

Medex API System Integration Manual

Introduction

General Review

API (Application Programming Interface, further shortened as API) is intended to accomplish the following tasks:

- To automatically receive orders from a client 24/7
- To automatically send reshipment requests
- To automatically receive tracking numbers
- To automatically receive tracking numbers statuses once the shipping service provides them.
- To automatically receive tracking numbers and their statuses for any orders.

The SOAP and XML request formats are available with API.

Why do I need this?

API will save your staff working day time as all orders will be received in a fully automatic mode. Orders will have their tracking numbers and statuses updated automatically.

How do I start?

Contact your Medex Support Manager and let him/her know you are willing to use API. Your manager will provide you with your personal login and password to access API. Then you can start to set up API.

How do I test API?

When you get connected to API, your account is set to test mode on default. Your orders are accepted but not forwarded to Orders Medex Ltd system. When you are through with testing your codes and decide to switch from test mode to regular work please inform Medex Support Team (see the contact information below) and you will be switched.

If any questions or issues arise when starting API

If any questions or issues arise when connecting to API, feel free to contact Medex API Tech Support (see the contact information below) or send a message to your Medex Support Manager.

What should I use - SOAP or XML?

You may choose the method according to your IT group skills and level. If your web site programs are mainly written in PHP we recommend to use SOAP method. For PERL as it has contradictions with SOAP libraries we recommend to use XML method. Please consider these recommendations as our personal opinion that may not always coincide with yours.

Uploading Orders

Products and shipping methods IDs coordination

Since API products and shipping methods are transferred by their IDs not names we should make ID products and shipping methods synchronization between Medex system and yours. This can be done in the following ways.

Option 1. Using Medex IDs

To process with this option please take Medex IDs from http://system.medextld.com/api_ids.php, replace your IDs by Medex IDs and send them by API.

Option 2. Using your IDs

If you consider the first option too complicated you are welcome to provide us with your own IDs so we could synchronize the IDs as a part of the API project.

For this please make the Excel or csv file with the following columns:

- Your product or pack ID
- Product name
- Qty unless the product or pack ID defines it. Input "1" if the product or pack ID defines the qty.

If one product ID corresponds to several products (of the same ID pack), input the products one by one with the same ID.

The 2nd file required should include the following columns:

- Your shipping method ID
- Your shipping method name

These files should be sent via e-mail to your Medex Support Manager or to the Medex API Tech Support using the following e-mail api@medextld.com. Kindly indicate your API login in the message.

Uploading orders via SOAP

PHP code example to transfer orders via SOAP

```
<?php
$client = new SoapClient("http://api.medextld.com/webservice.wsdl", array('trace' => 1));
$objAuth = new stdClass();
$objAuth->username = "%api_login%";
$objAuth->password = "%api_password%";
$objOrder = new stdClass();
$objOrder->authentication = $objAuth; // authorization
$objOrder->order_customer = "Levafan Petrovich"; // customer's First & Last Name
$objOrder->order_country = "USA"; // customer's country
$objOrder->order_address = "Customer Address"; // Delivery address
$objOrder->order_number = "12312"; // Your system order ID
$objOrder->order_city = "usa"; // Delivery city
$objOrder->order_state = ""; // if appropriate - input a state for delivery. If not appropriate equals to '-'(dash)
$objOrder->order_zip = ""; // delivery address Zip code
$objOrder->shipping_id = 10; // shipping method ID
$objRequest->products = array(); // create products array
$objProduct = new stdClass(); // create object for a product
$objProduct->product_id = 479; // product ID
$objProduct->quantity = 15; // quantity
$objOrder->products[] = $objProduct; // add a product to the order
$objProduct->product_id = 478;
$objProduct->quantity = 10;
$objOrder->products[] = $objProduct; // add the second product to the array
$response = $client->sendOrder($objOrder);
if($false == is_soap_fault($response)){
    print_r($response);
}
else{
    echo "error code: ". $response->faultstring . "<br>";
    echo "error message: ". $response->detail . "<br>";
}
?>
```

Uploading orders via XML

To upload orders using XML the xmlData variable should be sent to <http://api.medextld.com/webservicexml/sendOrder> by POST method with the following XML text.

```
<mdx>
<authentication>
<username>Your API login</username>
<password>Your API password</password>
</authentication>
<orderdata>
<order_customer> customer's First Name_Last Name </order_customer>
<order_address> delivery address </order_address>
<order_state> state (for USA), otherwise '-'(dash)</order_state>
<order_number> order number </order_number>
<order_city> city </order_city>
<order_country> country </order_country>
<order_zip> zip code </order_zip>
<shipping_id> shipping ID </shipping_id>
<products>
<product>
<product_id> product ID</product_id>
<quantity> quantity</quantity>
</product>
<product_id> product ID</product_id>
<quantity> quantity</quantity>
</product>
</products>
</orderdata>
</mdx>
```

WARNING! If product ID contains product with its qty then quantity should contain any number different from zero. If the request is completed successfully the server will reply with the following message

```
<mdx>
<status_ok>Ok</status_ok>
</mdx>
If an error occurs you will receive
<mdx>
<error>
<error_code> error code</error_code>
<error_str> error text</error_str>
</error>
</mdx>
```

Please note that once an error occurs orders are not uploaded to API. WARNING! All data must have UTF-8 encoding to be transferred!

View error codes below

PHP order example

```
<?php
$xml["xmlData"] = "<mdx><authentication><username><password>test_test</password></authentication>";
$xml["xmlData"] = "<orderdata>";
$xml["xmlData"] = "<order_customer>Djohn Mrakov</order_customer>";
$xml["xmlData"] = "<order_address>Big Lanch st., 10</order_address>";
$xml["xmlData"] = "<order_state>New York</order_state>";
$xml["xmlData"] = "<order_number>test_number_111</order_number>";
$xml["xmlData"] = "<order_city>New York</order_city>";
$xml["xmlData"] = "<order_country>usa</order_country>";
$xml["xmlData"] = "<order_zip>187654</order_zip>";
$xml["xmlData"] = "<shipping_id>3</shipping_id>";
$xml["xmlData"] = "<products>";
$xml["xmlData"] = "<product><product_id>458</product_id><quantity>11</quantity></product>";
// $xml["xmlData"] = "<product><product_id>10</product_id><quantity>3</quantity></product>";
$xml["xmlData"] = "<product_id>225</product_id><quantity>24</quantity></product>";
$xml["xmlData"] = "</orderdata>";
$xml["xmlData"] = "</mdx>";
$connection = curl_init();
curl_setopt($connection, CURLOPT_URL, "http://api.medextld.com/webservicexml/sendOrder");
curl_setopt($connection, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($connection, CURLOPT_POST, 1);
curl_setopt($connection, CURLOPT_POSTFIELDS, $xml);
curl_setopt($connection, CURLOPT_RETURNTRANSFER, 1);
set_time_limit(10000);
$response = curl_exec($connection);
if(curl_errno($connection)) {
    print "Curl error: ". curl_error($connection);
}
curl_close($connection);
print_r($response);
?>
```

Error codes that might occur while uploading orders

- 100 - authorisation failure
- 1 - error in the order_customer field
- 2 - error in the order_address field
- 3 - error in the order_country field
- 4 - error in the order_city field
- 5 - error in the order_state field
- 6 - error in the shipping_id field
- 7 - error in the order_zip field
- 8 - error in the products array
- 9 - error in the order_number field
- 20 - an order with the same ID has already been uploaded

Reshipment request

Introduction

To request reshipment you should define order number and comments with the reason explaining reshipment request. All comments should be UTF-8 encoded.

Request via SOAP

```
<?php
$client = new SoapClient("http://api.medextld.com/webservice.wsdl", array('trace' => 1));
$objAuth = new stdClass();
$objAuth->username = "%api_login%";
$objAuth->password = "%api_password%";
$objReship = new stdClass();
$objReship->authentication = $objAuth;
$objReship->order_number = "123456";
$objReship->comment = "My Comment";
$response = $client->reshipOrder($objReship);
if($false == is_soap_fault($response)){
    print_r($response);
}
else{
    echo "error code: ". $response->faultcode . "<br>";
    echo "error message: ". $response->faultstring . "<br>";
}
?>
```

Request via XML

To request reshipments the xmlData variable should be sent to <http://api.medextld.com/webservicexml/reshipOrder> by POST method with the following XML text.

```
<mdx>
<authentication>
<username>Your API login</username>
<password>Your API password</password>
</authentication>
<order_number>318477</order_number>
<comment>Reshipment comment</comment>
</mdx>
The following XML reply will be sent by server:
<mdx>
<status_ok>Ok</status_ok>
</mdx>
If an error occurs the following response will be received:
<mdx>
<error>
<error_code>Error code</error_code>
<error_str>Error text</error_str>
</error>
</mdx>
```

Processing tracking number receiving requests

Notifications are transferred via POST plain/text method to the address indicated earlier by you.

XML

```
<mdx>
<order_number>Order number Tracking Number was requested for</order_number>
<trackings>
<tracking_id>tracking number</tracking_id>
<tracking_id>tracking number</tracking_id>
<tracking_id>...</tracking_id>
</trackings>
</mdx>
```

The program shows "ok" if the request is completed successfully. If the response is different, the system considers it as an error and it will try to transfer the requested tracking number again during the next tracking numbers transfer.

Warning! Before you launch the program send your receiver program address to Medex API Tech Support e-mail.

Processing requests of tracking number status updates

The system checks tracking number status automatically throughout the day and notifies you regularly. Notifications are transferred by POST plain/text to the address you indicated earlier.

XML

```
<mdx>
<trackings>
<tracking_short_state>Delivered</tracking_short_state>
<tracking_full_state>Notice Left</tracking_full_state>
</trackings>
</mdx>
```

The program shows "ok" if the request is completed successfully. If the response is different, the system considers it as an error and it will try to transfer the requested tracking number again during the next tracking numbers transfer.

Warning! Before you launch the program, send your receiver program address to Medex API Tech Support e-mail.

Receiving tracking number lists via API request

Introduction

Any order can be requested for tracking number list at any time. To process this request use one of the options below.

Requesting tracking numbers list via SOAP

```
<?php
$client = new SoapClient("http://api.medextld.com/webservice.wsdl", array('trace' => 1));
$objAuth = new stdClass();
$objAuth->username = "%api_login%";
$objAuth->password = "%api_password%";
$objData = new stdClass();
$objData->authentication = $objAuth;
$objData->order_number = 'order number';
$response = $client->getTrackings($objData);
if($false == is_soap_fault($response)){
    print_r($response);
}
else{
    echo "error code: ". $response->faultstring . "<br>";
    echo "error message: ". $response->detail . "<br>";
}
?>
```

Requesting tracking numbers list via XML

To request tracking number list by an order number the xmlData variable should be sent to <http://api.medextld.com/webservicexml/getTrackings> by POST method with the following XML text.

```
<mdx>
<authentication>
<username>Your API login</username>
<password>Your API password</password>
</authentication>
<tracking_id>tracking number</tracking_id>
</mdx>
```

Server will reply with the following XML text

```
<mdx>
<tracking>
<tracking_id>tracking number request was sent for</tracking_id>
<tracking_state>tracking status</tracking_state>
</tracking>
</mdx>
```

If an error occurs the following response will be received

```
<mdx>
<error>
<error_code>error code</error_code>
<error_str>error text</error_str>
</error>
</mdx>
```

Online tracking number check up

Introduction

Any tracking number can be checked at any time. To process this request use one of the options below.

Online check up via SOAP

```
<?php
$client = new SoapClient("http://api.medextld.com/webservice.wsdl", array('trace' => 1));
$objAuth = new stdClass();
$objAuth->username = "%api_login%";
$objAuth->password = "%api_password%";
$objTrack = new stdClass();
$objTrack->authentication = $objAuth;
$objTrack->tracking_id = "RM007202341IN";
$response = $client->checkTrackNow($objTrack);
if($false == is_soap_fault($response)){
    print_r($response);
}
else{
    echo "error code: ". $response->faultcode . "<br>";
    echo "error message: ". $response->faultstring . "<br>";
}
?>
```

Online check up via XML

To request tracking number status using XML the xmlData variable should be sent to <http://api.medextld.com/webservicexml/checkTrackNow> by POST method with the following XML text.

```
<mdx>
<authentication>
<username>Your API login</username>
<password>Your API password</password>
</authentication>
<tracking_id>tracking number</tracking_id>
</mdx>
```

Server will reply with the following XML text

```
<mdx>
<tracking>
<tracking_id>tracking number request was sent for</tracking_id>
<tracking_state>tracking status</tracking_state>
</tracking>
</mdx>
```

If an error occurs the following response will be received

```
<mdx>
<error>
<error_code>error code</error_code>
<error_str>error text</error_str>
</error>
</mdx>
```

Medex API Tech Support contact details

Email: api@medextld.com
 ICQ: 61625446
 Skype: prombez
 Phone: +7 (952) 211-08-86