

Съдържание

1. Информация и инструкции за използване	2
1.1. Възможности на API за интеграция.....	2
1.2. Формат на API интеграцията.....	2
1.3. URL, от който се изпращат заявки и потребителски акаунти за достъп.....	2
1.4. Валидация на адреси, населени места, офиси на Еконт.....	2
2. Примерни заявки.....	3
2.1 Пример за създаване на пратка	3
2.2 Пример с информация за населените места	4
2.3 Пример с информация за офисите на Еконт Експрес	4
2.4 Пример с информация за профила на клиента с подадения акаунт	4
2.5 Пример с информация за улиците на населените места.....	5
2.6 Пример за изпращане на заявка към API и как се конвертират PHP масиви от и към XML	5

1. Информация и инструкции за използване

1.1. Възможности на API за интеграция

- калкулиране на цена на пратка
- създаване на товарителница и генериране на PDF за разпечатването ѝ, валидация на данните за товарителница
- заявка за посещение от куриер
- проверка на статус на пратка и наложен платеж, проследяване на пратка
- информация за офисите на Еконт
- информация за населените места, обслужвани от Еконт
- информация за улиците и кварталите в населените места
- информация за е-Econt потребителя и адресите му за изпращане и получаване на пратки

1.2. Формат на API интеграцията

Заявките и отговорите на API са в XML формат, като пълното описание на структурата му може да намерите [ТУК](#).

1.3. URL, от който се изпращат заявки и потребителски акаунти за достъп

- за заявки за калкулация, подготовка на пратки, парични преводи, заявки за куриер:

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

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

- за останалите видове заявки:

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

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

XML заявките се изпращат чрез http/https POST по един от следните начини:

- като прикачен файл с Content-type: multipart/form-data (същото като прикачен файл от форма в браузър)

- като POST параметър с име 'xml' и Content-type: application/x-www-form-urlencoded (същото като поле с име 'xml' от форма в браузър)

- директно като съдържание на POST заявката с Content-type: text/xml

За тестовата система може да използвате потребителско име: demo и парола: demo

За production системата можете да използвате съществуващ е-Econt потребител или да регистрирате нов на https://ee.econt.com/load_direct.php?target=Register или чрез връзка със служител на Еконт.

Ауторизация посредством SSL клиентски сертификати към момента не се поддържа, но може да използвате SSL връзка и да се ауторизирате по обичайния начин (в XML заявката).

1.4. Валидация на адреси, населени места, офиси на Еконт

Валидните имена на населени места, улици и квартали може да получите чрез API заявки към http://www.econt.com/e-econt/xml_service_tool.php:

- населени места и пощенски кодове: <request_type>cities</request_type>

- офиси: <request_type>offices</request_type>

- квартали: <request_type>cities_quarters</request_type>

- улици: <request_type>cities_streets</request_type>

Възможно е също така и да предоставите възможност на клиента да избира офиса от карта, като използвате готовия офис локатор на Еконт.

Минимална примерна имплементация на избиране на офис с офис локатора може да видите на http://ee.econt.com/office_locator_example.html

При подготовка на пратка адресите на подател, получател и евентуални други страни (за получаване

на сумата по наложен платеж, връщане на пратка и др.) е необходимо да са валидни по следния критерии:

1. Валидни пощенски код и име на населено място.
2. Поне едно от:
 - валидно име на улица (таг <street>) и номер (таг <street_num>)
 - валидно име на квартал (таг <quarter>) и стойност за поле "друго" (таг <street_other>)
 - за населени места, за които няма данни за улици: име на улица без валидация (таг <street>) и номер (таг <street_num>)
 - за пратки до поискване: валиден код на офис на Еконт (таг <office_code>)

За калкулиране на цена за доставка са достатъчни валидни населено място и пощенски код за подател и получател.

2. Примерни заявки

2.1 Пример за създаване на пратка

```
<?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>ПАКЪ</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>
    </services>
  </services>
</payment>
```

```

        <side>SENDER</side>
        <method>CASH</method>
    </payment>
    <instructions>
        <e><!-- при неуспешна доставка пратката да се върне автоматично на
Марин Маринов в офис Русе Градски площад -->
        <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>Русе</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 Пример с информация за населените места

```

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

```

2.3 Пример с информация за офисите на Еконт Експрес

```

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

```

2.4 Пример с информация за профила на клиента с подадения акаунт

```

<?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 Пример с информация за улиците на населените места

```
<?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 Пример за изпращане на заявка към API и как се конвертират PHP масиви от и към XML

Примерът съдържа само задължителните полета и най-често използваните услуги.

```
<?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($r === false) throw new Exception("Connection error.");
    //ako potrebiteq ili parolata ne sa verni, rezultata e HTML!!
    return json_decode(json_encode(new \SimpleXMLElement($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 - samo validira dannite, 0 - suzdava pratka
    'only_calculate' => 0, //1 - samo kalkulaciq na cena, 0 - suzdava pratka
    'response_type' => 'XML',
    'email_errors_to' => 'tihomir@tivart.com', // e-mail na koito shte se izpratqg greshkite
  ),
  'client' => array(
    'username' => 'vatev',
    'password' => 'vatev@pass',
  ),
  'loadings' => array(
    'row' => array(
      'sender' => array(
```

```

'name' => "ime na klienta",
'name_person' => "ime na fiz lice ako klienta e firma",
'email' => 'email na klienta',
'phone' => 'telefon na klienta',
'email_on_delivery' => 'email za notifikaciq pri dostavka na pratkata',
'city'=>'ime na naseleno mqsto',
'post_code' => 'pshtenski kod',
'office_code' => 'kod na ofis',//ako e ot ofis
'quarter' => 'ime na kvartal',
'street' => 'ime na ulica',
'street_num' => 'nomer',
'street_bl' => 'blok',
'street_vh' => 'vhod',
'street_et' => 'etaj',
'street_ap' => 'apartament',
'street_other' => 'dopolnitelna informaciq kum adresa'
),
'receiver' => array(
'name' => "ime na klienta",
'name_person' => "ime na fiz lice ako klienta e firma",
'email' => 'email na klienta',
'phone' => 'telefon na klienta',
'sms_no' => 'telefon za sms notifikaciq pri dostavka na pratkata',
'city'=>'ime na naseleno mqsto',
'post_code' => 'pshtenski kod',
'office_code' => 'kod na ofis',//ako e do ofis
'quarter' => 'ime na kvartal',
'street' => 'ime na ulica',
'street_num' => 'nomer',
'street_bl' => 'blok',
'street_vh' => 'vhod',
'street_et' => 'etaj',
'street_ap' => 'apartament',
'street_other' => 'dopolnitelna informaciq kum adresa'
),
'payment' => array(
'side' => 'SENDER',
'method' => 'CASH',
'key_word' => "
),
'shipment' => array(
'shipment_type' => 'PACK',
'description' => 'opisane na pratkata',
'pack_count' => 1,//broi paketi
'weight' => 1,
'tariff_sub_code' => 'DOOR_DOOR',// dali pratkata e ot/do vrata/ofis - DOOR_DOOR, DOOR_OFFICE,
OFFICE_DOOR, OFFICE_OFFICE
'pay_after_accept' => 1,//poluchatelq moje da pregleda pratkata predi da plati nalojeni platej
'pay_after_test' => 1,//poluchatelq moje da pregleda i testva pratkata predi da plati nalojeni platej
),
'services' => array(
'oc' => 18.42,//obqvena stoinost
'oc_currency' => 'BGN',//valuta na obqvenata stoinost
'cd' => 18.42,//nalojen platej
'cd_currency' => 'BGN',//valuta na nalojeni platej
'cd_agreement_num' => ",//nomer na sporazumenie za izplasztane na nalojen platej (pr. ako shte se izplashtra
po banka)
),
)
);

var_dump(EcontXMLClient::request('http://www.econt.com/e-econt/xml_parcel_import2.php',$params));

```