

Content

1. Information and instructions for use.....	2
1.1. Functionalities of the API for integration.....	2
1.2. Format of the API integration.....	2
1.3. URL from which requests are sent and user accounts for authentication	2
1.4. Validation of addresses, settlements, Econt's offices.....	2
2. Sample requests.....	3
2.1 Example for generating a request for shipment in e-Econt	3
2.2 Example with information about the serviced settlements.....	4
2.3 Example with information for the Econt's offices	4
2.4 Example with information about the client's profile with the provided account.....	4
2.5 Example with information about the settlements' streets.....	4
2.6 Example for sending a request to the API and how PHP arrays get converted to and from XML	5

1. Information and instructions for use

1.1. Functionalities of the API for integration

- Calculating the price of a shipment
- Generating a shipment request in e-Econt and generating the PDF for printing, validating of the shipment's data
- Order a courier
- Checking the status of a shipment and cash on delivery, tracking of a shipment
- Information about Econt's offices
- Information about the settlements serviced by Econt
- Information about the streets and quarters in the settlements
- Information about the e-Econt user and their addresses for sending and receiving a shipment

1.2. Format of the API integration

The requests and responses of the API are in XML format. The full description of the content's structure can be found here.

1.3. URL from which requests are sent and user accounts for authentication

- requests for price calculation, preparing of shipments, money transfers, courier requests:

* demo stage - http://demo.econt.com/e-econt/xml_parcel_import2.php

* production stage - http://www.econt.com/e-econt/xml_parcel_import2.php

- all other types of requests:

* demo stage - http://demo.econt.com/e-econt/xml_service_tool.php

* production stage - http://www.econt.com/e-econt/xml_service_tool.php

XML requests are sent through http/https POST by one of the following ways:

- as an attached file with Content-type: multipart/form-data (the same as attached file from a browser's form)
- as POST parameter named "xml" and Content-type: application/x-www-form-urlencoded (the same as a field with name "xml" from browser's form)
- directly as a content of the POST request with Content-type: text/xml

The account for the demo stage is: username – demo / password – demo

For production stage an existing e-Econt account may be used or you can register a new one at https://ee.econt.com/load_direct.php?target=Register . You may also contact an Econt's employee.

Authorization through SSL client certificates is not currently available, but SSL connection may be used with standard authorization (in the XML request).

1.4. Validation of addresses, settlements, Econt's offices

The valid names of settlements, streets and quarters can be found by API request to http://www.econt.com/e-econt/xml_service_tool.php

- settlements and post codes: `<request_type>cities</request_type>`

- offices: `<request_type>offices</request_type>`

- quarters: `<request_type>cities_quarters</request_type>`

- streets: `<request_type>cities_streets</request_type>`

It is also possible for the client to choose the office from a map, using the Econt's office locator.

A minimum example implementation of how to choose an office with the office locator can be found at: http://ee.econt.com/office_locator_example.html

When preparing a shipment, the sender's address, the receiver's address and third party's address (for

receiving the sum of cash on delivery, shipment return, etc) are necessary to be valid according to the following criteria:

1. Valid post code and settlement's name
2. At least one of the following:
 - Valid street name(tag <street>) and number(tag <street_num>)
 - Valid quarter name(tag <quarter>) and value of the "other" field(tag <street_other>)
 - For settlements that do not have streets data: street name without validation(tag <street>)

and number(tag <street_num>)

- For shipments upon request: valid Econt office code(tag <office_code>)

In order to calculate delivery price of a shipment, it is enough to provide a valid settlement and post code for receiver and sender.

2. Sample requests

2.1 Example for generating a request for shipment in e-Econt

```
<?xml version="1.0" encoding="UTF-8"?>
<parcels>
  <client>
    <username>demo</username>
    <password>demo</password>
  </client>
  <system>
    <validate>0</validate>
    <response_type>XML</response_type>
    <only_calculate>0</only_calculate>
  </system>
  <loadings>
    <row>
      <sender>
        <city>Ruse</city>
        <post_code>7000</post_code>
        <office_code>7000</office_code>
        <name>Иван Иванов</name>
        <name_person></name_person>
        <phone_num>0888888888</phone_num>
      </sender>
      <receiver>
        <city>Русе</city>
        <post_code>7000</post_code>
        <name>Петър Петров</name>
        <phone_num>0888888888</phone_num>
        <sms_no/>
        <street>Славянска</street>
        <street_num>16</street_num>
      </receiver>
      <shipment>
        <envelope_num>11111,22222,3332342</envelope_num>
        <shipment_type>PACK</shipment_type>
        <description>description of the content content</description>
        <pack_count>3</pack_count>
        <weight>1</weight>
        <tariff_sub_code>OFFICE_DOOR</tariff_sub_code>
        <invoice_before_pay_CD/>
        <pay_after_accept>1</pay_after_accept>
        <pay_after_test>0</pay_after_test>
      </shipment>
    </row>
  </loadings>
</parcels>
</payment>
```

```

        <side>SENDER</side>
        <method>CASH</method>
    </payment>
    <instructions>
        <e><!-- in case of failed delivery, the shipment to be returned automatically
to Marin Marinov at office Ruse Gradski Ploshtad-->
        <type>return</type>
        <delivery_fail_action>return_to_office</delivery_fail_action>
        <return_name>Марин Маринов</return_name>
        <return_phone>088888888</return_phone>
        <return_email>ddd@ddd.dd</return_email>
        <return_city>Рyce</return_city>
        <return_post_code>7000</return_post_code>
        <return_office_code>7004</return_office_code>
        <reject_delivery_payment_side>receiver</reject_delivery_payment_
side>
        <reject_return_payment_side>sender</reject_return_payment_side
>
        </e>
    </instructions>
</row>
</loadings>
</parcels>

```

2.2 Example with information about the serviced settlements

```

<request>
    <client>
        <username>demo</username>
        <password>demo</password>
    </client>
    <request_type>cities</request_type>
</request>

```

2.3 Example with information for the Econt's offices

```

<request>
    <client>
        <username>demo</username>
        <password>demo</password>
    </client>
    <request_type>offices</request_type>
</request>

```

2.4 Example with information about the client's profile with the provided account

```

<?xml version="1.0" encoding="UTF-8"?>
<request>
    <client>
        <username>iasp-dev</username>
        <password>iasp-dev</password>
    </client>
    <request_type>profile</request_type>
</request>

```

2.5 Example with information about the settlements' streets

```

<?xml version="1.0" encoding="UTF-8"?>
<request>

```

```

    <client>
        <username>demo</username>
        <password>demo</password>
    </client>
    <request_type>cities_streets</request_type>
</request>

```

2.6 Example for sending a request to the API and how PHP arrays get converted to and from XML

The example contains only the mandatory fields and the most commonly used services.

```

<?php
class EcontXMLClient {
    public static function request($url, $params = array(), $timeout = 10) {
        $request = new \SimpleXMLElement('<?xml version="1.0" encoding="UTF-8"?><request></request>');
        self::array2XMLNode($params, $request);
        $ch = curl_init($url);
        curl_setopt_array($ch, array(
            CURLOPT_RETURNTRANSFER => true,
            CURLOPT_TIMEOUT => $timeout,
            CURLOPT_POSTFIELDS => array(
                'xml' => $request->asXML()
            )
        ));
        $r = curl_exec($ch);
        if($r === false) throw new Exception("Connection error.");
        //If the user or the password are incorrect, the result is HTML!!
        return json_decode(json_encode(new \SimpleXMLElement($r)), true); //poor man's XML2Array
    }
    public static function array2XMLNode($array, \SimpleXMLElement $parentNode) {
        if(!is_array($array)) return;
        foreach($array as $k => $v) {
            if(is_array($v)) {
                if(!array_key_exists(0, $v)) $vv = array($v);
                else $vv = $v;
                foreach ($vv as $vvv) {
                    self::array2XMLNode($vvv, $parentNode->addChild($k));
                }
            } else {
                $parentNode->addChild($k, $v);
            }
        }
    }
}

$params = array(
    'system' => array(
        'validate' => 1, //1 – only validates the data, 0 – creates a shipment
        'only_calculate' => 0, //1 – only calculates price, 0 – creates a shipment
        'response_type' => 'XML',
        'email_errors_to' => 'tihomir@tivart.com', //The e-mail to which the errors will be sent
    ),
    'client' => array(
        'username' => 'vatev',
        'password' => 'vatev@pass',
    ),
    'loadings' => array(
        'row' => array(
            'sender' => array(
                'name' => "name of the client",
                'name_person' => "name of the natural person, if the client is a company",
                'email' => 'email of the client',
                'phone' => 'phone number of the client',
                'email_on_delivery' => 'notification email for a successful delivery',
            )
        )
    )
);

```

```

'city'=>'the name of the settlement',
'post_code' => 'post code',
'office_code' => 'office code',//if it is TO office
'quarter' =>'name of the quarter',
'street' => 'name of the street',
'street_num' => 'number of the street',
'street_bl' => 'block',
'street_vh' => 'entrance',
'street_et' => 'floor',
'street_ap' => 'apartment',
'street_other' => 'additional information for the address'
),
'receiver' => array(
'name' => "name of the client",
'name_person' => " name of the natural person, if the client is a company ",
'email' => 'email of the client',
'phone' => 'phone number of the client',
'sms_no' => 'phone number for sms notification upon successful delivery',
'city'=>'name of the settlement',
'post_code' => 'post code',
'office_code' => 'office code',//if it is TO office
'quarter' => 'name of the quarter',
'street' => 'name of the street',
'street_num' => 'number of the street',
'street_bl' => 'block',
'street_vh' => 'entrance',
'street_et' => 'floor',
'street_ap' => 'apartment',
'street_other' => 'additional information for the address'
),
'payment' => array(
'side' => 'SENDER',
'method' => 'CASH',
'key_word' => "
),
'shipment' => array(
'shipment_type' => 'PACK',
'description' => 'shipment description',
'pack_count' => 1,//packs count
'weight' => 1,
'tariff_sub_code' => 'DOOR_DOOR',// whether the shipment is to/from office/address - DOOR_DOOR,
DOOR_OFFICE, OFFICE_DOOR, OFFICE_OFFICE
'pay_after_accept' => 1,//the receiver can review the shipment before payment of the cash on delivery
'pay_after_test' => 1,//the receiver can review and test the shipment before payment of the cash on delivery
),
'services' => array(
'oc' => 18.42,//declared value
'oc_currency' => 'BGN',//currency of the declared value
'cd' => 18.42,//cash on delivery
'cd_currency' => 'BGN',//currency of the cash on delivery
'cd_agreement_num' => "",// Number of the agreement for cash on delivery payment(for example if it will be
paid via bank)
),
)
);
var_dump(EcontXMLClient::request('http://www.econt.com/e-econt/xml_parcel_import2.php',$params));

```