Technical Documentation API Cargus

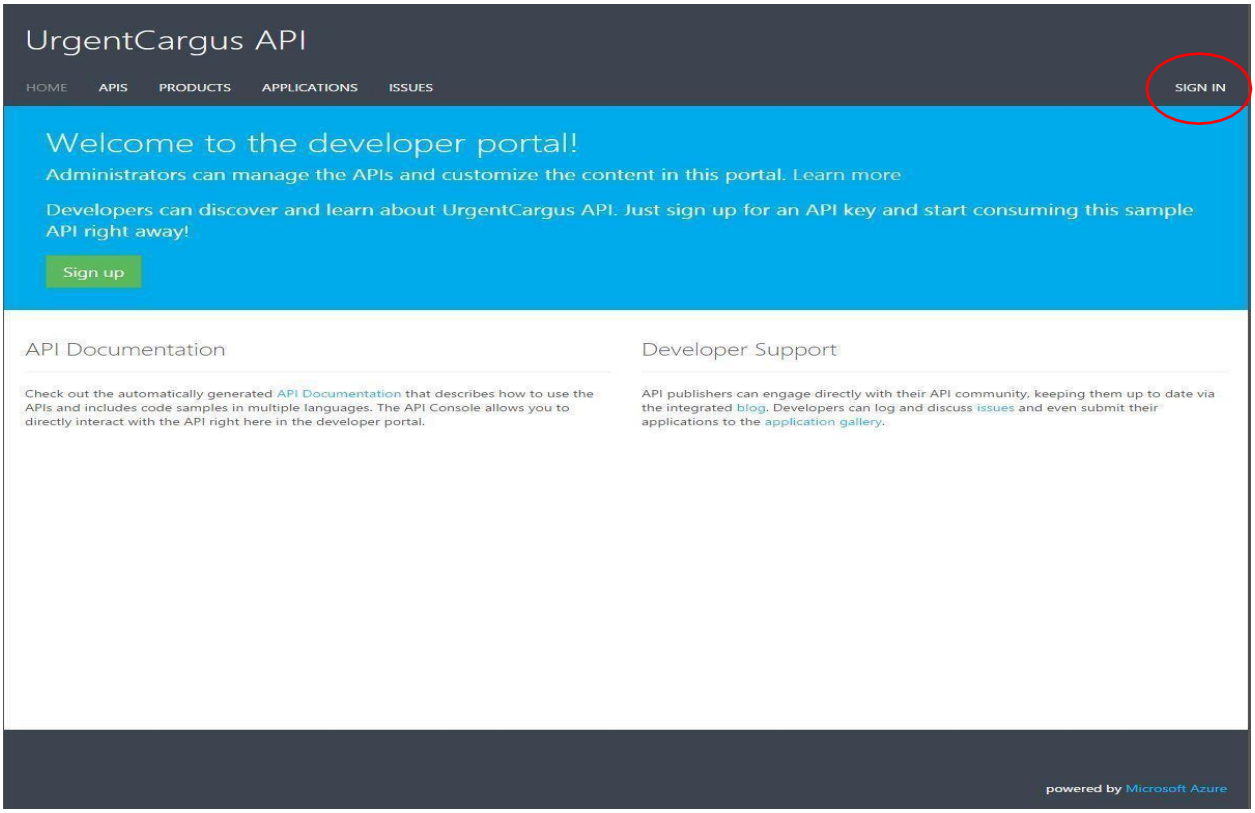
https://urgentcargus.portal.azure-api.net/

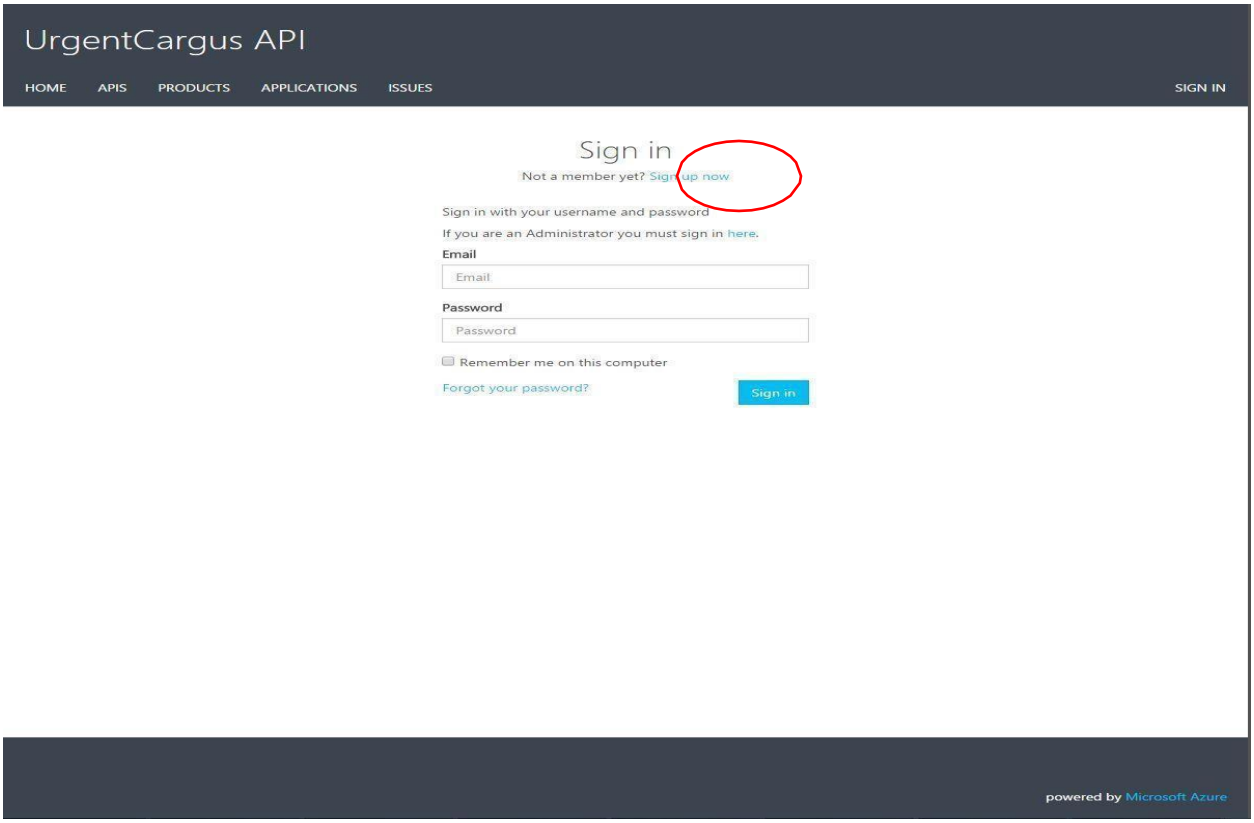
Technology Used: **REST Web Api Microsoft**

Contents

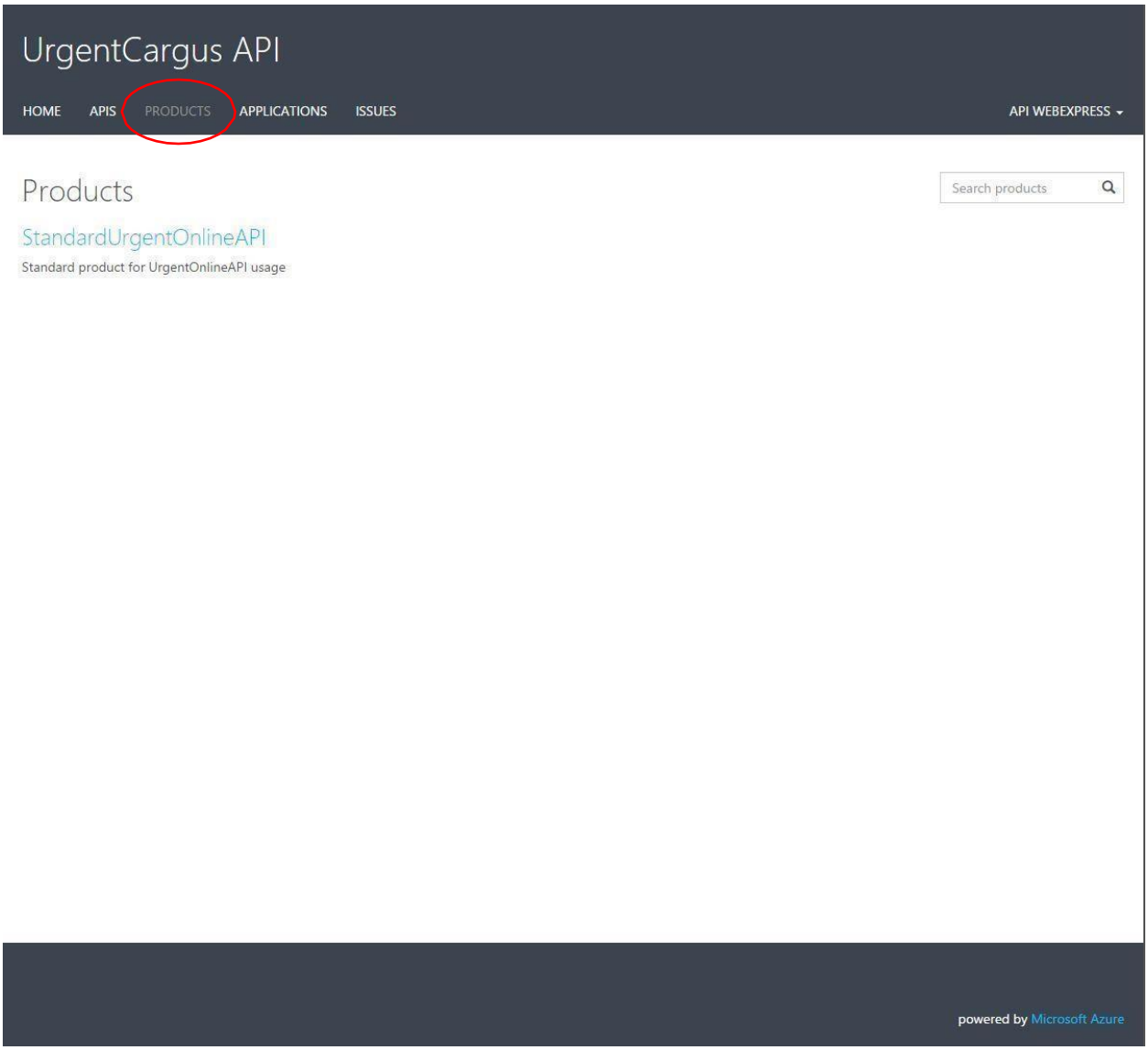
1 Registering for API StandardUrgentOnlineAPI

1. You have to create an account. SIGN IN, then SIGN UP:

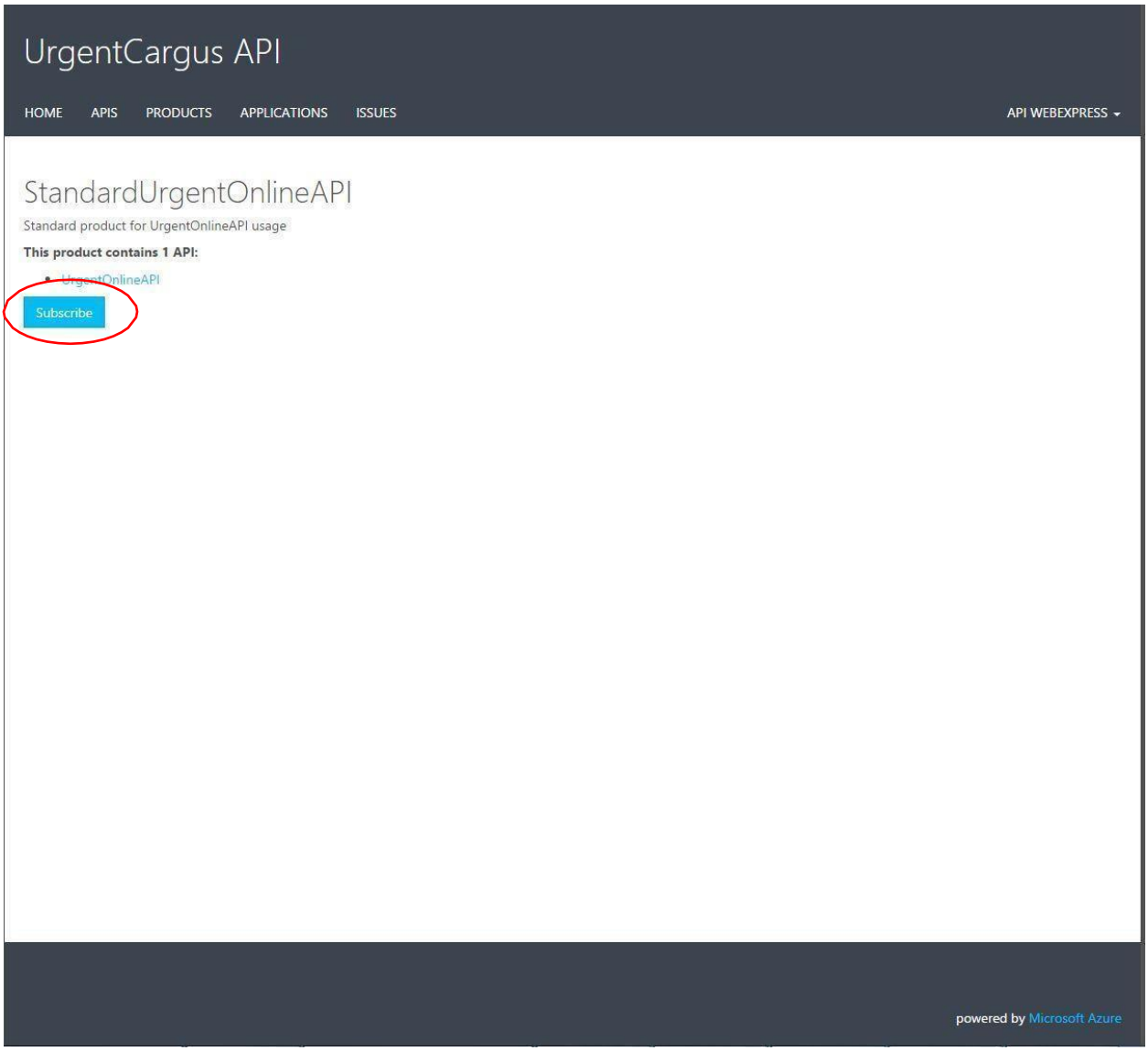




2. Fill in the required data, then confirm the registration for the service by following the confirmation link received by e-mail.
3. Being logged in, go back to the initial page and click on PRODUCTS

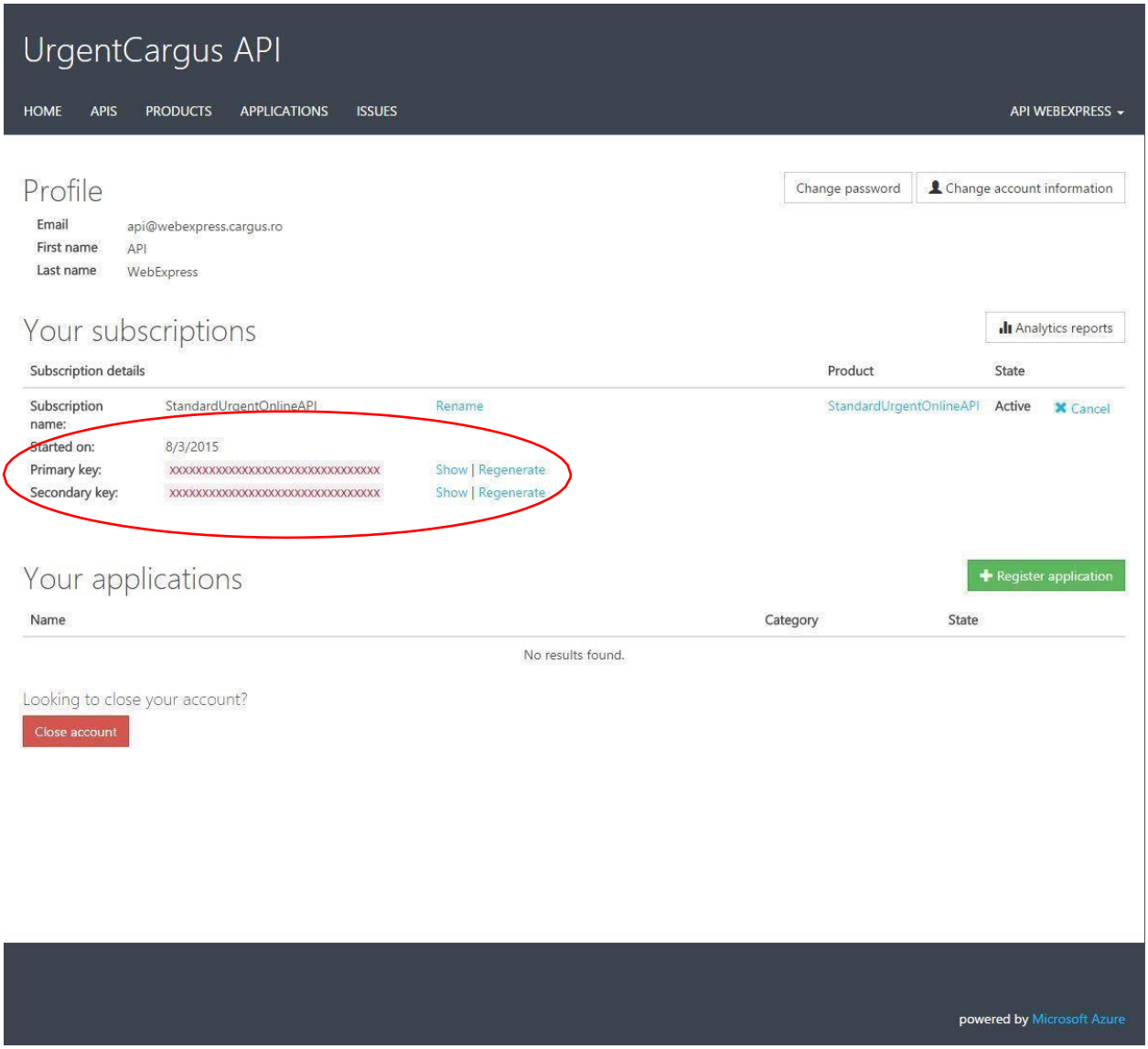

UrgentCargus API
HOME
APIS
PRODUCTS
APPLICATIONS
ISSUES
API WEBEXPRESS
Products
Search products
StandardUrgentOnlineAPI
Standard product for UrgentOnlineAPI usage
powered by Microsoft Azure

4. Subscribe to our API: **StandardUrgentOnlineAPI**



5. A service administrator will approve your subscription request. After the approval you will receive a confirmation e-mail, then you can start using the API service.

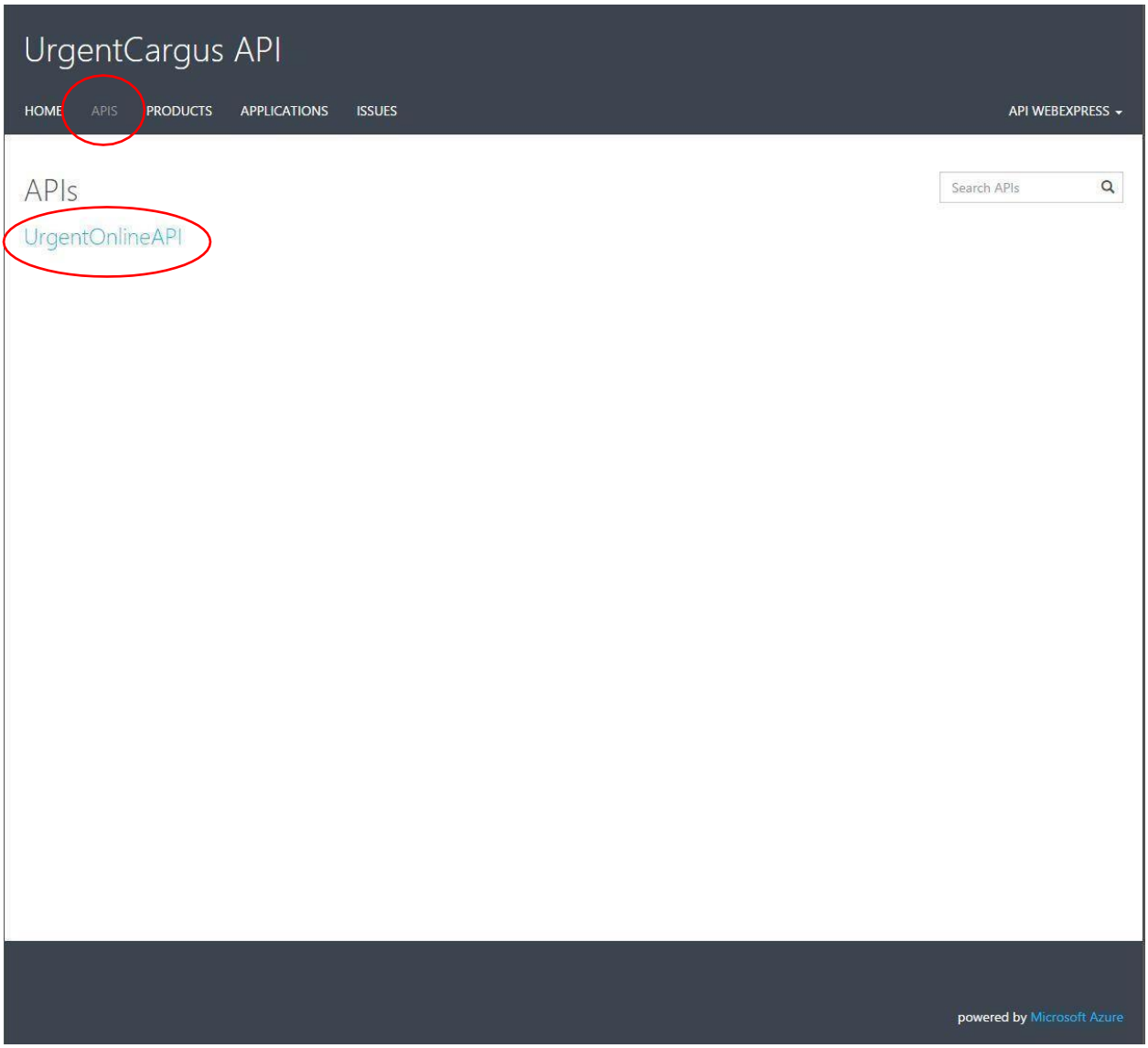
6. Click on PRODUCTS -> **StandardUrgentOnlineAPI -> StandardUrgentOnlineAPI** and retain the Primary Key for calling the API methods.

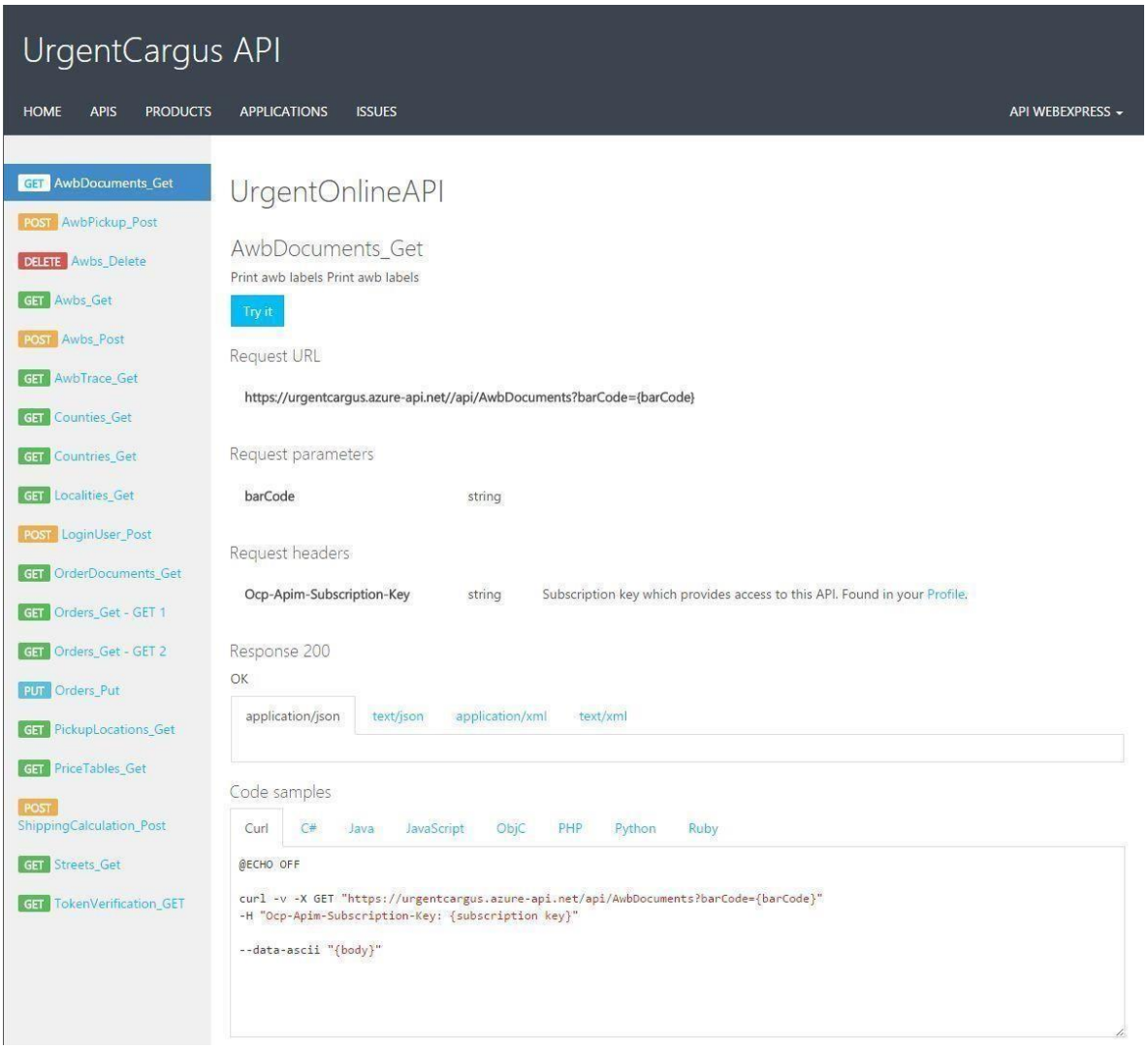


The value in this format xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx will be sent in the header of each request, in the form of two parameters: **'Ocp-Apim-Subscription-Key: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx' , 'Ocp-Apim-Trace:true'**

7. Also, for API authentication, you need to send in the header another parameter, of this form: 'Authorization: Bearer TOKEN ', where TOKEN is the character string obtained from calling the **LoginUser** method.

8. For a complete definition of the API and on-page calling you can use the interface, accessing APIS->UrgentOnlineAPI:




UrgentCargus API
HOME
APIS
PRODUCTS
APPLICATIONS
ISSUES
API WEBEXPRESS
GET AwbDocuments_Get
POST AwbPickup_Post
DELETE Awbs_Delete
GET Awbs_Get
POST Awbs_Post
GET AwbTrace_Get
GET Counties_Get
GET Countries_Get
GET Localities_Get
POST LoginUser_Post
GET OrderDocuments_Get
GET Orders_Get - GET 1
GET Orders_Get - GET 2
PUT Orders_Put
GET PickupLocations_Get
GET PriceTables_Get
POST ShippingCalculation_Post
GET Streets_Get
GET TokenVerification_GET
UrgentOnlineAPI
AwbDocuments_Get
Print awb labels Print awb labels
Try it
Request URL
https://urgentcargus.azure-api.net//api/AwbDocuments?barCode={barCode}
Request parameters
barCode
string
Request headers
Ocp-Apim-Subscription-Key
string
Subscription key which provides access to this API. Found in your Profile.
Response 200
OK
application/json
text/json
application/xml
text/xml
Code samples
Curl
C#
Java
JavaScript
ObjC
PHP
Python
Ruby
@ECHO OFF
curl -v -X GET "https://urgentcargus.azure-api.net/api/AwbDocuments?barCode={barCode}"
-H "Ocp-Apim-Subscription-Key: {subscription key}"
--data-ascii "{body}"

2 PHP call example:

```
<?php
class UrgentCurier
{
   private $Curl;
   public $url='https://urgentcargus.azure-api.net/api';

        function __construct()
        {
                $this->Curl = curl_init();
                curl_setopt($this->Curl, CURLOPT_RETURNTRANSFER, true);
                curl_setopt($this->Curl, CURLOPT_CONNECTTIMEOUT, 2);
                curl_setopt($this->Curl, CURLOPT_TIMEOUT, 3);
        }

        function CallMethod($function, $parameters="", $verb,$token=null)
        {
                curl_setopt($this->Curl, CURLOPT_POSTFIELDS, $parameters);
                curl_setopt($this->Curl, CURLOPT_CUSTOMREQUEST, $verb);
                curl_setopt($this->Curl, CURLOPT_URL, $this->url . '/' . $function);

                //LoginUser este singura metoda pentru care nu se trimite Token
                 if ($function=="LoginUser") {
                        curl_setopt($this->Curl, CURLOPT_HTTPHEADER, array('Ocp-Apim-Subscription-
                        Key:xxxxxxxxxxxxxxxxxxxxxxxxxx','Ocp-Apim-Trace:true','Content-Type:
                        application/json', 'Content-Length: ' . strlen($parameters)));
                }
                else {
                        curl_setopt($this->Curl, CURLOPT_HTTPHEADER, array('Ocp-Apim-Subscription-
                        Key: xxxxxxxxxxxxxxxxxxxxxxxxxx','Ocp-Apim-Trace:true','Authorization: Bearer
                        '.$token, 'Content-Type: application/json', 'Content-Length: ' .
                        strlen($parameters)));
                }


                $result=curl_exec($this->Curl);
                $header=curl_getinfo($this->Curl);

                 $output['message']=$result;
                $output['status']=$header['http_code'];
                return $output;
        }
}
$urgent = new UrgentCurier();
?>
```

3 Authentication

3.1 Logging in and obtaining token

The method **LoginUser** called by POST is used to authenticate and obtain the token. The obtained token is valid for 24 hours.

You send :

UserName – the webexpress user
Password – the user password

It returns :

Token – will be used in all the called methods as a means of authentication

Example:

```
$fields=array('UserName'  =>  'test.integration9', 'Password '=> 'a');
$json=json_encode($fields);
$login=$urgent->CallMethod('LoginUser', $json, 'POST');
if ($login['status'] != "200")  {

        echo "<b class='bad'>LoginUser: FALSE</b>";
}
else {
        $token = json_decode($login['message']);
        echo "<b class='good'>LoginUser: </b>".$token;
}
```

3.2 Token Verification

The method used is **TokenVerification.** It is a GET type method and it is used if you want to verify the authentication.

You send:

Token – returned by the LoginUser method

It returns :

true / false

Example:

```
$result=$urgent->CallMethod('TokenVerification', $json="", 'GET', $token);
if ($result['status']!="200") {
        echo "<b class='bad'>Token check: FALSE</b>";
}
else{
        $data=$result['message'];
```

```
echo "<b class='good'>Token check: </b>".$data; }
```

4 Geography

4.1 The list of countries in the system

The method used is **Countries** , called through GET and it returns the list of the countries in the Cargus system.

You send:

It returns:

A list with the accepted countries. The list provides the following information:

CountryId – the country id
CountryName – the country name
Abbreviation – the country code

Example:

```
$resultCountries = $urgent->CallMethod('Countries',  $json="", 'GET', $token);

if ($resultCountries['status'] != "200") {
        echo"<b class='bad'>GET Countries: </b>".$resultCountries['message'];
}
else {
        $jsonCountriesList = $resultCountries['message'];
        echo "<b class='good'>GET Countries: </b>".$jsonCountriesList;
}
```

4.2 The list of counties in a country

The **Counties** method is used to return the list of the counties in a country. It is a GET type method, and in the URL the country id is sent.

You send :

countryId - the country id, returned by the Countries method

It returns:

A list of the accepted. The list provides the following information:

CountyId – county id
Name – county name
Abbreviation – county code

Example:

```
$resultCounties = $urgent->CallMethod('Counties?countryId=1', $json="", 'GET', $token);
if ($resultCounties['status']!="200") {
        echo"<b class='bad'>GET Countries: </b>".$resultCounties['message'];
        }
```

```
else {
```

```
        $jsonCountiesList = $resultCounties['message'];
        echo "<b class='good'>GET Counties List: </b>".$jsonCountiesList;
}
```

4.3 The list of localities in a county

The **Localities** method, called through GET, is used to return the list of localities in a given county.

You send :

countryId - the country id returned by the Countries method
countyId – the county returned by the Counties method

It returns:

A list of accepted localities and the corresponding information. The list provides the following information:

LocalityId – locality id
Name – locality name
ParentId – the territorial agency to which the locality belongs
ParentName – the name of the territorial agency
ExtraKm – chargeable extra kilometers
InNetwork – in the network
CountyId – county id
CountryId – country id
CodPostal – postal code
MaxHour – pick up hour

Example:

```
$resultLocalities = $urgent->CallMethod('Localities?countryId=1&countyId=5', $json="",  'GET',
$token);
if ($resultLocalities ['status']!="200")  {
        echo"<b class='bad'>GET Localities: </b>".$resultLocalities['message'];
}
else {
        $jsonLocalitiesList = $resultLocalities['message'];
        echo "<b class='good'>GET Localities List: </b>".$jsonLocalitiesList;
}
```

4.4 The list of streets in a locality

The **Streets** method, called through GET, is used to return the list of the streets in the Cargus nomenclature, in a given locality.

You send :

localityId - the locality id, returned by the Localities method.

It returns:

A list of the streets in the nomenclature and the afferent information. The list provides the following information:

StreetId – street id
Name – street name

Example:

```
$resultStreets = $urgent->CallMethod('Streets?localityId=157', $json="", 'GET', $token);
if ($resultStreets ['status']!="200") {
    echo"<b class='bad'>GET Streets: </b>".$resultStreets['message'];
}
else {
    $jsonStreetsList = $resultStreets['message'];
    echo "<b class='good'>GET Streets List: </b>".$jsonStreetsList;
}
```

5 Pick up points

A customer has more pick up points. A user has access to one or multiple pick up points.

By default, a customer has one pick up point per headquarters, which is used in the application as LocationId=0.

5.1 The list of pick up points per customer

The **PickupLocations/GetForClient** method is a GET type method. It is used to return the list of pick up points for a customer.

You send :

token – identify the customer

It returns:

A list of pick up points and the afferent information. The list provides the following information:

LocationId – pick up point id
Name – pick up point name
CountyId – county id
CountyName – county name
LocalityId – locality id
LocalityName – locality name
StreetId – street id
StreetName – street name
BuildingNumber – street number
AddressText - address
ContactPerson – contact
PhoneNumber – contact phone number
Email – e-mail address
"CodPostal": - postal code for the pickup location

Example:

```
$resultPickupLocations = $urgent->CallMethod('PickupLocations/GetForClient', $json="", 'GET',
$token);
if ($resultPickupLocations ['status']!="200") {
        echo"<b class='bad'>GET PickupLocations: </b>".$resultPickupLocations['message'];
}
else {
        $jsonPickupLocations = $resultPickupLocations['message'];
        echo "<b class='good'>GET PickupLocations List: </b>".$jsonPickupLocations;
}
```

5.2 Adding (assigning) a pick up point for user

The **PickupLocations/AssignToUser** method, called through POST, is used to add (assign) a pick up point per user.

You send:

token – identifies the customer
LocationId – the id of the pick up point which will be assigned to the logged in user

It returns:

1 – if the activation is successful / error – if the activation was not successful

Example:

```
$resultAssignToUser = $urgent-
>CallMethod('PickupLocations/AssignToUser?LocationId=201008098', $json="", 'POST',$token);

if ($resultAssignToUser ['status']!="200") {
        echo"<b class='bad'>GET AssignToUser: </b>".$resultAssignToUser['message'];
}
else {
        $jsonAssignToUser = $resultAssignToUser['message'];
        echo "<b class='good'>POST AssignToUser Status: </b>".$jsonAssignToUser;
}
```

5.3 Listing the active pick up points per user

The **PickupLocations** method, called through GET, is used to list the pick up points that are active for a given user.

You send :

token – identify the customer

It returns:

LocationId – the pick up point id
Name – the pick up point name
CountyId – the county id
CountyName – the county name

LocalityId – the locality id

LocalityName – the locality name
StreetId – the street id
StreetName – the street name
BuildingNumber – the street number
AddressText – the address
ContactPerson – the contact
PhoneNumber – the contact's phone number
Email – the e-mail address
CodPostal– postal code for pickup point

Example:

```
$resultPickupLocations = $urgent->CallMethod('PickupLocations', $json="", 'GET',$token);
if ($resultPickupLocations ['status']!="200")  {
        echo"<b class='bad'>GET PickupLocations: </b>".$resultPickupLocations['message'];
}
else {
        $jsonPickupLocations = $resultPickupLocations['message'];
        echo "<b class='good'>GET PickupLocations Status: </b>".$jsonPickupLocations;
}
```

5.4 Adding a pick up point for a user

The **PickupLocations** method is used to add a new pick up point for a user. The pick up point is active by default. The method is a POST type method.

You send :

AutomaticEOD – booking closing hour
LocationId – pick up point id
Name – pick up point name
CountyId – county id
CountyName – county name
LocalityId – locality id
LocalityName – locality name
StreetId - street id or 0
StreetName – street name
BuildingNumber – street number
AddressText - address
ContactPerson – contact name
PhoneNumber – contact phone number
Email – e-mail address
CodPostal– postal code for pickup point

It returns:

the id of the pick up point

Example:

```
$jsonPickupLocations = '{
        "AutomaticEOD": "17:30",
        "LocationId": "",
        "Name": "Punct ridicare 2 Pitesti",
        "CountyId": 5,
        "CountyName": "Arges",
        "LocalityId": 157,
        "LocalityName": "Pitesti",
        "StreetId": 0,
        "StreetName": "Gheorghe S. Ioan, general, Strada",
         "BuildingNumber": "5",
        "AddressText": "Str. Ionescu Gheorghe",
       "ContactPerson": "Gheorghe",
       "PhoneNumber": "072769821",
       "CodPostal": "string",
        "Email": "a@a.com"
}';

$resultPickupLocationsPost = $urgent->CallMethod('PickupLocations',
$jsonPickupLocations, 'POST',$token);
if ($resultPickupLocationsPost ['status']!="200") {
        echo"<b class='bad'>Post PickupLocations:
</b>".$resultPickupLocationsPost['message'];
}
else {
        $jsonPickupLocationsPost = $resultPickupLocationsPost['message'];
        echo "<b class='good'>Post PickupLocations id-location:
</b>".$jsonPickupLocationsPost;
}
```

5.5 Modifying a pick up point

The **PickupLocations** method is used to modify a pick up point. This method is a PUT type method.

You send :

AutomaticEOD – booking closing hour; if none, type NULL
LocationId – pick up point id
Name – pick up point name
CountyId – county id
CountyName – county name
LocalityId – locality id
LocalityName – locality name
StreetId - street id or 0
StreetName – street name
BuildingNumber – street number
AddressText - pick up point address

ContactPerson – contact name
PhoneNumber – contact phone number
Email – e-mail address
CodPostal– postal code for pickup point

It returns:

True

Example:

```
$jsonPickupLocations =
 '{"AutomaticEOD":null,
 "LocationId": "31498",
 "Name": "ecomMODIFICAT",
 "CountyId": 5,
 "CountyName": "Arges",
 "LocalityId": 157,
 "LocalityName": "pitesti",
 "StreetId": 0,
  "StreetName": "Gheorghe S. Ioan, general, Strada",
  "BuildingNumber": "5",
  "AddressText": "bl 5 ap 5",
  "ContactPerson": "Gheorghe",
  "PhoneNumber": "072769821",
  "CodPostal": "string",
  "Email": "a@a.com"
}';


$result = $urgent->CallMethod('PickupLocations', $jsonPickupLocations, 'PUT',$token);


if ($result['status']!="200") {
        echo("<b class='bad'>PUT PickupLocations: </b>".$result['message']);
}
else {
        $data=$result['message']; echo "<b class='good'>PUT PickupLocations:
</b>".$data;
}
```

6 Show address book

The **Recipient** method, called through GET, is used to list established address book

You send :

token – identify the customer

It returns:

A list of your address book , which contains the following

LocationId – pick up point id
Name – recipient name
CountyId – county id
CountyName- county name
LocalityId – locality id
LocalityName – locality name
StreetId – street id
StreetName – street name
BuildingNumber – bulding number
AddressText – address
ContactPerson – contact person
PhoneNumber – phone number
Email – e-mail
CodPostal– postal code for delivery point

Example:

```
$resultRecipients = $urgent->CallMethod('Recipients', $json="", 'GET', $token);
if ($resultRecipients ['status']!="200") {
        echo"<b class='bad'>GET 'Recipients': </b>".$resultRecipients ['message'];
}
else {
        $jsonRecipients = $ resultRecipients ['message'];
        echo "<b class='good'>GET resultRecipients List: </b>".$ jsonRecipients;
}
```

7 Query Ship & Go centers

The PUDO_Get is called. It is used to return the list of Ship & Go points.

You send:

Token-Used to identify the client

It returns:

A list of delivery points and related information. The list includes the following information:

Example:


```
{
   "Id": 114142,
   "Name": "CARGUS SHIP & GO MAGURELE",
   "LocationId": 1,
   "CityId": 1793631,
   "City": "Magurele",
   "StreetId": 39689,
   "ZoneId": 8728,
   "PostalCode": "077125",
   "AdditionalAddressInfo": "",
   "Longitude": 26.041645,
   "Latitude": 44.367229,
   "PointType": 5,
   "OpenHoursMoStart": "08:00",
   "OpenHoursMoEnd": "18:00",
   "OpenHoursTuStart": "08:00",
   "OpenHoursTuEnd": "18:00",
   "OpenHoursWeStart": "08:00",
   "OpenHoursWeEnd": "18:00",
   "OpenHoursThStart": "08:00",
   "OpenHoursThEnd": "18:00",
   "OpenHoursFrStart": "08:00",
   "OpenHoursFrEnd": "18:00",
   "OpenHoursSaStart": "",
   "OpenHoursSaEnd": "",
   "OpenHoursSuStart": "",
   "OpenHoursSuEnd": "",
   "StreetNo": "99-115",
   "PhoneNumber": "021 9330",
   "ServiceCOD": false,
   "PaymentType": 1,
   "CountyId": 27,
   "County": "Ilfov",
   "Email": "",
   "MainPicture": "",
   "Entrance": null,
   "Floor": null,
   "Apartment": null,
   "Sector": null,
   "StreetName": "Atomistilor"
 },
```


Where:

"Id" – id of the delivery point
"Name" – name of the delivery point

"LocationId" – something internal
"CityId" – city id for delivery point
"City" – city name for delivery point
"StreetId" – street id for delivery point
"ZoneId" – something internal
"PostalCode" – zipcode for delivery point
"AdditionalAddressInfo" – additional address info for the delivery point
"Longitude" – coordinates on map for the delivery point
"Latitude" – coordinates on map for the delivery point
"PointType" – used for deliverypoints associated with Cargus Warehouse
"OpenHoursMoStart" – open time on monday
"OpenHoursMoEnd" – close time on monday
"OpenHoursTuStart" – open time on tuesday
"OpenHoursTuEnd" – close time on tuesday
"OpenHoursWeStart" – open time on wednesday
"OpenHoursWeEnd" – close time on wednesday
"OpenHoursThStart" – open time on thursday
"OpenHoursThEnd" – close time on thursday
"OpenHoursFrStart" – open time on friday
"OpenHoursFrEnd" – close time on friday
"OpenHoursSaStart" – open time on saturday
"OpenHoursSaEnd" – close time on saturday
"OpenHoursSuStart" – open time on sunday
"OpenHoursSuEnd" – close time on sunday
"StreetNo – street number of the delivery point
"PhoneNumber" – phone number of the delivery point
"ServiceCOD" – the delivery point dose or dose not allow COD
"PaymentType" – payment type of the delivery point (1 – no payment available2 – pay only by card, 3 – pay by cash or card, 4 – pay only cash)
"CountyId" – county id of the delivery point
"County" – county name of the delivery point
"Email" – email of the delivery point
"MainPicture" – picture of the delivery point
"Entrance" – entrance infomations of the delivery point
"Floor" – floor of the delivery point
"Apartment" – apartment of the delivery point
"Sector" – district of the delivery point
"StreetName" – street name for delivery point

8 Rates

8.1 The list of contracted prices

The **PriceTable** method, called through GET, is used to list the prices established in the contract.

You send :

token – identify the customer

it returns:

PriceTableId – price id
Name – price name

Example:

```
$resultPriceTables =$urgent->CallMethod('PriceTables', $json="", 'GET', $token);

if ($resultPriceTables ['status']!="200") {
        echo"<b class='bad
'>GET PriceTables : </b>".$resultPriceTables ['message'];
}
else {
        $data=$resultPriceTables ['message'];
        echo "<b class='good'>GET PriceTables : </b>".$data;
}
```

8.2 Price calculation for a shipment

The **ShippingCalculations** method, called through POST, is used to calculate the price of a shipment.

You send :

FromLocalityId – id of the sender's locality
ToLocalityId – id of the consignee's locality
FromCountyName - name of the sender's county / OPTIONAL
FromLocalityName – name of the sender's locality /OPTIONAL
ToCountyName – name of the consignee's county / OPTIONAL
ToLocalityName – name of the consignee's locality /OPTIONAL
Parcels – number of parcels
Envelopes – number of envelopes
TotalWeight – the total weight of the shipment
ServiceId – ServiceId: - 34 for totalweight 0 ≤ 31 kg ; 35 for 31 kg < totalweight ≤ 50 ; 50 for totalweitght > 50 kg
DeclaredValue – declared value
CashRepayment - cash repayment retu rned in the envelope
BankRepayment – bank repayment – in the bank account
OtherRepayment – anything returned// receipt, parcel in exchange
PaymentInstrumentId – payment instrument id// 1 – cheque, 2 –bo, 3
PaymentInstrumentValue – payment instrument value
OpenPackage – open package // use TRUE or FALSE
ShipmentPayer – shipment payer. Send 1 for sender and 2 for consignee
PriceTableId – price id, returned by the PriceTabels method

It returns:

BaseCost – base prices, without contract
ExtraKmCost – extra kilometers price
WeightCost – contracted price
InsuranceCost – insured price (insurance)
SpecialCost – special taxes
RepaymentCost – repayment cost : cash(envelope) or return (OtherRepayment)
Subtotal – total without VAT
Tax - VAT
GrandTotal – total price

Example:

```
$json='{
        "FromLocalityId" :150,
        "ToLocalityId" :150,
        "Parcels": 1,
        "Envelopes": 0,
        "TotalWeight": 1,
        "DeclaredValue": 0,
        "CashRepayment": 0,
        "BankRepayment": 0,
        "OtherRepayment": "",
        "OpenPackage": true,
        "PriceTableId": 23049,
        "ShipmentPayer": 1,
}';

$result=$urgent->CallMethod('ShippingCalculation', $json, 'POST', $token);
if ($result['status']!="200") {
        echo("<b class='bad'>POST ShippingCalculation: </b>".$result['message']);
}
else {
        $data=$result['message'];
        echo "<b class='good'>POST ShippingCalculation: </b>".$data;
}
```

9 Transport waybills administration

9.1 Pick-up from another location

POST called **AwbPickup** method, is used for generating a new AWB and for sending a new order to the courier.

You send:

PickupStartDate – start date and hour for pick up
PickupEndDate – end date and hour for pick up
SenderClientId – For all cases it is null. It is a feature created for those who want to create AWBs on behalf of another client.
TertiaryClientId–The third party id is used if you want to send on behalf of another client. It will be charged with the associated tariff for that customer .
Sender - Sender
Name – sender name
CountyId – county id
CountyName – county name
LocalityId – locality id
LocalityName – locality name
StreetId – street id
StreetName – street name, exactly as found in the Cargus nomenclature
BuildingNumber – street number
AddressText – address
ContactPerson – contact person for the current order
PhoneNumber – contact phone number
Email – e-mail address
CodPostal– postal code for pickup point

Recipient – Consignee
LocationId – pick up point id (recipient warehouse)
Name – recipient name
CountyId – county id
LocalityId – locality id
StreetId – street id
StreetName – street name // Optional
BuildingNumber – street number
AddressText – address

ContactPerson – contact person
PhoneNumber – contact phone number
Email – recipient e-mail address

Parcels – number of parcels
Envelopes – number of envelopes. Maximum of 9
TotalWeight – approximate total weight
DeclaredValue – declared value for insurance
CashRepayment – cash repayment (the sum returned to the sender in an envelope)
BankRepayment – bank repayment (the sum returned to the sender in the collector account)
OtherRepayment – other repayment
OpenPackage – open package
ServiceId - 34 for totalweight 0 ≤ 31 kg ; 35 for 31 kg < totalweight ≤ 50 ; 50 for totalweitght > 50 kg
HasTertReimbursement – if true the reimbursement comes back to TertiaryClientId and its obligatory
PriceTableId – price id
ShipmentPayer – shipment payer. Send 1 for sender and 2 for recipient
SaturdayDelivery – Saturday delivery
MorningDelivery – morning delivery
Observations - observations
PackageContent – pacage content
CustomString – reference customer series
SenderReference1 – sender reference 1
RecipientReference1 – recipient reference 1
RecipientReference2 – recipient reference 2
InvoiceReference – invoice reference

It returns:
Bar code number generated

Example:

```
$json='{
  "PickupStartDate": "2017-10-25T14:11",
  "PickupEndDate": "2017-10-25T18:55",

  "Sender": {

    "Name": "popescu vasile",
    "CountyName": "iasi",
    "LocalityName": "iasi",
    "AddressText": "Strada principala nr 1 bl 2",
    "ContactPerson": "popescu vasile",
    "PhoneNumber": "0723222222",
    "CodPostal": "string"
```

```
  "Email": "yahoo@yahoo.com"
},
"Recipient": {
  "LocationId": 201165677, //
},
"Parcels": 2,
"Envelopes": 0,
"TotalWeight":25,
"ServiceId": 34,
"DeclaredValue": 0,
"CashRepayment": 0,
"BankRepayment": 0,
"OtherRepayment": "string",
"BarCodeRepayment": "string",
"PaymentInstrumentId": 0,
"PaymentInstrumentValue": 0,
"HasTertReimbursement": false,
"OpenPackage": true,
"PriceTableId": 0,
"ShipmentPayer": 1,
"ShippingRepayment": 0,
"SaturdayDelivery": true,
"MorningDelivery": true,
"Observations": "string",
"PackageContent": "string",
"CustomString": "",
"BarCode": "",
"ParcelCodes": [
  {
    "Code": "0",
    "Type": 1,
    "Weight":25,
    "Length":20,
    "Width":20,
    "Height":20,
    "ParcelContent":"briefcase"

  }
],
"ValidationDate": "2019-10-24T12:33:12.035Z",
"ShippingCost": {
  "BaseCost": 0,
  "ExtraKmCost": 0,
  "WeightCost": 0,
  "InsuranceCost": 0,
  "SpecialCost": 0,
  "RepaymentCost": 0,
  "Subtotal": 0,
```

```
  "Tax": 0,
   "GrandTotal": 0
 },
 "Status": "string",
 "SenderReference1": "string",
 "RecipientReference1": "string",
 "RecipientReference2": "string",
 "InvoiceReference": "string",
 "Length": 0,
 "Width": 0,
 "Height": 0
}

        }';


        $result=$urgent->CallMethod('AwbPickup', $json, 'POST', $token);
        if ($result['status']!="200") {
                echo("<b class='bad'>POST Awb: </b>".$result['message']);
        }
        else {
                $data=$result['message'];
                echo "<b class='good'>POST Awb: </b>".$data;
        }
```

9.2 Generating transport waybill

The **Awbs** method, called through POST, is used to add a new AWB.

You send:

Sender

SenderClientId – In all cases it is null. It's a facility created for those who want to create AWBs in the name of another customer.

TertiaryClientId – tertiary id, if the shipment is made in another customer's account
LocationId – pick up point id (sender warehouse)

Recipient - Consignee

Name – recipient name
CountyId – county id
CountyName: - county name
LocalityId – locality id
LocalityName - locality name
StreetId – street id
StreetName – street name // Optional
BuildingNumber – street number
AddressText – address

ContactPerson – contact person

PhoneNumber – contact phone number
Email – recipient e-mail address
CodPostal– postal code of the consignee

Parcels – number of parcels
Envelopes – number of envelopes. Maximum of 9
TotalWeight – approximate total weight
DeclaredValue – declared value for insurance
CashRepayment – cash repayment (the sum returned to the sender in an envelope)
BankRepayment – bank repayment (the sum returned to the sender in the collector account)
OtherRepayment – other repayment
OpenPackage – open package
ServiceId: - 34 for totalweight 0 ≤ 31 kg ; 35 for 31 kg < totalweight ≤ 50 ; 50 for totalweitght > 50 kg
PriceTableId – price id
ShipmentPayer – shipment payer. Send 1 for sender and 2 for recipient
SaturdayDelivery – Saturday delivery
Observations - observation
PackageContent – package content
CustomString – reference customer series
SenderReference1 – reference sender 1
RecipientReference1 – reference recipient 1
RecipientReference2 – reference recipient 2
InvoiceReference – reference invoice
CodPostal– postal code for delivery point

It returns:

Generated barcode

Example:


```
{
 "SenderClientId": null,
 "TertiaryClientId": null,
 "TertiaryLocationId": 0,
 "Sender": {
  "LocationId": 201266091,
 },
 "Recipient": {
  "LocationId": 0,
  "Name": "Gheorghita Vasile",
  "CountyId": 0,
  "CountyName": "Iasi",
  "LocalityId": 0,
  "LocalityName": "Iasi",
  "StreetId": 0,
  "StreetName": "Str 11 iunie",
  "BuildingNumber": "14",
```

```
  "AddressText": "Str 11 iunie nr 14",
  "ContactPerson": "Gheorghita Vasile",
  "PhoneNumber": "0740123123",
  "Email": "Vasile@gmail.com",
  "CodPostal": "123456",
  "CountryId": 0
 },
 "Parcels": 2,
 "Envelopes": 0,
 "TotalWeight": 25,
 "ServiceId": 34,
 "DeclaredValue": 0,
 "CashRepayment": 0,
 "BankRepayment": 0,
 "OtherRepayment": "string",
 "BarCodeRepayment": "string",
 "PaymentInstrumentId": 0,
 "PaymentInstrumentValue": 0,
 "HasTertReimbursement": false,
 "OpenPackage": true,
 "PriceTableId": 0,
 "ShipmentPayer": 1,
 "ShippingRepayment": 0,
 "SaturdayDelivery": true,
 "MorningDelivery": true,
 "Observations": "string",
 "PackageContent": "string",
 "CustomString": "",
 "ParcelCodes": [
  {
   "Code": "0",
   "Type": 1,
   "Weight":25,
   "Length":25,
   "Width":25,
   "Height":25,
   "ParcelContent":"briefcase"

  }
 ]
}


        }';

        $result=$urgent->CallMethod('Awbs', $json, 'POST', $token);
        if ($result['status']!="200") {
                echo("<b class='bad'>POST Awb: </b>".$result['message']);
```

```
}
else {
        $data=$result['message'];
        echo "<b class='good'>POST Awb: </b>".$data;
}
```

9.3 EASY COLLECT transport waybill

This service involves sending the shipment to the Shipper & Go Cargus Ship, where it will be picked up by the shipper. Two steps are required, as follows:

9.4.1 – Query Ship & Go centers

The PODO_Get is called. It is used to return the list of Ship & Go points.

You send:

Token-Used to identify the client

It returns:

A list of delivery points and related information. The list includes the following information:

Example:


```
{
  "Id": 114142,
  "Name": "CARGUS SHIP & GO MAGURELE",
  "LocationId": 1,
  "CityId": 1793631,
  "City": "Magurele",
  "StreetId": 39689,
  "ZoneId": 8728,
  "PostalCode": "077125",
  "AdditionalAddressInfo": "",
  "Longitude": 26.041645,
  "Latitude": 44.367229,
  "PointType": 5,
  "OpenHoursMoStart": "08:00",
  "OpenHoursMoEnd": "18:00",
  "OpenHoursTuStart": "08:00",
  "OpenHoursTuEnd": "18:00",
  "OpenHoursWeStart": "08:00",
  "OpenHoursWeEnd": "18:00",
  "OpenHoursThStart": "08:00",
  "OpenHoursThEnd": "18:00",
```

```
  "OpenHoursFrStart": "08:00",
  "OpenHoursFrEnd": "18:00",
  "OpenHoursSaStart": "",
  "OpenHoursSaEnd": "",
  "OpenHoursSuStart": "",
  "OpenHoursSuEnd": "",
  "StreetNo": "99-115",
  "PhoneNumber": "021 9330",
  "ServiceCOD": false,
  "PaymentType": 1,
  "CountyId": 27,
  "County": "Ilfov",
  "Email": "",
  "MainPicture": "",
  "Entrance": null,
  "Floor": null,
  "Apartment": null,
  "Sector": null,
  "StreetName": "Atomistilor"
 },
```


Where:

"Id" – id of the delivery point
"Name" – name of the delivery point
"LocationId": 1,
"CityId" – city id for delivery point
"City" – city name for delivery point
"StreetId" – street id for delivery point
"ZoneId": 8728,
"PostalCode" – zipcode for delivery point
"AdditionalAddressInfo" – additional address info for the delivery point
"Longitude" – coordinates on map for the delivery point
"Latitude" – coordinates on map for the delivery point
"PointType" – used for deliverypoints associated with Cargus Warehouse
"OpenHoursMoStart" – open time on monday
"OpenHoursMoEnd" – close time on monday
"OpenHoursTuStart" – open time on tuesday
"OpenHoursTuEnd" – close time on tuesday
"OpenHoursWeStart" – open time on wednesday
"OpenHoursWeEnd" – close time on wednesday
"OpenHoursThStart" – open time on thursday
"OpenHoursThEnd" – close time on thursday
"OpenHoursFrStart" – open time on friday
"OpenHoursFrEnd" – close time on friday
"OpenHoursSaStart" – open time on saturday
"OpenHoursSaEnd" – close time on saturday
"OpenHoursSuStart" – open time on sunday

"OpenHoursSuEnd" – close time on sunday
"StreetNo – street number of the delivery point
"PhoneNumber" – phone number of the delivery point
"ServiceCOD": false,
"PaymentType" – payment type of the delivery point (1 – no payment available2 – pay only by card, 3 – pay by cash or card, 4 – pay only cash)
"CountyId" – county id of the delivery point
"County" – county name of the delivery point
"Email" – email of the delivery point
"MainPicture" – picture of the delivery point
"Entrance" – entrance infomations of the delivery point
"Floor" – floor of the delivery point
"Apartment" – apartment of the delivery point
"Sector" – district of the delivery point
"StreetName" – street name for delivery point

9.4.2 – Generating Easy Collect Waybill

The Awbs method called by POST is used to add a new awb.

You send:

SenderClientId – For all cases it is null. It is a feature created for those who want to create AWBs on behalf of another client.
TertiaryClientId–The third party id is used if you want to send on behalf of another client. It will be charged with the associated tariff for that customer .
DeliveryPudoPoint – Ship & Go delivery point id (pick up point id obtained using **PUDO_get)**
Sender - Sender
LocationId – pick up point id (sender warehouse)
Recipient – Consignee
ContactPerson – contact personct // mandatory
PhoneNumber – contact phone number// mandatory
Email – recipient e-mail address// mandatory

Parcels – number of parcels
Envelopes – number of envelopes. Maximum of 9
TotalWeight – approximate total weight
DeclaredValue – declared value for insurance
CashRepayment – cash repayment (the sum returned to the sender in an envelope)
BankRepayment – bank repayment (the sum returned to the sender in the collector account)
OtherRepayment – other repayment
OpenPackage – open package
HasTertReimbursement – tert reimbursement

PriceTableId – price id
ShipmentPayer – shipment payer. Send 1 for sender and 2 for recipient
ServiceId – 38 // service assigned to PUDO Delivery is 38
SaturdayDelivery – Saturday delivery // only for yhe locality that id opened on Saturday
Observations - observation
PackageContent – package content
CustomString – reference customer series
SenderReference1 – reference sender 1
RecipientReference1 – reference recipient 1
RecipientReference2 – reference recipient 2
InvoiceReference – reference invoice

It returns:

Generated barcode

Example:

```
$json='{
        "DeliveryPudoPoint": 114125
        "Sender":{"LocationId":28244},
        "Recipient":{
  "Name": "Gheorghita Vasile",
  "ContactPerson": "Gheorghita Vasile",
  "PhoneNumber": "0740123123",
  "Email": "Vasile@gmail.com"
},
 "Parcels": 1,
 "Envelopes": 0,
 "TotalWeight": 166,
 "ServiceId": 38,
 "DeclaredValue": 0,
 "CashRepayment": 0,
 "BankRepayment": 0,
 "OtherRepayment": "string",
 "BarCodeRepayment": "string",
 "PaymentInstrumentId": 0,
 "PaymentInstrumentValue": 0,
 "HasTertReimbursement": false,
 "OpenPackage": true,
 "PriceTableId": 0,
 "ShipmentPayer": 1,
 "ShippingRepayment": 0,
 "SaturdayDelivery": true,
 "MorningDelivery": true,
 "Observations": "string",
 "PackageContent": "string",
 "CustomString": "",
 "ParcelCodes": [
  {
```

```
    "Code": "0",
    "Type": 1,
    "Weight":25,
    "Length":20,
    "Width":20,
    "Height":10,
    "ParcelContent":"briefcase"

   }
  ]
}
```

```
$result=$urgent->CallMethod('Awbs', $json, 'POST', $token);
if ($result['status']!="200") {
        echo("<b class='bad'>POST Awb: </b>".$result['message']);
}
else {
        $data=$result['message'];
        echo "<b class='good'>POST Awb: </b>".$dat
 }
```

It returns: Main AWB

9.4 Generating transport waybills with AWB range

The **Awbs** method, called through POST, is used to add a new AWB with a number assigned

You send:

Sender

SenderClientId – In all cases it is null. It's a facility created for those who want to create AWBs in the name of another customer.

TertiaryClientId – tertiary id, if the shipment is made in another customer's account

LocationId – pick up point id (sender warehouse)

Recipient - Consignee

Name – recipient name

CountyId – county id

CountyName: - county name

LocalityId – locality id

LocalityName - locality name

StreetId – street id

StreetName – street name // Optional

BuildingNumber – street number

AddressText – address

ContactPerson – contact person

PhoneNumber – contact phone number
Email – recipient e-mail address
CodPostal– postal code of the consignee

Parcels – number of parcels
Envelopes – number of envelopes. Maximum of 9
TotalWeight – approximate total weight
DeclaredValue – declared value for insurance
CashRepayment – cash repayment (the sum returned to the sender in an envelope)
BankRepayment – bank repayment (the sum returned to the sender in the collector account)
OtherRepayment – other repayment
OpenPackage – open package
ServiceId: - 34 for totalweight 0 ≤ 31 kg ; 35 for 31 kg < totalweight ≤ 50 ; 50 for totalweitght > 50 kg
PriceTableId – price id
ShipmentPayer – shipment payer. Send 1 for sender and 2 for recipient
SaturdayDelivery – Saturday delivery
Observations - observation
PackageContent – package content
CustomString – reference customer series
SenderReference1 – reference sender 1
RecipientReference1 – reference recipient 1
RecipientReference2 – reference recipient 2
InvoiceReference – reference invoice
CodPostal– postal code for delivery point

It returns:
Generated barcode

Example:

```
$json='{
        "Sender":{"LocationId":28244},
        "Recipient":{
                "Name":"Ion Popescu",
                "CountyId":33,
                "LocalityId":161,
                "StreetId":0,
                "StreetName":"Str. Ionescu",
                "BuildingNumber":"25",
                "AddressText":"Bl. 129, Sc. 3, Ap. 4",
                "ContactPerson":" Ion Popescu ",
                "PhoneNumber":"072111222",
                "CodPostal": "string"
                "Email":"a@a.com"
        },
 "Parcels": 1,
 "Envelopes": 0,
```

```
"TotalWeight": 1,
"ServiceId": 34,
"DeclaredValue": 0,
"CashRepayment": 0,
"BankRepayment": 0,
"OtherRepayment": "string",
"BarCodeRepayment": "string",
"PaymentInstrumentId": 0,
"PaymentInstrumentValue": 0,
"HasTertReimbursement": false,
"OpenPackage": true,
"PriceTableId": 0,
"ShipmentPayer": 1,
"ShippingRepayment": 0,
"SaturdayDelivery": true,
"MorningDelivery": true,
"Observations": "string",
"PackageContent": "string",
"CustomString": "PLMI10875236",
"BarCode": "PLMI10875236",
"ParcelCodes": [
 {
  "Code": "XXXI10875236",
  "Type": 1,
  "Weight":1,
  "Length":10,
  "Width":10,
  "Height":10,

 }
],
"ValidationDate": "2019-10-24T12:33:12.035Z",
"ShippingCost": {
 "BaseCost": 0,
 "ExtraKmCost": 0,
 "WeightCost": 0,
 "InsuranceCost": 0,
 "SpecialCost": 0,
 "RepaymentCost": 0,
 "Subtotal": 0,
"Tax": 0,
 "GrandTotal": 0
},
"Status": "string",
"SenderReference1": "string",
"RecipientReference1": "string",
"RecipientReference2": "string",
"InvoiceReference": "string",
"Length": 0,
"Width": 0,
```

```
  "Height": 0
}
```

```
$result=$urgent->CallMethod('Awbs', $json, 'POST', $token);
if ($result['status']!="200") {
        echo("<b class='bad'>POST Awb: </b>".$result['message']);
}
else {
        $data=$result['message'];
        echo "<b class='good'>POST Awb: </b>".$data;
}
```

It returns: Main AWB

9.5 Deleting a transport note

The **Awbs** method deletes a transport note that has no checkpoint. It is a DELETE type method.

You send: barCode – the transport note number

It returns: true – if successful
false – if the transport note doesn't exist or has a checkpoint

Example:

```
$resultDelete=$urgent->CallMethod('Awbs?barCode= 804713464', $json="", 'DELETE', $token);
if ($resultDelete ['status']!="200") {
        echo("<b class='bad'>Awb deleted : </b>".$resultDelete ['message']);
}
else {
        $data=$resultDelete ['message'];
}
```

9.6 Get routing details by address

The GetRoutingAddress method, called through POST, is used to get routing details by address.

You send:


```
{
 "TotalWeight": 1,
 "Sender": {
 "CountyName": "IASI",
 "LocalityId": 0,
 "LocalityName": "IASI",
 "ZipCode": "700259"
 },
 "Recipient": {
 "CountyName":  "CONSTANTA",
 "LocalityName": "CONSTANTA",
 "AddressText": "PALAS 22",
 "ZipCode": "900320"
 }
}
```


It returns:


```
{
 "TransitPriority": "A",
 "TransitRoute": "Brasov",
 "TransitCodes": "500",
 "DeliveryRoute": "C2-Constanta\r900 - 01-45-04"
}
```


9.7 Listing the transport notes of a given period

The **AWBS/ GETBYDATE** method, called through GET, is used to return the list of AWBs of a certain period.

You send :

FromDate - start date in mm-dd-yyyy format
ToDate - end date in mm-dd-yyyy format
pageNumber – page number
itemsPerPage – number of AWBs per page

It returns:

A list of AWBs, containing the following information:


```
[
 {
  "SenderClientId": null,
  "TertiaryClientId": null,
```

```
                    "Sender": {
                     "LocationId": 1005962049,
                     "Name": "ECOM TEST",
                     "CountyId": 0,
                     "CountyName": "Ilfov",
                     "LocalityId": 29445916,
                     "LocalityName": "Tunari",
                     "StreetId": 0,
                     "StreetName": "",
                     "BuildingNumber": "",
                     "AddressText": "nr: 32; Sos centura nr 32",
                     "ContactPerson": "Ionut Petraru",
                     "PhoneNumber": "",
                      "CodPostal": "string"
                     "Email": ""
                    },
                    "Recipient": {
                     "LocationId": 1,
                     "Name": "test",
                     "CountyId": 1,
                     "CountyName": "Bucuresti",
                     "LocalityId": 150,
                     "LocalityName": "BUCURESTI",
                     "StreetId": 0,
                     "StreetName": "",
                     "BuildingNumber": "",
                     "AddressText": ", dsgfggfhgfhg",
                     "ContactPerson": "",
                     "PhoneNumber":  "",
                     "CodPostal": "string"
                     "Email": ""
                    },
"Parcels": 2,
"Envelopes": 0,

"TotalWeight": 344,
"ServiceId": 1,
"DeclaredValue": 0,
"CashRepayment": 0,
"BankRepayment": 0,
"OtherRepayment": "string",
"BarCodeRepayment": "string",
"PaymentInstrumentId": 0,
"PaymentInstrumentValue": 0,
"HasTertReimbursement": false,
"OpenPackage": true,
"PriceTableId": 0,
"ShipmentPayer": 1,
"ShippingRepayment": 0,
```

```
 "SaturdayDelivery": true,
 "MorningDelivery": true,
 "Observations": "string",
 "PackageContent": "string",
 "CustomString": "URGC10875236",
 "BarCode": "URGC10875236",
 "ParcelCodes": [
  {
    "Code": "URGC10875236",
    "Type": 1,
    "Weight":166.67,
    "Length":100,
    "Width":100,
    "Height":100,

  }
 ],
 "ValidationDate": "2019-10-24T12:33:12.035Z",
 "ShippingCost": {
  "BaseCost": 0,
  "ExtraKmCost": 0,
  "WeightCost": 0,
  "InsuranceCost": 0,
  "SpecialCost": 0,
  "RepaymentCost": 0,
  "Subtotal": 0,
 "Tax": 0,
  "GrandTotal": 0
 },
 "Status": "string",
 "SenderReference1": "string",
 "RecipientReference1": "string",
 "RecipientReference2": "string",
 "InvoiceReference": "string",
 "Length": 0,
 "Width": 0,
 "Height": 0
}
```


Example:

```
$resultGetByDate=$urgent->CallMethod('Awbs/GetByDate?FromDate=02-08-2016&ToDate=02-
08-2016&pageNumber=1&itemsPerPage=100', $json="",'GET', $token);

if ($resultGetByDate['status']!="200") {
        echo"<b class='bad'>GET GetByDate : </b>".$resultGetByDate['message'];
}
else {
        $dataGetByDate=$resultGetByDate['message'];
```

```
        echo "<b class='good'>GET GetByDate : </b>".$dataGetByDate;
}
```

9.8 Listing transport notes information for an order

The **Awbs** method, called through GET, returns information about the sent notes.

You send:

barCode – the AWB

or

orderId – order number

Important! Do not send both.

It returns:


```
[
 {
  "SenderClientId":   null,
  "TertiaryClientId": null,
  "Sender": {
   "LocationId": 1005962049,
   "Name": "ECOM TEST",
   "CountyId": 0,
   "CountyName": "Ilfov",
   "LocalityId": 29445916,
   "LocalityName": "Tunari",
   "StreetId": 0,
   "StreetName": "",
   "BuildingNumber": "",
   "AddressText": "nr: 32; Sos centura nr 32",
   "ContactPerson": "Ionut Petraru",
   "PhoneNumber": "",
   "CodPostal": "string"
   "Email": ""
  },
  "Recipient": {
   "LocationId": 1,
   "Name": "test",
   "CountyId": 1,
   "CountyName": "Bucuresti",
   "LocalityId": 150,
   "LocalityName": "BUCURESTI",
   "StreetId": 0,
   "StreetName": "",
   "BuildingNumber": "",
   "AddressText": ", dsgfggfhgfhg",
   "ContactPerson": "",
   "PhoneNumber": "",
   "CodPostal": "string"
```

```
    "Email": ""
  },
  "Parcels": 0,
  "Envelopes": 1,
  "TotalWeight": 1,
  "ServiceId": null,
  "DeclaredValue": 0,
  "CashRepayment": 120,
  "BankRepayment": 0,
  "OtherRepayment": "",
  "PaymentInstrumentId": 0,
  "PaymentInstrumentValue": 0,
  "HasTertReimbursement": false,
  "OpenPackage": false,
  "PriceTableId": 17430,
  "ShipmentPayer": 3,
  "SaturdayDelivery": false,
  "MorningDelivery": false,
  "Observations": "",
  "PackageContent": "",
  "CustomString": "",
  "BarCode": "804523201",
  "ParcelCodes": null,
  "ValidationDate": "2016-02-08T06:38:06.267",
  "ShippingCost": null,
  "Status": "Tiparit",
  "SenderReference1": "",
  "RecipientReference1": "",
  "RecipientReference2": "",
  "InvoiceReference": ""
```


Example:

For an AWB:

```
$resultAwbDetails=$urgent->CallMethod('Awbs?barCode=804737910', $json="", 'GET',
$token);
if ($resultAwbDetails['status']!="200") {
        echo"<b class='bad'>GET awb details: </b>".$resultAwbDetails['message'];
}
else {
        $data=$resultAwbDetails['message'];
        echo "<b class='good'>GET awb details By AWB: </b>".$data;
}
```

For an order:

```
$resultAwbDetails=$urgent->CallMethod('Awbs?orderId=201243788', $json="", 'GET',
$token);
if ($resultAwbDetails['status']!="200") {
        echo"<b class='bad'>GET awb details: </b>".$resultAwbDetails['message'];
```

```
}
else {
        $data=$resultAwbDetails['message'];
        echo "<b class='good'>GET awb details By order ID : </b>".$data;
}
```

9.9 Printing transport notes in PDF or HTML format

The **AwbDocuments** method, called through GET, is used to print the transport notes in PDF or HTML. You may specify either A4 or Label

You send :

barCode - a list (json) of AWBs to print
type – PDF or HTML
format - 0 - A4; 1 - Label 10x14
printMainOnce - 0 or nothing - Main AWB will print twice; 1 - Main AWB will print once, 2 - Main AWB will print once format label 10x14

It returns:

The code in base64

Example:

```
header("Content-type:application/pdf");

$awb=array(804419419,804418863);
$jsonAwb=json_encode($awb);


$result=$urgent-
>CallMethod('AwbDocuments?barCodes='.$jsonAwb.'&type=PDF&format=1&printMainOnce=1'
, $json="", 'GET', $token);
if ($result['status']!="200") {
        echo("<b class='bad'>Awb Documents: </b>".$result['message']);
}
else {
        $data=$result['message'];
        echo base64_decode($data);
}
```

9.10 Tracking the shipments with return or redirected AWB's

The **AwbTrace/WithRedirect**, method, called through GET is used for tracking shipments of return or redirected AWB's

You send :

barCode - a list of searched AWB's

It returns:

Information with the events . the list has the following information:

Code – AWB number
Type – package =1 sau envelope =0
MeasuredWeight - weight
VolumetricWeight – volumetric weight

ConfirmationName – person who confirmed the shippment
Observation – observations regarding the awb
ResponseCode – the return or redirected AWB

Event

Date – event date
Description – event description
LocalityName – locality name

Example:

```
$awbList=array(804115458, 804346669);
$jsonAwb=json_encode($awbList);
$resultTrace = $urgent->CallMethod('AwbTrace/WithRedirect?barCode='.$jsonAwb, $json="",
'GET', $token);
if ($resultTrace['status'] != "200") {
        echo"<b class='bad'>GET ERROR Trace: </b>".$resultTrace['message'];
}
else {
        $result = $resultTrace['message'];
        echo "<b class='good'>GET Trace: </b>".$result;
}
```

9.11 Listing returning AWB's

The **AwbRetur** method, called through GET is used for pulling up all of the AWB's the return to the customer in the specified date.

You send :

date - the date in the following format yyyy-mm-dd

It returns:

Awb – information regarding the AWB
Sender - Sender
LocationId – pick up point id

Name – sender name
CountyId – county id
CountyName – county name
LocalityId – locality id
LocalityName – locality name
StreetId – street id
StreetName – street name, exactly as found in the Cargus nomenclature
BuildingNumber – street number
AddressText – address
ContactPerson – contact person for the current order
PhoneNumber – contact phone number
Email – e-mail address

Recipient – Consignee
Name – recipient name
CountyId – county id
LocalityId – locality id
StreetId – street id
StreetName – street name // Optional
BuildingNumber – street number
AddressText – address
ContactPerson – contact person
PhoneNumber – contact phone number
Email – recipient e-mail address

Parcels – number of parcels
Envelopes – number of envelopes. Maximum of 9
TotalWeight – approximate total weight
DeclaredValue – declared value for insurance
CashRepayment – cash repayment (the sum returned to the sender in an envelope)
BankRepayment – bank repayment (the sum returned to the sender in the collector account)
OtherRepayment – other repayment
OpenPackage – open package
HasTertReimbursement – if true the reimbursement comes back to TertiaryClientId and it's obligatory
PriceTableId – price id
ShipmentPayer – shipment payer. Send 1 for sender and 2 for recipient
SaturdayDelivery – Saturday delivery
MorningDelivery – morning delivery
Observations - observations
PackageContent – package content
CustomString – reference customer series
SenderReference1 – sender reference 1
RecipientReference1 – recipient reference 1
RecipientReference2 – recipient reference 2
InvoiceReference – invoice reference

ParcelCodes – list that contains the main AWB and the secondary ones that are attached

Code – code of the envelope or parcel.
Type - envelope=1 or parcel=0
ValidationDate – date that it entered in the system
ShippingCost – details of the shipping cost
Status – shipping status
AwbRetur – details about the returning AWB
Sender - sender name
LocationId – pick up point id
Name – sender name
CountyId – county id
CountyName – county name
LocalityId – locality id
LocalityName – locality name
StreetId – street id
StreetName – street name, exactly as found in the Cargus nomenclature
BuildingNumber – street number
AddressText – address
ContactPerson – contact person for the current order
PhoneNumber – contact phone number
Email – e-mail address

Recipient – Consignee
LocationId – pick up point id
Name – recipient name
CountyId – county id
LocalityId – locality id
StreetId – street id
StreetName – street name // Optional
BuildingNumber – street number
AddressText – address
ContactPerson – contact person
PhoneNumber – contact phone number
Email – recipient e-mail address

Parcels – number of parcels
Envelopes – number of envelopes. Maximum of 9
TotalWeight – approximate total weight
DeclaredValue – declared value for insurance
CashRepayment – cash repayment (the sum returned to the sender in an envelope)
BankRepayment – bank repayment (the sum returned to the sender in the collector account)
OtherRepayment – other repayment
OpenPackage – open package

HasTertReimbursement – if true the reimbursement comes back to TertiaryClientId and it's obligatory
PriceTableId – price id
ShipmentPayer – shipment payer. Send 1 for sender and 2 for recipient
SaturdayDelivery – Saturday delivery
MorningDelivery – morning delivery
Observations - observations
PackageContent – package content
CustomString – reference customer series
SenderReference1 – sender reference 1
RecipientReference1 – recipient reference 1
RecipientReference2 – recipient reference 2
InvoiceReference – invoice reference
ParcelCodes – list that contains the main AWB and the secondary ones that are attached

Code – code of the envelope or parcel.
Type - envelope=1 or parcel=0
ValidationDate – date that it entered in the system
ShippingCost – details cost of the shipping
Courier – courier
TimestampDeparture – date when the courier scans it
ClientObservation – observations for the return AWB

Example:

```
$result=$urgent->CallMethod('AwbRetur?data='.$Date, $json="",'GET',$token);
if ($result['status']!="200")
{
        echo("<b class='bad'>POST Awb: </b>".$result['message']);
}
else
{
        $data=$result['message'];
        echo "<b class='good'>POST Awb: </b>".$data;
}
```

9.12 Check last envent from a set interval

The **AwbTrace/GetDeltaEvents** method, called through GET is used for checking the staus from a set interval .

You send :

FromDate - from date (format :mm-dd-yyyy)
ToDate - to date (format :mm-dd-yyyy)

It returns:

Date – date of creation of AWB
Barcode – AWB number
Sender – sender
Receiver – receiver
FromLocality – sender locality
ToLocality – receiver locality
CustomString – reference customer series
RepaymentDate – repayment date
RepaymentId – repayment id
RepaymentValue – repayment amount
DeductionDate – date of payment order
DeductionId – number of payment order
StatusExpression – detailed status
ConfirmationDate – confirmation date
ConfirmationPersonaName – confirmation name
Packages
Code – AWB number
Type – package =1 sau envelope =0
MeasuredWeight - weight
VolumetricWeight – volumetric weight
Events – Events
Date – date
EventId – id-ul evenimentului
Description – description of event
LocalityName – locality name where the event was added

Example:

```
$result=$urgent-
>CallMethod('AwbTrace/GetDeltaEvents?FromDate='.$FromDate.'&ToDate='.$ToDate,
$json="",'GET',$token);
if ($result['status']!="200")
{
        echo("<b class='bad'>POST Awb: </b>".$result['message']);
}
else
{
        $data=$result['message'];
```

```
        echo "<b class='good'>POST Awb: </b>".$data;
}
```

9.13 Display confirmation picture

The **AwbScan**, method, called through GET is used to puul up pictures for confirmation

You send :

barCode - AWB code

It returns:

Code in base64

Example:

```
$result=$urgent->CallMethod('AwbScan?barCodes='804419419, $json="", 'GET', $token);
if ($result['status']!="200") {
        echo("<b class='bad'>Awb Scan: </b>".$result['message']);
}
else {
        $data=$result['message'];
        header("Content-type:image");
        echo base64_decode($data);
}
```

10 Order management (courier request)

10.1 Launching or cancelling an order for a pick up point

The **Orders** method, called through PUT, validates or cancels an order from a pick up point.

You send:

LocationId – pick up point id – 0 for headquarters
action – action id . if you send it, 0 cancels and 1 validates
PickupStartDate – star date and hour for pick up
PickupEndDate – end date and hour for pick up

It returns:

Order number

Example:

```
$resultPutOrder = $urgent-
>CallMethod('Orders?locationId=201008098&action=0&PickupStartDate=2016-02-
16T12:16:15.827Z&PickupEndDate=2016-02-16T12:19:15.827Z',$json="",'PUT',$token);
if ($resultPutOrder['status']!="200") {
        echo("<b class='bad'>PUT Order: </b>".$resultPutOrder['message']);
}
else {
        $dataPutOrder=$resultPutOrder['message'];
        echo "<b class='good'>PUT Order: </b>".$dataPutOrder;
}
```

10.2 Launching or cancelling all orders

The **Orders/PutAll** method validates or cancels all the orders, on all pick up points.

You send:

action – if you send, 0 cancels and 1 validates
PickupStartDate – start date and hour for pick up
PickupEndDate – end date and hour for pick up

It returns:

Order number

Example:

```
$resultPutOrder = $urgent->CallMethod('Orders/PutAll?action=1&PickupStartDate=2016-02-
16T12:16:15.827Z&PickupEndDate=2016-02-16T12:19:15.827Z',$json="",'PUT',$token);
if ($resultPutOrder['status']!="200") {
        echo("<b class='bad'>PUT Order: </b>".$resultPutOrder['message']);
}
else {
        $dataPutOrder=$resultPutOrder['message'];
        echo "<b class='good'>PUT Order: </b>".$dataPutOrder;
}
```

10.3 Listing orders information for a pick up point

The **Orders** method, called through GET, offers information about the orders from a specific pick up point.

You send:

LocationId – pick up point id – 0 for headquarters

status – order status – 0 current order / 1 validated order
pageNumber – number of pages displayed
itemsPerPage – number of orders per page

It returns:

An order list. An order contains the following information:

OrderId – order number
LocationId – pick up point id
PickUpSiteName – pick up point address
PickupStartDate – start date and hour for order pick up
PickupEndDate - end date and hour for order pick up
if the order status is 0 then the order generation date is displayed.
NoParcel – total number of parcels
NoEnvelop – total number of envelopes
TotalWeight – order total weight
NoAwb – order total number of AWBs
Observations – order observation
PackageContent - package content
CreationDate – date and hour of order generation
ValidationDate – date and hour of order shipping
ProcessedDate - date of processed order
OrdStatus – order status
AdresaExp – sender address
Email – this should be the e-mail of the pick up point
OrderType – order type (on which channel it has been generated)

Example:

```
$resultOrders = $urgent-
>CallMethod('Orders?locationId=0&status=1&pageNumber=1&itemsPerPage=100', $json="",
'GET', $token);

if ($resultOrders['status'] != "200") {
        echo("<b class='bad'>GET Order: </b>".$resultOrders['message']);
}
else {
        $dataOrders=$resultOrders['message'];
        echo "<b class='good'>GET Orders: </b>".$dataOrders;
}
```

10.4 Listing order information for a given period

The **Orders/GetByDate** method is called through GET and offers information about the orders in a certain period.

You send:

FromDate - start date in yyyy-mm-dd format

ToDate - end date in yyyy-mm-dd format
pageNumber – number of page displayed
itemsPerPage – number of AWBs per page

It returns:

An order list. An order contains the following information:

OrderId – order number
LocationId – pick up point id
PickUpSiteName – pick up point address
PickupStartDate – start date and hour for order pick up
PickupEndDate - end date and hour of order pick up
If the order has the status 0 then the order generation date is displayed
NoParcel – total number of parcels
NoEnvelop – total number of envelopes
TotalWeight – total weight of the shipment
NoAwb – total number of AWBs
Observations – order observation
PackageContent - parcel content
CreationDate – date and hour of order creation
ValidationDate – date and hour of order shipping
ProcessedDate - order processed date
OrdStatus – order status
AdresaExp – sender address
Email – this should be the pick up point e-mail address
OrderType – order type (on which channel it has been generated)

Example:

```
$resultOrdersGetByDate =$urgent->CallMethod('Orders/GetByDate?FromDate=2016-02-
15&ToDate=2016-02-19&pageNumber=1&itemsPerPage=100', $json="",'GET', $token);


if ($resultOrdersGetByDate ['status']!="200") {
        echo"<b class='bad'>GET Result Orders ByDate :
</b>".$resultOrdersGetByDate['message'];
}
else {
        $dataOrdersGetByDate = $resultOrdersGetByDate['message'];
        echo "<b class='good'>GET Result Orders ByDate: </b>".$dataOrdersGetByDate;
}
```

10.5 Listing order information using its number

The **Orders/GetByOrderId** method is called through GET and offers information about the orders searched using its number

You send:

orderId – order number

It returns:

Information regarding that order :

OrderId – order number
LocationId – pick up point id
PickUpSiteName – pick up point address
PickupStartDate – start date and hour for order pick up
PickupEndDate - end date and hour of order pick up
If the order has the status 0 then the order generation date is displayed
NoParcel – total number of parcels
NoEnvelop – total number of envelopes
TotalWeight – total weight of the shipment
NoAwb – total number of AWBs
Observations – order observation
PackageContent - parcel content
CreationDate – date and hour of order creation
ValidationDate – date and hour of order shipping
ProcessedDate - order processed date
OrdStatus – order status
AdresaExp – sender address
Email – this should be the pick up point e-mail address
OrderType – order type (on which channel it has been generated)

Example:

```
$resultOrders = $urgent->CallMethod('Orders/GetByOrderId?orderId=201245248', $json="",
'GET', $token);

if ($resultOrders['status'] != "200") {
        echo("<b class='bad'>GET Order: </b>".$resultOrders['message']);
}
else {
        $dataOrders=$resultOrders['message'];
        echo "<b class='good'>GET Orders: </b>".$dataOrders;
}
```

11 Tracking cash on delivery

The **CashAccount** method is used to display the information about repayment in collector account.

11.1 List cash on delivery from a set period of time

The **CashAccount/GetByDate** method is called through GET and pulls up information from a set period of time .

You send:

FromDate - start date in yyyy-mm-dd format
ToDate - end date in yyyy-mm-dd format
Token – identify the customer

It returns:

Date – emission date
BarCode - AWB code
Sender – sender
Receiver – receiver
FromLocality – sender locality
ToLocality – receiver locality
CustomString – reference customer series
RepaymentDate – repayment date
RepaymentId – repayment id
RepaymentValue – repayment amount
DeductionDate – date of payment order
DeductionId – number of payment order

Example:

```
$resultCashAccount =$urgent->CallMethod('CashAccount/GetByDate?FromDate=2016-03-24&ToDate=2016-03-25', $json="",'GET', $token);


if ($resultCashAccount ['status']!="200") {
        echo"<b class='bad'>GET CashAccount  : </b>".$resultCashAccount['message'];
}
else {
        $dataCashAccount =$resultCashAccount['message'];
        echo "<b class='good'>GET Get Cash Account  : </b>".$dataCashAccount;
}
```

11.2 Listing refunds after a certain date

Metoda **CashAccount/GetByDeductionDate** method is called through GET and pulls up information payment orders from a specific date .

You send:

DeductionDate – date of payment order

Token_identifythecustomer

It returns:

Date – emission date
BarCode - AWB code
Sender – sender
Receiver – receiver
FromLocality – sender locality
ToLocality – receiver locality
CustomString – reference customer series
RepaymentDate – repayment date
RepaymentId – repayment id
RepaymentValue – repayment amount
DeductionDate – date of payment order
DeductionId – number of payment order

Example:

```
$result=$urgent->CallMethod('CashAccount/GetByDeductionDate?DeductionDate=2016-03-29',$json="",'GET',$token);

print_r($result);


if ($result['status']!="200") {
	echo"<b class='bad'>GET CashAccount?GetByDeductionDate: </b>".$result['message'];
}
else {
	$data=$result['message'];
	echo "<b class='good'>GET CashAccount?GetByDeductionDate: </b>".$data;
}
```

11.3 Refund listing according to a certain barcode

The **CashAccount** method is used to display the information about repayment in collector account.

You send:

Token – identify the customer
BarCode - AWB code

It returns:

Date – emission date
BarCode - AWB code
Sender – sender
Receiver – receiver

FromLocality – sender locality
ToLocality – receiver locality
CustomString – reference customer series
RepaymentDate – repayment date
RepaymentId – repayment id
RepaymentValue – repayment amount
DeductionDate – date of payment order
DeductionId – number of payment order

Example:

```
$resultCashAccount =$urgent->CallMethod('CashAccount?barCode=804373743', $json="",'GET',
$token);


if ($resultCashAccount ['status']!="200") {
        echo"<b class='bad'>GET CashAccount  : </b>".$resultCashAccount['message'];
}
else {
        $dataCashAccount =$resultCashAccount['message'];
        echo "<b class='good'>GET Get Cash Account  : </b>".$dataCashAccount;
}
```

12 Invoices

12.1 Invoices list

The **Invoices** method returns information about the invoices in a certain period.

You send :

FromDate - start date in yyyy-mm-dd format
ToDate - end date in yyyy-mm-dd format
pageNumber – number of page displayed
itemsPerPage – number of AWBs per page

It returns:

InvoiceId - 2109644734,
Date – emission date
DueDate – invoice date
Series – invoice series
Number – invoice number

Value – value without VAT
Total – total
Closed – closed invoice
Balance - balance
CurrencyId – currency

Example:

```
$resultInvoices =$urgent->CallMethod('Invoices?FromDate=2016-01-01&ToDate=2016-02-19&pageNumber=1&itemsPerPage=100', $json="",'GET', $token);
if ($resultInvoices ['status']!="200") {
        echo"<b class='bad'>GET Invoices  : </b>".$resultInvoices['message'];
} else {
        $dataInvoices =$resultInvoices['message'];
        echo "<b class='good'>GET Invoices  : </b>".$dataInvoices;
}
```

12.2 Printing invoices in PDF format

The **InvoiceDocuments** method is a GET type method and returns a string that must be decoded in base64.

You send :

InvoiceId - the invoice id returned by the Invoices method

It returns:

The code in base64

Example:

```
header("Content-type:application/pdf");
$resultInvoiceDocuments = $urgent->CallMethod('InvoiceDocuments?InvoiceId=1010219190', $json="",'GET', $token);
if ($resultInvoiceDocuments['status'] != "200") {
        //echo"<b class='bad'>GET InvoiceDocuments  : </b>".$resultInvoiceDocuments['message'];
}
else {
        $dataInvoiceDocuments = $resultInvoiceDocuments['message'];
        //echo "<b class='good'>GET InvoiceDocuments  : </b>".$dataInvoiceDocuments;

        echo base64_decode($dataInvoiceDocuments);
}
```

13 Work flow for integration

The PickupLocations is called to add: PL1 lift point. Id returned idPL1.

1. You can use the Localities method to pick up the local id
2. You can use the Streets method to get the Street id

• Note that a working point has AutomaticEOD (end of day)

• Call Awbs to add awb. The sender uses idPL1. At this point you have an open command on idPL1.

The order closes when:

•

• Call Orders with PUT for the idPL1 lift point with action = 1 to launch the command.

or

Touch AutomaticEOD to the hoist point

14 Implementation conditions for the multipiece service

The multipiese service doesn't allow:

- a taxable weight greater than 31 kg per piece;
- more than 15 pieces per shipment;
- a weight greater than 465 Kg for a shipment.

15 ANNEX

SERVICEID

Id	Name
34	Economic Standard
35	Standard Plus
36	Palet Standard
38	PUDO Delivery
39	Multipiece

Status: 0 – open order
1 – closed order

OrderType: 1 – online –generated on Webexpress
2 – by thelephone
3 – external – generated by API
4 – pick up – generated by ” Pick Up From Another Location” in Webexpress

5 - email- generated by e-mail
6 – predetermined –generated in the internal system every day