

## DELAVO API SPECIFICATION

Version: 1.3

This document describes the format of Delavo API calls and supported parameters. It includes the following sections:

1. API "Authorize User"
2. API "List Users"
3. API "Get User"
4. API "Get User Membership"
5. API "Set User Membership"
6. API "Add User"
7. API "Update User"
8. API "Process Sale"
9. API "Import Sale"

1. API "Authorize User": allows remote authentication of a user through passing user's e-mail and password to Delavo.

*Request URL:*

```
http://<delavo_domain>/<delavo_directory>/action/Jin/APIUser/authorize.txt?  
gate_id=<gate_id>&secret=<secret>&email=<user_email>&password=<user_password>
```

The request is a HTTP/HTTPS request made either through GET or POST method.

*Request parameters:*

**<delavo\_domain>** - domain where Delavo is installed

**<delavo\_directory>** - directory in which Delavo is installed

**<gate\_id>** - ID of an API gate which should be set up by Delavo admin at Tools / API Users

**<secret>** - secret word defined for the API gate defined at Tools / API Users

**<user\_email>** and **<user\_password>** - e-mail and password of a user to authenticate, as if entered in a log in form

*Request example:*

```
http://delavosecure.com/delavo/action/Jin/APIUser/authorize.txt?  
gate_id=3&secret=gateaccess2635&email=john.white@yahoo.com&password=john28365
```

*Response format:*

When a request is passed to API "Authorize User", Delavo checks the passed e-mail and password against Delavo users database and returns two types of response depending on whether the user is found or not and whether there was an error.

The response is a text message containing two or more lines. The first line is always "SUCCESS" or "ERROR", and the other lines contain some additional information.

If the user is found and the password matches, the response will be:

```
SUCCESS  
id=2234  
referer_id=53  
points=17100  
email=john.white@yahoo.com  
fname=John  
lname=White  
name=John White  
address=John's address  
city=John's city  
state=US_CA  
zipcode=12345  
country=US
```

If an error occurs, the response will look like:

```
ERROR  
Invalid gate
```

or

```
ERROR  
Authorization failed 1
```

*User selections:*

It is important to understand how API gates provide access to certain user selections from users database in Delavo. Admin can define a certain user selection in each API gate. For example, it can be "only users that have membership Gold" or "only users who bought Product A", or even a few user record manually selected from the database.

When a user is being authenticated through an API gate, then for successful authentication it's necessary that user with passed e-mail exist in the DB, his password matches the passed password and the user belongs to the user selection defined in API gate. This means that even if the credential passed are correct for an existing user, but the user does not belong to the user selection, then authentication will fail. This enables the following application of API "Authorize User":

- - user selection can include all users in Delavo DB - in this case the gate works as a remote authentication tool
- - user selection can include specific group of users (like members of a certain membership) - in this case the gate works as a way to check if someone belongs to a certain group (has a certain membership)

2. API "List Users": provides a list of all users included in the user selection defined by API gate

*Request URL:*

```
http://<delavo_domain>/<delavo_directory>/action/Jin/APIUser/listusers.xml?  
gate_id=<gate_id>&secret=<secret>
```

The request is a HTTP/HTTPS request made either through GET or POST method.

*Request parameters:*

**<delavo\_domain>** - domain where Delavo is installed  
**<delavo\_directory>** - directory in which Delavo is installed  
**<gate\_id>** - ID of an API gate which should be set up by Delavo admin at Tools / API Users  
**<secret>** - secret word defined for the API gate defined at Tools / API Users

*Request example:*

```
http://delavosecure.com/delavo/action/Jin/APIUser/listusers.xml?  
gate_id=3&secret=gateaccess2635&email=john.white@yahoo.com&password=john28365
```

*Response format:*

The response in case of successful operation is an XML in the following format:

```
<?xml version="1.0" encoding="utf-8"?>  
<DelavoAPIListUsers>  
  <users>  
    <user id="1">  
      <email>matt@jvmanager.com</email>  
      <name>Matthew Finn</name>  
    </user>  
    <user id="2234">  
      <email>john.white@yahoo.com</email>  
      <name>John White</name>  
    </user>  
    <user id="483">  
      <email>steve.dow@gmail.com</email>  
      <name>Steve Dow</name>  
    </user>  
  </users>  
</DelavoAPIListUsers>
```

The <users> section includes as many <user> sections as there are users in the user selection defined in API gate. So if API gate provides access to all Delavo users and there are a lot, be ready to receive megabytes in the response.

The response in case of an error in an XML in the following format:

```
<?xml version="1.0" encoding="utf-8"?>  
<DelavoAPIListUsers>  
  <errorcode>INCORRECT_SECRET</errorcode>  
  <error>Incorrect secret code</error>  
</DelavoAPIListUsers>
```

### 3. API "Get User": provides personal data of a Delavo user

#### *Request URL:*

```
http://<delavo_domain>/<delavo_directory>/action/Jin/APIUser/getuser.xml?gate_id=<gate_id>&secret=<secret>&email=<user_email>&disable=<exclude_sections>
```

#### *OR*

```
http://<delavo_domain>/<delavo_directory>/action/Jin/APIUser/getuser.xml?gate_id=<gate_id>&secret=<secret>&id=<user_id>&disable=<exclude_sections>
```

The request is a HTTP/HTTPS request made either through GET or POST method.

#### *Request parameters:*

**<delavo\_domain>** - domain where Delavo is installed

**<delavo\_directory>** - directory in which Delavo is installed

**<gate\_id>** - ID of an API gate which should be set up by Delavo admin at Tools / API Users

**<secret>** - secret word defined for the API gate defined at Tools / API Users

**<user\_email>** or **<user\_id>** - two alternative ways to identify a user whose data are requested

**<disable\_sections>** - any of the following words separated by comma: "sales", "memberships", "roles", "affprogs". Each word, if present, tells Delavo to exclude the corresponding section from the response. If this parameter is empty or missing, Delavo will produce the full report.

#### *Request example:*

```
http://delavosecure.com/delavo/action/Jin/APIUser/getuser.xml?gate_id=3&secret=gateaccess2635&id=2346&disable=sales,memberships
```

#### *Response format:*

The response in case of successful operation is an XML in the following format:

```
<?xml version="1.0" encoding="utf-8"?>
<DelavoAPIGetUser>
  <user>
    <id>1</id>
    <signature>44E241BF140EA4C0</signature>
    <registered>1</registered>
    <referer_id>3</referer_id>
    <regtime>2010-01-20 19:20:27</regtime>
    <sestime>2011-01-19 12:42:34</sestime>
    <reg_ip>::1</reg_ip>
    <points>308</points>
    <email>matt@jvmanager.com</email>
    <passwd>test</passwd>
    <fname>Matthew</fname>
    <lname>Finn</lname>
    <name>Matthew Finn</name>
    <address>matt&apos;s address</address>
    <city>matt&apos;s city</city>
    <state>US_AZ</state>
    <zipcode>12345</zipcode>
    <country>US</country>
    <shipping_address>sh ord addr</shipping_address>
    <shipping_city></shipping_city>
    <shipping_state></shipping_state>
    <shipping_zipcode>sh ord zip</shipping_zipcode>
    <shipping_country>US</shipping_country>
  </url></url>
```

```

<company></company>
<phone></phone>
<checks></checks>
<tax_id></tax_id>
<pg_paypal_email></pg_paypal_email>
<cb_aff_id></cb_aff_id>
<no_admin_mail>0</no_admin_mail>
<no_upline_mail>0</no_upline_mail>
<admin_notes></admin_notes>
</user>
<sales>
  <sale id="164">
    <tstamp>2010-08-21 23:57:58</tstamp>
    <ctime></ctime>
    <signature>4042219u</signature>
    <specprice_descr></specprice_descr>
    <amount>120</amount>
    <amount_prod>120</amount_prod>
    <amount_discount>0</amount_discount>
    <amount_tax_pc></amount_tax_pc>
    <amount_tax_embed>0</amount_tax_embed>
    <amount_tax>0</amount_tax>
    <amount_shipping></amount_shipping>
    <amount_comms>120</amount_comms>
    <prod_id>97</prod_id>
    <prod_title>flat</prod_title>
    <numprod>1</numprod>
    <paid>0</paid>
    <txn_pgate_type></txn_pgate_type>
    <txn_pgate_id>0</txn_pgate_id>
    <txn_code></txn_code>
    <txn_waiting>0</txn_waiting>
    <order_id>137</order_id>
    <user_id>1</user_id>
    <aff_user_id>0</aff_user_id>
    <aff_campaign_id></aff_campaign_id>
    <metaaff_id>0</metaaff_id>
    <container_sale_id>0</container_sale_id>
    <old_sale_id>0</old_sale_id>
    <refund_deadline>2010-08-21</refund_deadline>
    <refunded>0</refunded>
    <remote_ip>::1</remote_ip>
    <subscr_id></subscr_id>
    <subscr_not_initial>0</subscr_not_initial>
    <subscr_inst_number>0</subscr_inst_number>
    <subscr_descr_more></subscr_descr_more>
    <subscr_refsale>0</subscr_refsale>
    <subscr_amount></subscr_amount>
    <subscr_amount_prod></subscr_amount_prod>
    <no_renew>0</no_renew>
    <no_cust_salelist>0</no_cust_salelist>
    <oto_shown>0</oto_shown>
    <aweber_done>0</aweber_done>
    <is_quickresell>0</is_quickresell>
    <download_ctime></download_ctime>
    <coupon_code>0</coupon_code>
    <coupon_discount></coupon_discount>
    <points_cashed></points_cashed>
    <points_earned></points_earned>
    <shipping_id>0</shipping_id>
    <shipping_subid></shipping_subid>
    <shipping_recur>0</shipping_recur>
    <admin_notes></admin_notes>
  </sale>

```

```

</sales>
<memberships>
  <membership id="3">
    <ctime>2010-11-29 17:40:12</ctime>
    <obj_id>4</obj_id>
    <obj_title>http://localhost/minima/test/lmem2/</obj_title>
    <sale_id>249</sale_id>
    <expires>2012-01-24</expires>
    <expired>0</expired>
  </membership>
</memberships>
<roles>
  <role id="1">
    <role_id>1</role_id>
    <role_title>Supervisor</role_title>
    <sale_id>0</sale_id>
    <expires></expires>
    <expired>0</expired>
    <notapproved>0</notapproved>
  </role>
</roles>
<affprogs>
  <affprog id="13">
    <ctime>2011-01-19 18:30:02</ctime>
    <affprog_id>1</affprog_id>
    <affprog_title>ALL</affprog_title>
    <sale_id>0</sale_id>
    <expires></expires>
    <expired>0</expired>
    <notapproved>0</notapproved>
  </affprog>
</affprogs>
</DelavoAPIGetUser>

```

Each of <sales>, <memberships>, <roles>, <affprogs> sections includes all sales, memberships, privilege packages and affiliate programs, associated with the requested user's record. Any of these sections can be omitted by setting the "disable" parameter in the request.

The response in case of an error in an XML in the following format:

```

<?xml version="1.0" encoding="utf-8"?>
<DelavoAPIGetUser>
  <errorcode>INCORRECT_SECRET</errorcode>
  <error>Incorrect secret code</error>
</DelavoAPIGetUser>

```

### **Secure mode:**

This API can be turned in to the "secure" mode by checking the "Enable secure option" checkbox in the API gate settings at Tools / API Users. If checked, then this API will require that user's password be passed in the "password" parameter. The password will be checked against the user's record, and if does not match, an error will be produced.

Also, if the user record does not belong to the user selection, defined in API gate, then an error will be produced.

#### 4. API "Get User Membership": provides a Delavo user membership's data

##### *Request URL:*

```
http://<delavo_domain>/<delavo_directory>/action/Jin/APIUser/getusermembership.xml?gate_id=<gate_id>&secret=<secret>&user_id=<user_id>&membership_id=<membership_id>
```

The request is a HTTP/HTTPS request made either through GET or POST method.

##### *Request parameters:*

**<delavo\_domain>** - domain where Delavo is installed  
**<delavo\_directory>** - directory in which Delavo is installed  
**<gate\_id>** - ID of an API gate which should be set up by Delavo admin at Tools / API Users  
**<secret>** - secret word defined for the API gate defined at Tools / API Users  
**<user\_id>** - user's ID  
**<membership\_id>** - membership's ID (taken from Delavo, Store / Memberships)

##### *Request example:*

```
http://delavosecure.com/delavo/action/Jin/APIUser/getusermembership.xml?gate_id=2&secret=secret2134&user_id=1&membership_id=4
```

##### *Response format:*

The response in case of successful operation is an XML in the following format:

```
<?xml version="1.0" encoding="utf-8"?>
<DelavoAPIGetUserMembership>
  <membership id="3">
    <ctime>2010-11-29 17:40:12</ctime>
    <obj_id>4</obj_id>
    <obj_title>http://localhost/minima/test/lmem2/</obj_title>
    <sale_id>249</sale_id>
    <expires>2012-01-24</expires>
    <expired>0</expired>
  </membership>
  <sale id="249">
    <tstamp>2010-11-29 17:39:10</tstamp>
    <ctime>2010-11-29 17:40:12</ctime>
    <signature>0sls0199</signature>
    <specprice_descr></specprice_descr>
    <amount>0</amount>
    <amount_prod>0</amount_prod>
    <amount_discount>0</amount_discount>
    <amount_tax_pc>0</amount_tax_pc>
    <amount_tax_embed>0</amount_tax_embed>
    <amount_tax>0</amount_tax>
    <amount_shipping></amount_shipping>
    <amount_comms>0</amount_comms>
    <prod_id>119</prod_id>
    <prod_title>Test upgrade product (upgrade)</prod_title>
    <numprod>1</numprod>
    <paid>1</paid>
    <txn_pgate_type>manual</txn_pgate_type>
    <txn_pgate_id>0</txn_pgate_id>
    <txn_code>MANUAL-211-249</txn_code>
    <txn_waiting>0</txn_waiting>
    <order_id>211</order_id>
    <user_id>1</user_id>
```

```

<aff_user_id>2</aff_user_id>
<aff_camp_id>0</aff_camp_id>
<metaaff_id>0</metaaff_id>
<container_sale_id>0</container_sale_id>
<old_sale_id>0</old_sale_id>
<refund_deadline>2010-11-29</refund_deadline>
<refunded>0</refunded>
<remote_ip>::1</remote_ip>
<subscr_id>jvmsub12</subscr_id>
<subscr_not_initial>0</subscr_not_initial>
<subscr_inst_number>0</subscr_inst_number>
<subscr_descr_more>subscription - free trial</subscr_descr_more>
<subscr_refsale>0</subscr_refsale>
<subscr_amount>200</subscr_amount>
<subscr_amount_prod>200</subscr_amount_prod>
<no_renew>0</no_renew>
<no_cust_salelist>0</no_cust_salelist>
<oto_shown>0</oto_shown>
<aweber_done>0</aweber_done>
<is_quickresell>0</is_quickresell>
<download_ctime>2010-11-29 17:40:12</download_ctime>
<coupon_code>0</coupon_code>
<coupon_discount></coupon_discount>
<points_cashed></points_cashed>
<points_earned></points_earned>
<shipping_id>0</shipping_id>
<shipping_subid></shipping_subid>
<shipping_recur>0</shipping_recur>
<admin_notes></admin_notes>
</sale>
<order id="211">
  <ctime>2010-11-29 17:39:10</ctime>
  <signature>wz812w79</signature>
  <sequence>Sale:249</sequence>
  <complete>1</complete>
  <pack_id>1</pack_id>
  <pack_title>test</pack_title>
  <aff_user_id>2</aff_user_id>
  <aff_camp_id>0</aff_camp_id>
  <user_id>1</user_id>
  <user_email>matt@jvmanager.com</user_email>
  <user_fname>M</user_fname>
  <user_lname>F</user_lname>
  <user_name>M F</user_name>
  <user_address>matt's address</user_address>
  <user_city>matt's city</user_city>
  <user_state>AU_NSW</user_state>
  <user_zipcode>12345</user_zipcode>
  <user_country>AU</user_country>
  <shipping_address>sh ord addr</shipping_address>
  <shipping_city></shipping_city>
  <shipping_state></shipping_state>
  <shipping_zipcode>sh ord zip</shipping_zipcode>
  <shipping_country>AU</shipping_country>
  <user_phone></user_phone>
  <user_company></user_company>
</order>
</DelavoAPIGetUserMembership>

```

Sections <sale> and <order> are present only if the membership was obtained through a purchase in Delavo.

The response in case of an error in an XML in the following format:

```
<?xml version="1.0" encoding="utf-8"?>
<DelavoAPIGetUserMembership>
  <errorcode>INCORRECT_SECRET</errorcode>
  <error>Incorrect secret code</error>
</DelavoAPIGetUserMembership>
```

The successful response is generated if there is a membership record with the passed membership ID, associated with the passed user ID. Note that this API call, although utilizing API gate mechanism, does not require that the requested user belong to the user selection, defined in the API gate. In this API call API gate is used only for the security reason. So, this PAI call works with the entire Delavo memberships database.

5. API "Set User Membership": allows adding, deleting and changing user membership's data

*Request URL:*

```
http://<delavo_domain>/<delavo_directory>/action/Jin/APIUser/setusermembership.xml?gate_id=<gate_id>&secret=<secret>&user_id=<user_id>&membership_id=<membership_id>&expires=<new expiration date>
```

The request is a HTTP/HTTPS request made either through GET or POST method.

*Request parameters:*

- <delavo\_domain>** - domain where Delavo is installed
- <delavo\_directory>** - directory in which Delavo is installed
- <gate\_id>** - ID of an API gate which should be set up by Delavo admin at Tools / API Users
- <secret>** - secret word defined for the API gate defined at Tools / API Users
- <user\_id>** - user's ID
- <membership\_id>** - membership's ID (taken from Delavo, Store / Memberships)
- <expires>** - expiration date to set in the SQL date format. Please note that the parameter is considered mandatory, and if it's empty or missing, the membership expiration will be set to lifetime. So, empty value in this field means a lifetime membership.

*Request example:*

```
http://delavosecure.com/delavo/action/Jin/APIUser/setusermembership.xml?gate_id=2&secret=secret2134&user_id=1&membership_id=4&expires=2015-01-24
```

*Response format:*

The response in case of successful operation is an XML in the following format:

```
<?xml version="1.0" encoding="utf-8"?>
<DelavoAPIActionResult>
  <complete/>
  <id>3</id>
</DelavoAPIActionResult>
```

Element <id> is the ID of the user-to-membership link object in Delavo.

The response in case of an error in an XML in the following format:

```
<?xml version="1.0" encoding="utf-8"?>
<DelavoAPIActionResult>
  <errorcode>INCORRECT_SECRET</errorcode>
  <error>Incorrect secret code</error>
</DelavoAPIActionResult>
```

The manipulation of the user's membership record is done through the "expires" parameter. If the date in the "expires" parameter is set to the past, the membership will be instantly deactivated and the membership record will be marked as expired.

Adding a new user's membership record will happen if there is no membership record in Delavo DB with the passed membership ID for the user with the passed user ID.

This API call, although utilizing API gate mechanism, does not require that the referred user belong to the user selection, defined in the API gate. In this API call API gate is used only for the security reason. So, this API call works with the entire Delavo memberships database.

## 6. API "Add User": allows adding a new user record in Delavo

### Request URL:

```
http://<delavo_domain>/<delavo_directory>/action/Jin/APIUser/adduser.xml?gate_id=<gate_id>&secret=<secret>&<USER_PARAMS>&assign_membership_<mem_id>=<expires>&assign_role_<role_id>=<expires>&assign_affprog_<affprog_id>=<expires>
```

The request is a HTTP/HTTPS request made either through GET or POST method. It is recommended to send the request via POST method. All fields should be URL-encoded.

### Request parameters:

**<delavo\_domain>** - domain where Delavo is installed

**<delavo\_directory>** - directory in which Delavo is installed

**<gate\_id>** - ID of an API gate which should be set up by Delavo admin at Tools / API Users

**<secret>** - secret word defined for the API gate defined at Tools / API Users

**assign\_membership\_<mem\_id>=<expires>** - instruction for Delavo to assign a membership with membership ID <mem\_id> and expiration date <expires> to the user record after it's created. The <mem\_id> is a part of the parameter name. There can be more than one parameter of this type in the request, each one with different <mem\_id>. The <expires> date is an SQL date. For a lifetime membership, it should be empty.

**assign\_role\_<role\_id>=<expires>, assign\_affprog\_<affprog\_id>=<expires>** - two types of parameters, similar to "assign\_membership\_\*". They allow to assign a privilege package and an affiliate program to the user record after it's created.

**<USER\_PARAMS>** - a group of parameters describing user's record. The supported parameters list follows below. One parameter is mandatory - "email". Please note that API does not check if the passed e-mail is correct.

**referer\_id** - referring user's ID

**points** - number of points on user's account (integer)

**email** - user's email

**passwd** - user's password in plain (unencrypted) form

**fname** - user's first name

**lname** - user's last name

**address** - user's address

**city** - user's city

**state** - user's state code in the following format: "<country\_code>\_<state\_code>". So, for America, California the state code should be "US\_CA"

**zipcode** - user's zipcode

**country** - user's two-symbol country code (e.g. "US", "CA", "AU")

**shipping\_address** - user's shipping address

**shipping\_city** - user's shipping city

**shipping\_state** - user's shipping state code

**shipping\_zipcode** - user's shipping zipcode

**shipping\_country** - user's shipping country code

**url** - user's site URL

**company** - user's company name

**phone** - user's phone

**checks** - user's "Write checks to"

**tax\_id** - user's tax ID

**pg\_paypal\_email** - user's Paypal merchant e-mail

**cb\_aff\_id** - user's Clickbank Affiliate ID

**admin\_notes** - admin notes

**reg\_ip** - registration IP address (optional)

**tracklink\_id** - tracklink link id to associate user's record with (Sales / Tracking link)

*Request example:*

```
http://delavosecure.com/delavo/action/Jin/APIUser/adduser.xml?gate_id=2&secret=secret2134&email=sarah.jones@gmail.com&passwd=ASDFGH&fname=Sarah&lname=Jones&points=15000&assign_membership_2=2020-12-20&assign_membership_4=&assign_role_5=2050-11-12&assign_affprog_1=
```

*Response format:*

The response in case of successful operation is an XML in the following format:

```
<?xml version="1.0" encoding="utf-8"?>
<DelavoAPIActionResult>
  <complete/>
  <id>3</id>
</DelavoAPIActionResult>
```

Element <id> is the ID of the newly created user record.

The response in case of an error in an XML in the following format:

```
<?xml version="1.0" encoding="utf-8"?>
<DelavoAPIActionResult>
  <errorcode>INCORRECT_SECRET</errorcode>
  <error>Incorrect secret code</error>
</DelavoAPIActionResult>
```

This API call, although utilizing API gate mechanism, does not require that the referred user belong to the user selection, defined in the API gate. In this API call API gate is used only for the security reason. So, this API call works with the entire Delavo users database.

7. API "Update User": allows updating a user record in Delavo, as well as adding/deleting and changing user's memberships, privilege packages and affiliate programs

*Request URL:*

```
http://<delavo_domain>/<delavo_directory>/action/Jin/APIUser/updateuser.xml?gate_id=<gate_id>&secret=<secret>&id=<user_id><USER_PARAMS>&assign_membership_<mem_id>=<expires>&assign_role_<role_id>=<expires>&assign_affprog_<affprog_id>=<expires>
```

The request is a HTTP/HTTPS request made either through GET or POST method. It is recommended to send the request via POST method. All fields should be URL-encoded.

*Request parameters:*

**<delavo\_domain>** - domain where Delavo is installed

**<delavo\_directory>** - directory in which Delavo is installed

**<gate\_id>** - ID of an API gate which should be set up by Delavo admin at Tools / API Users

**<secret>** - secret word defined for the API gate defined at Tools / API Users

**<user\_id>** - ID of the user's record to update. A record with this ID must exist in Delavo database

**assign\_membership\_<mem\_id>=<expires>** - instruction for Delavo to assign a membership with membership ID <mem\_id> and expiration date <expires> to the user record after it's created. The <mem\_id> is a part of the parameter name. There can be more than one parameter of this type in the request, each one with different <mem\_id>. The <expires> date is an SQL date. For a lifetime membership, it should be empty. With this parameter, a new membership can be added to the user's record, existing membership's expiration date can be updated, or a membership can be deactivated - by passing a past expiration date.

**assign\_role\_<role\_id>=<expires>, assign\_affprog\_<affprog\_id>=<expires>** - two types of parameters, similar to "assign\_membership\_\*". They allow to assign a privilege package and an affiliate program to the user record after it's created. They also allow to change or delete existing privilege packages and/or affiliate programs for a user's record.

**<USER\_PARAMS>** - a group of parameters describing user's record. The supported parameters list is the same as for API "Add User".

*Request example:*

```
http://delavosecure.com/delavo/action/Jin/APIUser/updateuser.xml?gate_id=2&secret=secret2134&id=245&passwd=cb48yvb9&address=New+address&points=15120&assign_membership_2=2020-12-20&assign_membership_4=&assign_role_5=2050-11-12&assign_affprog_1=
```

*Response format:*

The response in case of successful operation is an XML in the following format:

```
<?xml version="1.0" encoding="utf-8"?>
<DelavoAPIActionResult>
  <complete/>
  <id>3</id>
</DelavoAPIActionResult>
```

Element <id> is the ID of the user's record.

The response in case of an error in an XML in the following format:

```
<?xml version="1.0" encoding="utf-8"?>
<DelavoAPIActionResult>
```

```
<errorcode>INCORRECT_SECRET</errorcode>  
<error>Incorrect secret code</error>  
</DelavoAPIActionResult>
```

This API call, although utilizing API gate mechanism, does not require that the referred user belong to the user selection, defined in the API gate. In this API call API gate is used only for the security reason. So, this API call works with the entire Delavo users database.

8. API "Process Sale": allows a remote creation and processing of a sale in Delavo

*Request URL:*

`http://<delavo_domain>/<delavo_directory>/action/Jin/Order/Web/api_process_sale.xml?<parameters>`

The request is a HTTP/HTTPS request made either through GET or POST method. The supported parameters are:

**pack\_id** - a Delavo Package or Funnel object ID (mandatory)  
**prod\_id** - a Delavo Product ID (mandatory)  
**pgate\_id** - a Delavo Payment Option object ID (optional). If it's missing, API will attempt to use the "Live Payment option" from the Funnel record or the first payment option from the "Bulk payment" filed of the Package  
**order\_id** - a Delavo Order object ID. If specified, the newly created sale will be attached to an existing Order object, and the Order data will be updated, otherwise a new Order object will be created  
**aff\_id** - affiliate ID  
**tracklink\_id** - trackink link id to associate user's and sale's records with (Sales / Tracking link)  
**camp\_id** - campaign ID  
**remote\_ip** - IP address from where the sale was made  
**user\_email** - customer's email (mandatory)  
**user\_fname** - customer's first name  
**user\_lname** - customer's last name  
**user\_address** - customer's address  
**user\_city** - customer's city  
**user\_state** - customer's state code in the following format: "<country\_code>\_<state\_code>". So, for America, California the state code should be "US\_CA"  
**user\_zipcode** - customer's zipcode  
**user\_country** - customer's two-symbol country code (e.g. "US", "CA", "AU")  
**user\_phone** - customer's phone  
**user\_company** - customer's company name  
**shipping\_address** - customer's shipping address  
**shipping\_city** - customer's shipping city  
**shipping\_state** - customer's shipping state code  
**shipping\_zipcode** - customer's shipping zipcode  
**shipping\_country** - customer's shipping country code  
**cardtype** - credit card type  
**cardnumber** - credit card (mandatory)  
**cardexpmonth** - 1- or 2-digit credit card expiration month number (mandatory)  
**cardexpyear** - 4-digit credit card expiration year (mandatory)  
**cardcvc** - credit card CVC2/CVV2 code

*Response format:*

The response in case of successful operation is an XML in the following format:

```
<?xml version="1.0" encoding="utf-8"?>
<DelavoAPIActionResult>
  <complete/>
  <id>3</id>
</DelavoAPIActionResult>
```

Element <id> is the ID of the newly created Sale record.

The response in case of an error in an XML in the following format:

```
<?xml version="1.0" encoding="utf-8"?>
<DelavoAPIActionResult>
```

```
<errorcode>INVALID_PROD_ID</errorcode>  
<error>Invalid or no Product ID</error>  
</DelavoAPIActionResult>
```

9. API “Process Sale”: allows importing a sale record in Delavo

*Request URL:*

```
http://<delavo_domain>/<delavo_directory>/action/Jin/Order/Web/api_import_sale.xml?<parameters>
```

The request is a HTTP/HTTPS request made either through GET or POST method. The supported parameters are:

**pack\_id** - a Delavo Package or Funnel object ID (mandatory)  
**prod\_id** - a Delavo Product ID (mandatory)  
**pgate\_id** - a Delavo Payment Option object ID (mandatory). The payment will not be processed, this parameters is needed for the integrity of the records  
**order\_id** - a Delavo Order object ID. If specified, the newly created sale will be attached to an existing Order object, and the Order data will be updated, otherwise a new Order object will be created.  
**aff\_id** - affiliate ID  
**tracklink\_id** – trackink link id to associate user's and sale's records with (Sales / Tracking link)  
**camp\_id** - campaign ID  
**user\_email** - customer's email (mandatory)  
**user\_fname** - customer's first name  
**user\_lname** - customer's last name  
**user\_address** - customer's address  
**user\_city** - customer's city  
**user\_state** - customer's state code in the following format: "<country\_code>\_<state\_code>". So, for America, California the state code should be "US\_CA"  
**user\_zipcode** - customer's zipcode  
**user\_country** - customer's two-symbol country code (e.g. "US", "CA", "AU")  
**user\_phone** - customer's phone  
**user\_company** - customer's company name  
**shipping\_address** - customer's shipping address  
**shipping\_city** - customer's shipping city  
**shipping\_state** - customer's shipping state code  
**shipping\_zipcode** - customer's shipping zipcode  
**shipping\_country** - customer's shipping country code  
**tstamp** – optional timestamp of the record creation in SQL date or SQL date-and-time format. If omitted, the current timestamp will be taken  
**descr** – optional sale description. If omitted, the product title will be taken as the description  
**remote\_ip** – IP address from where the sale was made (optional)  
**admin\_notes** – optional admin notes for the sale  
**amount** – sale amount  
**transaction\_id** – transaction ID (mandatory)  
**is\_subscr** – 0 if it's a flat sale and 1 if it's a part of subscription (either initial sale or a recurring installment)  
**subscr\_id** – subscription\_id (mandatory for subscriptions)  
**is\_recurring** – 1 if it's a recurring installment of a subscription  
**init\_sale\_id** – Delavo Sale ID of the initial sale of the subscription. Mandatory for all recurring installments. If order\_id parameter is passed, it will be ignored for the recurring installments  
**subscr\_inst\_number** – optional installment number. 0 for trial period, 1 for the next installment (or for the first installment if there is no trial), 2..etc for all other installments  
**create\_deliverables** – 1 to create deliverables when the sale record is imported  
**create\_commissions** – 1 to create commissions when the sale record is imported

*Response format:*

The response in case of successful operation is an XML in the following format:

```
<?xml version="1.0" encoding="utf-8"?>  
<DelavoAPIActionResult>  
  <complete/>
```

```
    <id>3</id>
</DelavoAPIActionResult>
```

Element `<id>` is the ID of the newly created Sale record.

The response in case of an error in an XML in the following format:

```
<?xml version="1.0" encoding="utf-8"?>
<DelavoAPIActionResult>
  <errorcode>INVALID_PROD_ID</errorcode>
  <error>Invalid or no Product ID</error>
</DelavoAPIActionResult>
```