cost>
    <egop:shopref>90bcdcc5-d31d-401a-b9f8-35e86c31aefe</egop:shopref>
  </egop:confirm>
</soapenv:Body>
</soapenv:Envelope>
```

Response parameters: Are the same as in p. 2 (order registration response).

**Note:** Incorrect request will lead to SOAP fault in response (p. 1.3).

If the **confirm** request is completed successfully, Payment gateway changes **order status** to **acknowledged**. In case of missed order confirmation request, Payment gateway changes **order status** to **canceled**.

When Store uses manual confirmation scheme, Payment gateway changes status of every successfully payed order to **not\_authorized**. If the **confirm** request is completed successfully, Payment gateway changes **order status** to **acknowledged**. Depending on the chosen type of Payment gateway reaction on expiry time of confirmation awaiting, **order status** can be changed to:

- **confirmed** in case of **confirm-on-expire** scheme;
- **cancel** in case of **cancel-on-expire**.

## 5. ORDER CANCELTION

**Cancel** method is used to stop payment process of the order, which is already registered in Payment gateway, but isn't paid for any reason.

Method usage rules and regulations:

1. **Cancel** method can be used **only** if **order status** is **registered**.
2. Store system may send **more than one cancel request** for one order. It will receive positive response for every request.
3. If Store system sends **cancel** requests while **order status is not registered** then Payment gateway will return **already\_processed** failure (p. 8.1).

Service: **order**

Method: **cancel**

Request parameters:

Path	Mandatory(M) / Optional(O)	Type	Length	Description	Example
<b>order</b>	M	complex			
<b>order/shop_id</b>	M	number (integer)		Store identifier	123
<b>order/number</b>	M	string	max 64	Payment order number	100500AA

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:egop="http://egopay.ru">
<soapenv:Header />
<soapenv:Body>
  <egop:cancel>
    <egop:order>
      <egop:shop_id>22222</egop:shop_id>
      <egop:number>123-123</egop:number>
    </egop:order>
  </egop:cancel>
</soapenv:Body>
</soapenv:Envelope>
```

Response parameters: Are the same as in p. 2 (order registration response).

**Note:** Incorrect request will lead to SOAP fault in response (p. 1.3).

If the **cancel** request is completed successfully, Payment gateway changes **order status** to **not\_authorized**.

## 6. ORDER REJECT

**Reject** method is used to reverse payment authorization of the order, which is already paid for in Payment gateway. Only authorized payment can be reversed.

Method usage rules and regulations:

1. **Reject** method can be used **only** if **order status** is **not\_acknowledged**.
2. Store system may send **more than one cancel request** for one order. It will receive positive response for every request.
3. If Store system sends **reject** requests while **order status is not not\_acknowledged** then Payment gateway will return **already\_processed** failure (p. 8.1).

Service: **order**

Method: **reject**

Request parameters:

Path	Mandatory(M) / Optional(O)	Type	Length	Description	Example
<b>order</b>	M	complex			
<b>order/shop_id</b>	M	number (integer)		Store identifier	123
<b>order/number</b>	M	string	max 64	Payment order number	100500AA

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:egop="http://egopay.ru">
<soapenv:Header />
<soapenv:Body>
  <egop:reject>
    <egop:order>
      <egop:shop_id>22222</egop:shop_id>
      <egop:number>123-123</egop:number>
    </egop:order>
  </egop:reject>
</soapenv:Body>
</soapenv:Envelope>
```

Response parameters: Are the same as in p. 2 (order registration response).

**Note:** Incorrect request will lead to SOAP fault in response (p. 1.3).

If the **reject** request is completed successfully, Payment gateway sets **order status** to **canceled**.

## 7. PAYMENT REFUND

The Store must use **refund** method to complete the **payment refund** (after the order was confirmed and transaction process is completed).

Method usage rules and regulations:

1. Method may be used **only** after order confirmation procedure complete successfully (after using **confirm** method with positive response or authorization in auto confirmation case).
2. **amount** must be equal or less then **confirmed and not refunded** amount, else Store system will receive **wrong\_amount** failure (p. 8.1).
3. **amount** may be less then **confirmed and not refunded** amount **only** if Store opened **partial refund** feature. Ask Support Team for it.
4. Store system may send **more than one refund request** for one order, if Store opened **multiple refunds** feature. Ask Support Team for it.

**Note:** In scenario when Store system did not receive response it may resend it. Payment gateway use **shopref** parameter (is generated for each new request by Store system) to identify is Payment gateway process this request or not. If case when Payment gateway identify that request already processed, Store system will receive **already\_processed** failure (p. 8.1).

Service: **order**

Method: **refund**

Request parameters:

Path	Mandatory(M) / Optional(O)	Type	Length	Description	Example
<b>order</b>	M	complex			
<b>order/shop_id</b>	M	number (integer)		Store identifier	123
<b>order/number</b>	M	string	max 64	Payment order number	100500AA
<b>payment_id</b>	O	number (integer)	max 12	Authorization identification – <b>this parameter returns in status service responses</b> . It can be absent if the order consists of only one transaction.	1231231
<b>cost</b>	M	complex		Information about the amount that must be refunded	
<b>cost/amount</b>	M	float		Amount	100 equals 100.00

cost/currency	M	string	Currency code	RUB
<b>shopref</b>	O	simple	min 1 max 128	Store system additional reference to <b>operation</b> in payment order
<b>items</b>	O	complex		Denotes the list of order items. Travel specific parameters.

Example: Total cost of the order is 100 RUB, 100 RUB must be refunded.

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:egop="http://egopay.ru">
<soapenv:Header />
<soapenv:Body>
  <egop:refund>
    <egop:order>
      <egop:shop_id>22222</egop:shop_id>
      <egop:number>123-123</egop:number>
    </egop:order>
    <egop:payment_id>222222111111</egop:payment_id>
    <egop:cost>
      <egop:currency>RUB</egop:currency>
      <egop:amount>100</egop:amount>
    </egop:cost>
    <egop:shopref>d7715995-85ce-4465-bb1b-64f7359e3681</egop:shopref>
  </egop:refund>
</soapenv:Body>
</soapenv:Envelope>
```

Response parameters: Are the same as in p. 2 (order registration response).

**Note:** Incorrect request will lead to SOAP fault in response (p. 1.3).

If the **refund** request is completed successfully, Payment gateway sets **order status** to **refunded**.

The Store must use **refund** method to complete the **partial payment refund** (after the order was confirmed and transaction process is completed). The Store system should correct **refund/cost/amount** parameter to perform partial refund.

Service: **order**

Method: **refund**

Request parameters: Are the same as for **refund** method.

Example: Total cost of the order is 100 RUB, 60 RUB must be refunded.

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:egop="http://egopay.ru">
<soapenv:Header />
<soapenv:Body>
  <egop:refund>
    <egop:order>
      <egop:shop_id>22222</egop:shop_id>
      <egop:number>123-123</egop:number>
    </egop:order>
    <egop:payment_id>222222111111</egop:payment_id>
    <egop:cost>
      <egop:currency>RUB</egop:currency>
      <egop:amount>60</egop:amount>
    </egop:cost>
    <egop:shopref>d7715995-85ce-4465-bb1b-64f7359e3681</egop:shopref>
  </egop:refund>
</soapenv:Body>
</soapenv:Envelope>
```

Response parameters: Are the same as in p. 2 (order registration response).

**Note:** Incorrect request will lead to SOAP fault in response (p. 1.3).

If the **refund** request is completed successfully, Payment gateway sets **order status** to **refunded**.

## 8. REFERENCES

### 8.1. REQUEST PROCESSING SOAP FAULT CODES

Error code	Description	Action expected
ALREADY_PROCESSED	order number is already registered in Payment gateway	Check and correct order number Resend registration request with new order number.
ACCESS_DENIED	unsuccessful Store authentication	Check credentials and shop_id; Resend request.
SYSTEM_ERROR	system malfunction	Resend request later; Contact Acquirer Support Team.
		Check request accuracy;

SYSTEM_ERROR	incorrect request performance	Resend request.
ORDER_ERROR	order not found in service provider system	Check number or host parameters; Resend request.
INVALID_ORDER	such order number is not registered in Payment gateway	Check request Resend request
PRICING_ERROR	mismatched good/service price in Store system and in service provider system	Check relevant good/service price; Resend request.
BOOKING_ERROR	incorrect booking status in GDS system	Check booking status and items parameter; Correct and resend request
WRONG_AMOUNT	incorrect amount parameter	Check order cost, correct it; Resend request.
SESSION_ERROR	GDS session receipt fail	Resend request
MAX_ATTEMPTS	number of allowed payment attempts is exceeded	Pay attention to Customer
REPEAT_NEEDED	request processing fail	Resend request later

## 8.2. ORDER PROCESSING ERRORS CODES

ID	CATEGORY	CODE	TCODE	Cause	Solution
0	system	ok	OK	Operation is completed successfully	
501	system	black-list	BL	Customer card is blacklisted	Contact Acquirer Support Team to receive updated antifraud settings.
998	system	max_attempts	LE	number of payments attempts is exceeded	Recommend Customer to appeal to issue bank (of his card) support.
999	system	in_progress	IN	order is still processing	Check order status after a while and repeat the request if needed
					Repeat the request within the frames of established timelimit.
102/103	system	network	CE		In case if the problem remains (for example 30% of orders registered within 30 minutes remain unpaid because of system/common error) contact Acquirer Support Team.  ! Please, be able to provide us with the examples of such orders.
670	system	fraud	FM	operation is declined by Payment gateway anti-fraud system	Store system support should pay attention to Customer, his order cart and card
776	system	operator	RE	authorization process was interrupted by Backoffice operator	
					Repeat the request within the frames of established timelimit.
333	system	common	SE	services malfunction	In case if the problem remains for example 30% of orders registered within 30 minutes – remain unpaid because of system/error error – contact Acquirer Support Team.  ! Please, be able to provide us with the

						examples of such orders.
334	system	timeout	TE	processing timeout		Check order status; Resend request.
335	system	session	SE	Payment gateway is unable to process the request		Resend request; If the problem remains - contact Acquirer Support Team.
336	system	denied	SD	payment can't be performed via this card		Check payment document.
400	system	already_processed	AP	the order is already processed		Check request
681	system	unavailable_method	SD	Incorrect service/method combination		Check request
12/13/57/58	bank	i-prohibition	TD	online payments are restricted for this card		Recommend Customer to appeal to issue bank (of his card) support.
4/7/41/43	bank	fraud	FM	operation is declined by Bank anti-fraud system		Store system support should pay attention to Customer, his order cart and card.
666/5	bank	common	CE	common/indeterminate error		Repeat the request within the frames of established timelimit.  In case if the problem remains for example 30% of orders registered within 30 minutes – remain unpaid because of bank/common error – contact Acquirer Support Team.  ! Please, be able to provide us with the examples of such orders.
96/82/101	bank	network	NE	Acquirer's system is unavailable		Repeat the request within the frames of established timelimit.  In case if the problem remains for example 30% of orders registered within 30 minutes – remain unpaid because of bankerror – contact Acquirer Support Team.  ! Please, be able to provide us with the examples of such orders.
51	bank	funds	IF	insufficient funds for order processing		Recommend Customer to appeal to issue bank (of his card) support or check card balance via bank online resources.
54/678	bank	account	IE	Customer entered incorrect card data		Check the accuracy of entered card data, notify Customer about the mistake.
61	bank	limit	LE	operations limit over card is exceeded		Recommend Customer to appeal to issue bank (of his card) support.
674	bank	timeout	TE	order processing timeout		Check order status; Advice Customer to repeat payment attempt.
667	3dsecure	reject	UE	Customer closed or refreshed secure bank page		
668	3dsecure	failed	AE	Customer entered incorrect secure bank code		

673/672	3dsecure	timeout	TE	3ds verification processing timeout	
6731/6721	3dsecure	network	NE	3ds verification processing timeout	
6690	shop	common	OE	incorrect shop_id settings	Contact Acquirer Support Team.
6691	shop	price	OE	incorrect order cost	Check total order cost and cost of every order item.
6692	shop	status	OE	the order is already processed (paid or canceled)	Check order number and current order status.
6693	shop	network	CE	order processing timeout	Check order status; Resend request.
6694	shop	commission	OE	incorrect order commission cost	Check and correct commission cost in request to GDS.
6695	shop	invalid_fop	OE	incorrect form of payment	Contact Acquirer Support Team.
6696	shop	wrong_amount	OE	incorrect value of amount parameter	Check "Amount" parameter in request.
6700/6701/6702/6703/6704/6705/6706/6707/6708/6709/6710/6711	shop	error	OE		
6712	shop	session	OE	Payment gateway and Store system communication failure	Contact Acquirer Support Team.
671	shop	network	OE	Store system is unavailable	Check the availability of Store system.
886	shop	common	FE	common/indeterminate error	Contact Acquirer Support Team.
777	shop	cancel	RE	order processing was canceled by Store system	
675	user	timeout	TE	Customer ran out of time to perform payment	Check order timelimit. We recommend to set timelimit no less than 5-7 min – according to our statistics it is average time the customer spend on making payment.
676	user	cancel	UE	Customer clicked the button “Back” / “Cancel payment” or Customer ‘refreshed’ the payment page	We recommend to notify customer not to leave page while the payment is processing and not to refresh it.
677	user	duplicate	UE	Store system sent several equal order registration request simultaneously	Check Online store settings.
679	user	card_not_binded	IE	incorrect token/customer id	Check request; Resend request.
680	user	unsupported_card	IE	payment can't be performed via this card	Check payment document.
682	user	network	UN	network connection fail	Customer connection failed during 3ds secure authentication process.

## 8.3.ORDER STATUS

Status	Description	Action expected
registered	order is registered in Payment gateway system	no action expected
in_progress	order processing, authorization response is expected	no action expected

authorized	successful authorization, good/service receipt formation	no action expected
not_authorized	unsuccessful order processing	Store system can: perform no request; create new order; initiate new payment session
canceled	authorization is canceled	no action expected
failed	unsuccessful good/service receipt formation	Store should check availability of good/service, then: either confirm it manually via Backoffice system or cancel it manually via Backoffice system.
not_acknowledged	unsuccessful order confirmation or unsuccessful good/service receipt formation	Send confirm or reject request.  In case of unsuccessful good/service receipt formation - no action expected.
acknowledged	successful order processing, successful receipt formation and successful order confirmation	no action expected
refunded	successful payment refund	no action expected

## 8.4 EVENT CODES

Code	Description
register	order registration in PG system
change	order content has changed
accept	order processing
reject	payment reject
acknowledge	order acknowledgement
refund	payment refund
cancel	order cancelation
partial_cancel	partial order cancelation
attempt	unsuccessful authorization attempt
check	order content check
notify	PG services auto notification
payment_cancel	payment cancelation
payment_change	order cost has changed
charge	payment write-off
subscription_create	recurrent payments schedule creation
subscription_activate	recurrent payments schedule activation
subscription_cancel	recurrent payments schedule cancelation

## 8.5 3DS AUTHENTICATION STATUS

Status	Description
attempted	3ds secure code input attempt
unavailable	3ds check is unavailable
started	3ds check is in process - card is enrolled
created	3ds check is in process - re-direct to secure bank page
unprovided	card doesn't support 3ds check mechanism
authenticated	successful authentication
ready	3ds check is in process
not_used	3ds check wasn't used (in case of non-3ds scheme of work)

not_determined	card's bank hasn't sent the enrolment response
not_authenticated	unsuccessful authentication
not_enrolled	card is not enrolled

## 8.6 User interface language

Abbreviation	Description
ru	Russian
en	English
de	German
cn	Chinese

## 8.7 Payment session types

Abbreviation	Description
redirect	Re-direct to Payment gateway secure page
iframe	iframe format adapted secure Payment gateway page opening
mobile	mobile format adapted secure Payment gateway page opening
rest	host-to-host scheme of card data transfer
token	recurring payment performance

## 8.8 Card codes

Abbreviation	Description
VI	Visa
CA	MasterCard
EU	Maestro
MR	MIR
CP	Union Pay

## 8.9 Card data input modes (**ChosenPayMode**)

Abbreviation	Description
moto	Mail Order Telephone order
applepay	Apple Pay App functional
androidpay	Android Pay App functional