

Express.ru personal account API

Version dated 19.05.2015

Table of contents

Express.ru personal account API.....	1
API functions and their parameters.....	3
Creating a purchase order and related invoices.....	4
Creating invoices to a previously created purchase order.....	6
Getting code-value pairs for possible order statuses.....	7
Getting code-value pairs for possible invoices statuses.....	8
Obtaining the orders history for a period.....	9
Obtaining invoice data by number.....	10
Obtaining order data by number.....	12
Tracking order/invoice status.....	14
Shipping cost calculation.....	15

In order to use personal account's APIs at the Express.ru website, you have to obtain 2 keys, i.e., the authorization key, and the signature key. The keys identify the client and all requests made using these keys will be carried out on your behalf.

Do not pass the key data to anyone!

The authorization key is sent with each request to API. The authorization key is transferred via HTTP request's Authorization header.

Each request to the API should be signed. The signature is passed in the sig parameter of the query string.

The signature should be generated by the following algorithm.

md5 (client code. Query URL. GET params . POST params . signature key);

The «params» value is a concatenation of "name=value" pairs sorted in the alphabet order by "name", where "name" is the name of the parameter passed to the API function, and "value" is the value of the parameter. Separators are not used in concatenation. The «sig» parameter is ignored in signature calculation, while all other query parameters should be used in calculation.

An example of a PHP function which returns the value of the signature.

```
function generate_signature($username, $signature_secret_key, $url, array
$get_array, array $post_array)
{
    $concat_query = $concat_post = '';
    ksort($get_array);
    foreach ($get_array as $key => $value) {
        $concat_query .= $key . '=' . $value;
    }
    ksort($post_array);
    foreach ($post_array as $key => $value) {
        $concat_post .= $key . '=' . $value;
    }
    return md5($username . $url . $concat_query . $concat_post .
$signature_secret_key);
}
```

API functions and their parameters

When API is used, the query string contains the parameters required for retrieving the parameters of data or functions .

The data required for creating new orders, invoices or other records are sent using the **POST** method in the **data** parameter.

The **data** in the data parameter should be represented in **JSON** format.

JSON in the data parameter should be passed in a single line, without formatting.

If case of global errors and exceptions, all functions return an error in a single format.

```
{"status":"error","error":true,"errors":{"global_error":"Invalid authorization key"},"result":null}
```

Creating a purchase order and related invoices

http://www.express.ru/api/office/new_order_w_invoices

Query string parameters:

sig*	Query signature
-------------	-----------------

In the query string, the function receives only the value of the signature.

The data needed for creating an order and invoices are sent in **POST data** parameter in **JSON** format.

Order data are mandatory. Invoice data may be omitted. One or more invoices to the order may be passed in the array of the invoices. Fields marked with * are mandatory.

JSON schema for function

```
{
  "order": {
    "sampl_date": "*date of invoice collection (only in the dd.mm.yyyy or dd/mm/yyyy format)",
    "sender": "*the Sender's",
    "geo_sender": "*the Sender geography (the name of city or country (in case of shipping around the world))",
    "address_sender": "* Sender's address",
    "contact_sender": "*Sender's contact person",
    "phone_sender": "*Sender's phone number",
    "recipient": "*the Recipient",
    "geo_recipient": "*Recipient's geography (the name of city or country (in case of shipping around the world))",
    "address_recipient": "*Recipient's address",
    "contact_recipient": "*Recipient's contact person",
    "phone_recipient": "*Recipient's phone number",
    "comment": "*Shipment description",
    "gruz_mest": "*Number of packages",
    "gruz-ves": "*Weight",
    "gruz-ob-ves": "Volumetric weight",
    "dop_comment": "Note",
    "contact": "Customer's contact",
    "payer": "*Payer (customer, recipient, sender)",
    "payment": "*Payment method (bank transfer, invoice, website, mini terminal, cash)"
  },
  "invoices": [
    {
      "recipient": "*the Recipient",
      "geo_recipient": "*Recipient's geography (the name of city or country (in case of shipping around the world))",
      "address_recipient": "*Recipient's address",
      "contact_recipient": "*Recipient's contact person",
      "phone_recipient": "*Recipient's phone number",
      "comment": "*Shipment description",
      "gruz_mest": "*Number of packages",
      "gruz-ves": "*Weight",
      "gruz-ob-ves": "Volumetric weight",
      "dop_comment": "Note",
      "contact": "Customer's contact",
      "payer": "*Payer (customer, recipient, sender)",
      "payment": "*Payment method (bank transfer, invoice, website, mini terminal, cash)",
      "cashOnDelivery": "CoD amount (for online stores)"
    },
    {
      "recipient": "*the Recipient",
      "geo_recipient": "*Recipient's geography (the name of city or country (in case of shipping around the world))",
      "address_recipient": "*Recipient's address",
      "contact_recipient": "*Recipient's contact person",
      "phone_recipient": "*Recipient's phone number",
      "comment": "*Shipment description",
      "gruz_mest": "*Number of packages",
      "gruz-ves": "*Weight",
      "gruz-ob-ves": "Volumetric weight",
      "dop_comment": "Note",
      "contact": "Customer's contact",
      "payer": "*Payer (customer, recipient, sender)",
      "payment": "*Payment method (bank transfer, invoice, website, mini terminal, cash)",
      "cashOnDelivery": "CoD amount (for online stores)"
    }
  ]
}
```

Examples of function replies

An example of reply after successful creation of an order and 2 related invoices

```
{"status":"ok","error":false,"result":{"order_number":"Web-0000300171","invoices_numbers":["WEB00000000000000162","WEB00000000000000163"]}}
```

An example of reply when an order and one invoice have been created, the second invoice has not passed validation

```
{"status":"some_errors","error":true,"errors":{"invoices_errors":{"invoice1":{"invoice_validation_errors":{"errors":{"payer":"The payer field may list: customer, recipient, sender"}}}}},"result":{"order_number":"Web-0000300173","invoices_numbers":["WEB00000000000000164"]}}
```

An example of reply when there is an error in the order data, invoices data were not sent (order has not been created)

```
{"status":"error","error":true,"errors":{"order_validation_errors":{"payer":"The payer field may list: customer, recipient, sender"}},"result":{"order_number":null,"invoices_numbers":[]}}
```

An example of reply when there is an error in the order and the invoice data (order and invoices have not been created)

```
{"status":"error","error":true,"errors":{"order_validation_errors":{"gruz_mest":"The number of packages must be a digit"},"invoices_errors":{"invoice1":{"invoice_validation_errors":{"errors":{"payer":"The payer field may list: customer, recipient, sender"}}}}},"result":{"order_number":null,"invoices_numbers":[]}}
```

Creating invoices to a previously created purchase order

http://www.express.ru/api/office/new_invoices_for_order

Query string parameters:

order_number*	The number of the order
sig*	Query signature

The data needed for creating invoices are sent in **POST data** parameter in **JSON** format.

One or more invoices to the order may be passed in the array of the invoices. Fields marked with * are mandatory.

JSON schema for function

```
{
  "invoices": [
    {
      "recipient": "**the Recipient",
      "geo_recipient": "**Recipient's geography (the name of city or country (in case of shipping around the world))",
      "address_recipient": "**Recipient's address",
      "contact_sender": "**Recipient's contact person",
      "phone_recipient": "**Recipient's phone number",
      "comment": "**Shipment description",
      "gruz_mest": "**Number of packages",
      "gruz-ves": "**Weight",
      "gruz-ob-ves": "Volumetric weight",
      "dop-comment": "Note",
      "contact": "Customer's contact",
      "payer": "**Payer (customer, recipient, sender)",
      "payment": "**Payment method (bank transfer, invoice, website, mini terminal, cash)"
    },
    {
      "recipient": "**the Recipient",
      "geo_recipient": "**Recipient's geography (the name of city or country (in case of shipping around the world))",
      "address_recipient": "**Recipient's address",
      "contact_sender": "**Recipient's contact person",
      "phone_recipient": "**Recipient's phone number",
      "comment": "**Shipment description",
      "gruz_mest": "**Number of packages",
      "gruz-ves": "**Weight",
      "gruz-ob-ves": "Volumetric weight",
      "dop-comment": "Note",
      "contact": "Customer's contact",
      "payer": "**Payer (customer, recipient, sender)",
      "payment": "**Payment method (bank transfer, invoice, website, mini terminal, cash)"
    }
  ]
}
```

The answer is similar to the reply in the `new_order_w_invoices` method in the block related to invoices.

Getting code-value pairs for possible order statuses

http://www.express.ru/api/office/get_available_order_statuses

Query string parameters:

sig*	Query signature
-------------	-----------------

Examples of function reply

```
{
  "status": "ok",
  "error": false,
  "result": {
    "statuses": [{
      "code": "00006",
      "text": "Shipment collected"
    }, {
      "code": 11,
      "text": "A courier has been assigned"
    }, {
      "code": 8,
      "text": "Processing order"
    }, {
      "code": "00005",
      "text": "Invoice created"
    }, {
      "code": "00002",
      "text": "Accepted"
    }, {
      "code": "00003",
      "text": "Approved"
    }, {
      "code": "NRSM1",
      "text": "Under consideration"
    }
  ]
}
```

Getting code-value pairs for possible invoices statuses

http://www.express.ru/api/office/get_available_invoice_statuses

Query string parameters:

sig*	Query signature
-------------	-----------------

Examples of function reply

```
{
  "status": "ok",
  "error": false,
  "result": {
    "statuses": [{
      "code": "00005",
      "text": "The shipment returned"
    }, {
      "code": "00009",
      "text": "The shipment has left storage location"
    }, {
      "code": "00003",
      "text": "The shipment has been delivered"
    }, {
      "code": 12,
      "text": "The shipment has been passed for delivery"
    }, {
      "code": "00011",
      "text": "The shipment has been received by the agent"
    }, {
      "code": "00010",
      "text": "The shipment has arrived to the storage location"
    }, {
      "code": 33,
      "text": "Hold for pickup"
    }, {
      "code": "00007",
      "text": "Consolidation of shipment"
    }, {
      "code": "00002",
      "text": "Invoice created"
    }, {
      "code": 34,
      "text": "Customer pickup"
    }, {
      "code": "00006",
      "text": "Trace"
    }
  ]
}
```


Obtaining the orders history for a period

http://www.express.ru/api/office/get_history

Query string parameters:

date_from	Period start date
date_to	Period end date
orderStatuses[]	Filter by codes of orders statuses
invoiceStatuses[]	Filter by codes of invoices statuses
sig*	Query signature

All parameters except for the signature are optional. If the period is not specified, orders for the last 30 days are displayed.

Examples of function reply

```
{
  "status": "ok",
  "error": false,
  "result": {
    "orders": [ //Array of orders
      {
        "order_number": "Web-0002102755", //Order number
        "date": "16.01.2014", //Date of ordering
        "sender": "Sender",
        "status": "Under consideration", //Current order status
        "invoices": [ //Invoices array to the order
          {
            "invoice_number": "WEB-0000073", //Invoice number
            "date": "16.01.2014", //Date of invoice creation
            "recipient": "Recipient",
            "sum": "1.18", //Invoice amount
            "weight": "1", //Shipment weight
            "volume_weight": "1", //Shipment volumetric weight
            "urgency": "Scooters", //Urgency
            "payer": "Sender",
            "status": "Shipment delivered" //Current invoice status
          },
          {
            "invoice_number": "WEB-0000074",
            "date": "16.01.2014",
            "recipient": "Recipient",
            "sum": "1025",
            "weight": "1",
            "volume_weight": "1",
            "urgency": "Scooters",
            "payer": "Sender",
            "status": "Shipment delivered"
          }
        ]
      }
    ]
  }
}
```

Obtaining invoice data by number

http://www.express.ru/api/office/get_invoice_data

Query string parameters:

invoice_number*	Invoice number
sig*	Query signature

In the query string, the function receives the invoice number and the value of the signature.

Examples of function reply:

```
{
  "status": "ok", //query status
  "error": false, //Presence of an error
  "result": {
    "number": "WEB-0000099", //Invoice number
    "date": "23.01.2014 22:25:14", //Date of invoice creation
    "docosn": "Web-0002105118", //Order number
    "sender": "Sender",
    "geosender": "Moscow", //City of origin
    "geosendercode": "0000172949", //Code of the city of origin
    "geosendercodekladr": "780000000000000000000000", //Code of the city of origin acc. to the All-Russian Classifier
of Addresses
    "landsender": "RUSSIA", //Country of origin
    "landsendercode": "0000000001", //Code of the country of origin
    "senderaddress": "Sender's address",
    "senderphone": "+0000000000", //Sender's phone number
    "sendercontact": "Sender's contact",
    "sendercomment": "", //Sender's comment
    "recipient": "Recipient",
    "georecipient": "Moscow", //City of collection
    "georecipientcode": "0000172949", //Code of the city of collection
    "geosendercodekladr": "780000000000000000000000", //Code of the city of collection acc. to the All-Russian
Classifier of Addresses
    "landrecipient": "RUSSIA", //Country of collection
    "landrecipientcode": "0000000001", // Code of the country of collection
    "recipientaddress": "Recipient's address"
    "recipientphone": "+0000000000", //Recipient's phone number
    "recipientcontact": "Recipient's contact person",
    "recipientcomment": "", //Sender's comment
    "comment": "", //Comment to the shipment
    "status": "00002", //Current status of the invoice
    "satdelivery": false, //Saturday delivery
    "sundelivery": false, //Sunday delivery
    "byhand": false, //Hand delivery
    "urgency": "Express", //Urgency
    "urgent": "5", //Urgency code
    "doonly": "", // Documents only
    "gruzvid": "000000002", //Shipment type
    "pickup": "0", //Customer pickup
    "return": "0", //To be returned
    "reception": "0", //Shipment collected in the office
    "gruzinfo": "Shipment description",
    "mest": 1, //Number of packages
    "ves": "1", //Weight
    "obves": "0", //Volumetric weight
```

```
"smsinfo": false, //Inform via SMS
"smsnumber": "", //Number for SMS
"mailinfo": false, //Inform via email
"mail": "", //email address to notifications
"payment": "One-time account", //Payment method
"payer": "Customer", //Payer
"payername": "Test group", // Payer name
"gruzprice": "0", //Declared value of the shipment
"fragility": "0", //Fragile shipment
"inspay": "0", //Insurance fee
"sum": "1.18", //Invoice amount
"sumDetails": { //Invoice amount breakdown. Available for online stores only
  "services": {
    "Other services": [
      {
        "serviceName": "Delivery from Moscow to the Moscow region",
        "serviceSum": "1,18"
      }
    ]
  },
  "fullSum": "1,18" //Total invoice amount
}
}
```

Obtaining order data by number

http://www.express.ru/api/office/get_order_data

Query string parameters:

order_number*	The number of the order
sig*	Query signature

In the query string, the function receives the order number and the value of the signature.

Examples of function reply:

```
{
  "status": "ok", //query status
  "error": false, //Presence of an error
  "result": {
    "number": "WEB0000000", //Order number
    "date": "23.01.2014 22:25:14", //Date of ordering
    "sender": "Sender",
    "geosender": "Moscow", //City of origin
    "geosendercode": "0000172949", //Code of the city of origin
    "geosendercodekladr": "780000000000000000000000", //Code of the city of origin acc. to the All-Russian Classifier
of Addresses
    "landsender": "RUSSIA", //Country of origin
    "landsendercode": "0000000001", //Code of the country of origin
    "senderaddress": "Sender's address",
    "senderphone": "+0000000000", //Sender's phone number
    "sendercontact": "Sender's contact",
    "sendercomment": "", //Sender's comment
    "recipient": "Recipient",
    "georecipient": "Moscow", //City of collection
    "georecipientcode": "0000172949", //Code of the city of collection
    "geosendercodekladr": "780000000000000000000000", //Code of the city of collection acc. to the All-Russian
Classifier of Addresses
    "landrecipient": "RUSSIA", //Country of collection
    "landrecipientcode": "0000000001", // Code of the country of collection
    "recipientaddress": "Recipient's address",
    "recipientphone": "+0000000000", //Recipient's phone number
    "recipientcontact": "Recipient's contact person",
    "recipientcomment": "", //Sender's comment
    "comment": "", //Comment to the shipment
    "status": "00002", //Current status of the invoice
    "satdelivery": false, //Saturday delivery
    "sundelivery": false, //Sunday delivery
    "byhand": false, //Hand delivery
    "urgency": "Express", //Urgency
    "gruzvid": "000000002", //Shipment type
    "reception": "0", //Shipment collected in the office
    "gruzinfo": "Shipment description",
    "mest": 1, //Number of packages
    "ves": "1", //Weight
    "obves": "0", //Volumetric weight
    "smsinfo": false, //Inform via SMS
    "smsnumber": "", //Number for SMS
    "mailinfo": false, //Inform via email
    "mail": "", //email address to notifications
  }
}
```

Moscow, tel./fax (495) 231-21-21
Saint-petersburg, tel./fax (812) 603-22-22
Volgograd, Kazan, Krasnodar, Nizhny Novgorod,
Tver: 8-800-505-08-08

express 

ЭКСПРЕСС-ДОСТАВКА ДОКУМЕНТОВ И ГРУЗОВ

```
"payment": "One-time account", //Payment method  
"payer": "Customer", //Payer  
"gruzprice": "0", //Declared value of the shipment  
"inspay": "0", //Insurance fee
```

```
}  
}
```

Tracking order/invoice status

http://www.express.ru/api/office/get_tracking

Query string parameters:

track_code *	The number of the invoice or the order, multiple numbers may be indicated, comma-separated
date	Date in a standard format, time may be specified after a space (hh:mm:ss). The statuses before this date will not be included into the selection.
sig *	Query signature

An example query answer:

```
{
  "status":"ok",
  "error":false,
  "result":{
    "S477354":[
      {
        "date":"14.03.2013 9:23:48",
        "status":"The shipment has been sent to the courier for delivery. Please await delivery",
        "text": "The shipment has been sent for delivery" Courier: Aliyeva Muslimat Nabievna"
      },
      {
        "date":"15.03.2013 9:23:29",
        "status": "The shipment has been delivered to the recipient"
        "note":"Knhykina (14.03.2013 15:40:00)"
      }
    ],
    "S386823":[
      {
        "date":"14.03.2013 9:17:08",
        "status":"The shipment has been sent to the courier for delivery. Please await delivery",
        "text": "The shipment has been sent for delivery" Courier: Ivanov Igor Ivanovich 89162631461 \u211624"
      },
      {
        "date":"14.03.2013 21:32:50",
        "status": "The shipment has been delivered to the recipient"
        "note":"Scherkunov (14.03.2013 16:20:00)"
      }
    ]
  }
}
```

Shipping cost calculation

<http://www.express.ru/api/office/calculate>

Query string parameters:

country_from *	<i>(string) Country of origin</i>
place_from *	<i>(string) Place of origin</i>
country_to *	<i>(string) Country of destination</i>
place_to *	<i>(string) Place of destination</i>
cargo_type *	<i>(int) Type of shipment. 1 — oversize shipment, 2 — shipment, 3 — documents</i>
weight *	<i>(float) Weight, kg.</i>
cargo_form *	<i>(int) Form of shipment 1 — tube, 2 — carton, 3 — roll</i>
box_length	<i>(float) Box length</i>
box_width	<i>(float) Box width</i>
box_height	<i>(float) Box height</i>
rulon_radius	<i>(float) Roll radius</i>
rulon_length	<i>(float) Roll length</i>
tubus_length	<i>(float) Tube side</i>
tubus_side_a	<i>(float) Tube side</i>
tubus_side_b	<i>(float) Tube side</i>
tubus_side_c	<i>(float) Tube side</i>
Items	<i>(int) Number of packages in the shipment</i>
Fragile	<i>(bool) Fragile shipment. 50% markup</i>
sig*	Query signature

Enter the points of origin and destination acc. to the All-Russian Classifier of Addresses, indicating the type of the settlement after its name. Cities, such as "Moscow city", can be entered without the type of settlement, like "Moscow".

An example query answer:

```
{
  "status":"ok",
  "error":false,
  "result":[
    {
      "id":"2626",
      "gid":"defaults",
      "group":"global",
      "name":"Basic",
      "from":"Moscow",
      "to":"SPB",
      "weightId":"9",
      "type":"4",
      "docType":"1",
      "price":"550 rubles",
      "priceKG":"0.00",
      "weightFrom":"0.000",
      "weightTo":"1.000",
      "weightStep":"0.000",
      "typeLabel":"Office - to - office",
      "dayFrom":"0",
      "dayTo":"0",
      "dayComment":"next day delivery",
```

```
"deliveryTime":"next day delivery",
"weight":"0.5",
"rawPrice":550,
"endPointPayment":true
},
{
  "id":"2784",
  "gid":"defaults",
  "group":"global",
  "name":"Basic",
  "from":"Moscow",
  "to":"SPB",
  "weightId":"9",
  "type":"1",
  "docType":"1",
  "price":"620 rubles",
  "priceKG":"0.00",
  "weightFrom":"0.000",
  "weightTo":"1.000",
  "weightStep":"0.000",
  "typeLabel":"Standard",
  "dayFrom":"0",
  "dayTo":"0",
  "dayComment":"next day delivery",
  "deliveryTime":"next day delivery",
  "weight":"0.5",
  "rawPrice":620,
  "endPointPayment":true
},
{
  "id":"2768",
  "gid":"defaults",
  "group":"global",
  "name":"Basic",
  "from":"Moscow",
  "to":"SPB",
  "weightId":"9",
  "type":"2",
  "docType":"1",
  "price":"1090 rubles",
  "priceKG":"0.00",
  "weightFrom":"0.000",
  "weightTo":"1.000",
  "weightStep":"0.000",
  "typeLabel":"Urgent",
  "dayFrom":"0",
  "dayTo":"0",
  "dayComment":"by certain time",
  "deliveryTime":"by certain time",
  "weight":"0.5",
  "rawPrice":1090,
  "endPointPayment":true
}
]
}
```


Obtaining the array of countries and cities

http://www.express.ru/api/office/get_countries_and_cities

Query string parameters:

sig*	Query signature
-------------	-----------------

Examples of function reply

```
[
  {
    "uuid": "0000176677",
    "country": "AUSTRIA"
  },
  {
    "uuid": "0000174847",
    "country": "AZERBAIJAN",
    "cities": [
      {
        "id": "0000176904",
        "name": "Absheron"
      },
      {
        "id": "0000174848",
        "name": "Agdam"
      }
    ]
  }
]
```

www.express.ru