

Documentation OpenEMM Webservices 1.2.1

The OpenEMM SOAP webservices interface of **OpenEMM 6.0** offers 15 methods to remote control OpenEMM (by a CMS or CRM application for example).

Deployment information:

To create a webservices user, add a self defined user name and password to database table ws_admin_tbl with a SQL statement like this:

```
INSERT INTO "ws_admin_tbl" values (1, "wsusername", "wspassword");
```

EmmSoapClient.sh (Linux) EmmSoapClient.cmd (Windows):

This script is a test-client for the OpenEMM webservices. Please adjust the path information in the second line of the file to point to your deploy directory. When using the default paths of OpenEMM this is

```
../webapps/openemm (Linux)
```

Executing the script EmmSoapClient.[sh|cmd] launches class org.agnitas.webservice.EmmSoapClient and gives you an overview of all available services and their parameters. The client tries to connect to http://localhost:8080 and therefore has to be started on the server side.

EmmCreateWSDL.sh (Linux) / EmmCreateWSDL.cmd (Windows):

This script creates a valid WSDL file named ws.wsdl based on the OpenEMM interface class EMMWebService_Port. Please adjust the path information in the second line of the file to point to your deploy directory. When using the default paths of OpenEMM this is

```
../webapps/openemm
```

The Linux script can be started like this (no line breaks):

```
./EmmCreateWSDL.sh -o ws.wsdl -l "http://yourdomain:8080/emm_webservice"  
-n "urn:agnitas-webservice" -w ALL  
org.agnitas.webservice.EmmWebService_Port
```

The Windows script can be started like this (no line breaks):

```
EmmCreateWSDL.cmd -o ws.wsdl -l "http://yourdomain:8080/emm_webservice"  
-n "urn:agnitas-webservice" -w ALL  
org.agnitas.webservice.EmmWebService_Port
```

List of all available webservice methods:

<i>Webservice method</i>	<i>Description</i>
newEmailMailing	Creates a new email-mailing.
newEmailMailingWithReply	Creates a new email-mailing with reply-to address.
updateEmailMailing	Updates an email-mailing.
insertContent	Adds content to an existing mailing.
deleteContent	Deletes one mailing content from the database.
sendMailing	Sends admin-, test-, and world-mailings.
addMailinglist	Creates a new mailing list
deleteMailinglist	Deletes an existing mailing list
addSubscriber	Adds a subscriber to database.
updateSubscriber	Updates a subscriber in the database.
findSubscriber	Searches for a subscriber in the database.
getSubscriber	Gets subscriber-data from the database.
deleteSubscriber	Deletes one subscriber from the database.
setSubscriberBinding	Changes binding of subscribers in mailinglists.
getSubscriberBinding	Gets subscriber binding from the database.

newEmailMailing

This Method creates a new email-mailing. It creates the mailing only, use **insertContent** to fill in the text and **sendMailing** to send out the email.

<i>Parameter</i>	<i>Description</i>
username	Username for webservice.
password	Password for webservice.
shortname	Name of the mailing. Only visible in OpenEMM, not visible for the recipients of the mailing.
description	Description (max. 1000 chars) of the mailing. Only visible in OpenEMM, not visible for the recipients of the mailing.
mailinglistID	ID of the mailinglist, must exist in OpenEMM.
targetID	IDs of target groups (semicolon separated list). Possible values: 0: no target group, all recipients from the chosen mailinglist >0: an existing target group ID
mailingType	Type of mailing. Possible values: 0: normal mailing 1: event-based mailing 2: rule-based mailing (like automated birthday mailings)
templateID	ID of a template used for creating the mailing. Can be every existing template- or mailingID in the OpenEMM or 0 for no template. If no template is used, two content-blocks are available for insertContent : <i>emailText</i> and <i>emailHtml</i> .
emailSubject	Subjectline of the email.
emailSender	Senderaddress of the email. Can be a simple email-address like <i>info@agnitas.de</i> or combined with a realname like: <i>"Agnitas AG"</i> < <i>info@agnitas.de</i> >.
emailCharset	Characterstet to be used for encoding the email, usually <i>iso-8859-1</i> .
emailLinefeed	Automated linefeed in the text-version of the email.

<i>Parameter</i>	<i>Description</i>
emailFormat	Format of the email. Possible values: 0: only text 1: online-HTML (Multipart) 2: offline-HTML (Multipart + embedded graphics)

Note:

If you are using a template, the base settings of this template and content blocks will be used for the new mailing.

For the base settings the values for **mailingType**, **emailCharset**, **emailLinefeed** and **emailFormat** will be taken from the template and can not be changed by the method call.

The values for **mailinglistID**, **targetID** and **emailSender** will be taken from the template in case you set them to 0 and "" (empty string for emailSender) respectively. All other valid values will overwrite the values of the template.

The **emailSubject** will be overwritten everytime.

method-definition

```

public int newEmailMailing(java.lang.String username,
                          java.lang.String password,
                          java.lang.String shortname,
                          java.lang.String description,
                          int mailinglistID,
                          StringArrayType targetID,
                          int mailingType,
                          int templateID,
                          java.lang.String emailSubject,
                          java.lang.String emailSender,
                          java.lang.String emailCharset,
                          int emailLinefeed,
                          int emailFormat)

```

return ID of created mailing or 0 (if failed)

newEmailMailingWithReply

This Method creates a new email-mailing with a reply address. It creates the mailing only, use **insertContent** to fill in the text and **sendMailing** to send out the email.

<i>Parameter</i>	<i>Description</i>
username	Username for webservice.
password	Password for webservice.
shortname	Name of the mailing. Only visible in OpenEMM, not visible for the recipients of the mailing.
description	Description (max. 1000 chars) of the mailing. Only visible in OpenEMM, not visible for the recipients of the mailing.
mailinglistID	ID of the mailinglist, must exist in OpenEMM.
targetID	IDs of target groups (semicolon separated list). Possible values: 0: all recipients from a given mailinglist >0: an existing target group ID
mailingType	Type of mailing. Possible values: 0: normal mailing 1: event-based mailing 2: rule-based mailing (like automated birthday mailings)
templateID	ID of a template used for creating the mailing. Can be every existing template- or mailingID in the OpenEMM or 0 for no template. If no template is used, two content-blocks are available for insertContent : <i>emailText</i> and <i>emailHtml</i> .
emailSubject	Subjectline of the email.
emailSender	Senderaddress of the email. Can be a simple email-address like <i>info@agnitas.de</i> or a realname like: "Agnitas AG" < <i>info@agnitas.de</i> >.
emailReply	Reply-to address of the email. Can be a simple email-address like <i>info@agnitas.de</i> or combined with a realname like: "Agnitas AG" < <i>info@agnitas.de</i> >.
emailCharset	Characterstet to be used for encoding the email, usually <i>iso-8859-1</i> .
emailLinefeed	Automated linefeed in the text-version of the email.

<i>Parameter</i>	<i>Description</i>
emailFormat	Format of the email. Possible values: 0: only text 1: online-HTML (Multipart) 2: offline-HTML (Multipart + embedded graphics)

Note:

If you are using a template, the base settings of this template and content blocks will be used for the new mailing.

For the base settings the values for **mailingType**, **emailCharset**, **emailLinefeed** and **emailFormat** will be taken from the template and can not be changed by the method call.

The values for **mailinglistID**, **targetID** and **emailSender** will be taken from the template in case you set them to 0 and "" (empty string for emailSender) respectively. All other valid values will overwrite the values of the template.

The **emailSubject** will be overwritten everytime.

method-definition

```

public int newEmailMailingWithReply(java.lang.String username,
                                     java.lang.String password,
                                     java.lang.String shortname,
                                     java.lang.String description,
                                     int mailinglistID,
                                     StringArrayType targetID,
                                     int mailingType,
                                     int templateID,
                                     java.lang.String emailSubject,
                                     java.lang.String emailSender,
                                     java.lang.String emailReply,
                                     java.lang.String emailCharset,
                                     int emailLinefeed,
                                     int emailFormat)
  
```

return ID of created mailing or 0 (if failed)

updateEmailMailing

Method for updating an email-mailing. It only updates the Mailing, use **insertContent** to fill in the text and **sendMailing** to send out the E-Mail.

<i>Parameter</i>	<i>Description</i>
username	Username for webservice.
password	Password for webservice.
mailingID	ID of the mailing which should be changed.
shortname	Name of the mailing. Only visible in OpenEMM, not visible for the recipients of the mailing.
description	Description (max. 1000 chars) for the mailing. Only visible in OpenEMM, not visible for the recipients of the mailing.
mailinglistID	ID of the mailinglist, must exist in OpenEMM.
targetID	IDs of target groups (semicolon separated list). Possible values: 0: all recipients from given mailinglist >0: an existing target group id
mailingType	Type of mailing. Possible values: 0: normal mailing 1: event-based mailing 2: rule-based mailing (like automated birthday mailings)
emailSubject	Subjectline for the email.
emailSender	Senderaddress for the email. Can be a simple email-address like <i>info@agnitas.de</i> or combined with a realname like: <i>"Agnitas AG"</i> < <i>info@agnitas.de</i> >.
emailReply	Reply-to address of the email. Can be a simple email-address like <i>info@agnitas.de</i> or combined with a realname like: <i>"Agnitas AG"</i> < <i>info@agnitas.de</i> >.
emailCharset	Characterstet to be used for encoding the email, usually <i>iso-8859-1</i> .
emailLinefeed	Automated linefeed in the text-version of the email.

<i>Parameter</i>	<i>Description</i>
emailFormat	Format of the email. Possible values: 0: only text 1: online-HTML (Multipart) 2: offline-HTML (Multipart + embedded graphics)

Note:

If you are using a template, the base settings of this template and content blocks will be used for the new mailing.

For the base settings the values for **mailingType**, **emailCharset**, **emailLinefeed** and **emailFormat** will be taken from the template and can not be changed by the method call.

The values for **mailinglistID**, **targetID** and **emailSender** will be taken from the template in case you set them to 0 and "" (empty string for emailSender) respectively. All other valid values will overwrite the values of the template.

The **emailSubject** will be overwritten everytime.

method-definition

```

public boolean updateEmailMailing(java.lang.String username,
                                   java.lang.String password,
                                   int mailingID,
                                   java.lang.String shortname,
                                   java.lang.String description,
                                   int mailinglistID,
                                   StringArrayType targetID,
                                   int mailingType,
                                   java.lang.String emailSubject,
                                   java.lang.String emailSender,
                                   java.lang.String emailReply,
                                   java.lang.String emailCharset,
                                   int emailLinefeed,
                                   int emailFormat)

```

return true (successful update) or false (failed update)

insertContent

Method for adding content to an existing mailing.

<i>Parameter</i>	<i>Description</i>
username	Username for webservice.
password	Password for webservice.
mailingID	ID of the mailing to which a content-blocks should be added.
blockName	Name of content-block. Depends on the template used in newEmailMailing or newEmailMailingWithReply . If no template is used, the available block-names are <i>emailText</i> and <i>emailHtml</i> .
blockContent	Content of the blockname. Multiple contents, divided by target groups, can be provided for one blockname. The recipient only sees the content defined for him.
targetID	ID of the target group of this block-content. Use 0 for all subscribers.
priority	Priority of this block-content. Smaller numbers denote a higher priority.

method-definition

```
public int insertContent(java.lang.String username,  
                        java.lang.String password,  
                        int mailingID,  
                        java.lang.String blockName,  
                        java.lang.String blockContent,  
                        int targetID,  
                        int priority)
```

return length of provided block-content or 0

deleteContent

Method for deleting content of an existing mailing.

<i>Parameter</i>	<i>Description</i>
username	Username for webservice.
password	Password for webservice.
contentID	ID of the mailingcontent to which should be deleted.

method-definition

```
public int insertContent(java.lang.String username,  
                          java.lang.String password,  
                          int mailingID,  
                          java.lang.String blockName,  
                          java.lang.String blockContent,  
                          int targetID,  
                          int priority)  
  
return 1 (success) or 0 (failure)
```

sendMailing

Method for sending admin-, test- and world-mailings.

<i>Parameter</i>	<i>Description</i>
username	Username for webservice.
password	Password for webservice.
mailingID	ID of the mailing which should be sent.
sendGroup	User group to which the email has to be sent to. Possible values: 'A': only admin-subscribers (for testing) 'T': only test- and admin-subscribers (for testing) 'W': all subscribers (can only be executed once)
sendTime	Scheduled send-time in seconds since January 1, 1970, 00:00:00 GMT
stepping	For artificially slowing down the send-process. Seconds between delivery of two mailing-blocks. Set to 0 unless you know exactly what you are doing.
blocksize	For artificially slowing down the send-process. Number of emails in one mailing-block. Set to 0 unless you know exactly what you are doing.

method-definition

```
public int sendMailing(java.lang.String username,  
                      java.lang.String password,  
                      int mailingID,  
                      java.lang.String sendGroup,  
                      int sendTime,  
                      int stepping,  
                      int blocksize)  
  
return 1 (success) or 0 (failure)
```

addMailinglist

Method for creating a new mailing list.

<i>Parameter</i>	<i>Description</i>
username	Username for webservice.
password	Password for webservice.
shortname	Name of the mailing list. Only visible in OpenEMM.
description	Description (max. 1000 chars) of the mailing list. Only visible in OpenEMM.

method-definition

```
public int addMailinglist(java.lang.String username,  
                           java.lang.String password,  
                           java.lang.String shortname,  
                           java.lang.String description)
```

```
return ID of created mailing list or 0 (if failed)
```

deleteMailinglist

Method to delete an existing mailing list.

<i>Parameter</i>	<i>Description</i>
username	Username for webservice.
password	Password for webservice.
mailinglistID	ID of the mailing list to be deleted.

method-definition

```
public int addMailinglist(java.lang.String username,  
                           java.lang.String password,  
                           int mailinglistID)
```

```
return 1 (success) or 0 (failure)
```

addSubscriber

Method for adding a subscriber to database.

<i>Parameter</i>	<i>Description</i>
username	Username for webservice.
password	Password for webservice.
doubleCheck	If <i>true</i> , checks if the recipient is already in database and avoids that a recipient is in the database more than once. If <i>false</i> , recipient will be added in any case (independently from state of overwrite parameter).
keyColumn	Specifies the column used for doubleCheck.
overwrite	If <i>true</i> (and DoubleCheck <i>true</i>), recipients that are already in the database get updated and missing parameters will be set to NULL. If <i>false</i> , subscriber will be added depending on state of doubleCheck..
paramNames	Name of columns that will be filled with data (semicolon separated list, see note below).
paramValues	Values for paramNames (semicolon separated list).

To transfer parameter fields with dates , you have to split the content into several sub fields like this:

Necessary sub fields:

- fieldname_DAY_DATE
- fieldname_MONTH_DATE
- fieldname_YEAR_DATE

Optional sub fields:

- fieldname_HOUR_DATE
- fieldname_MINUTE_DATE
- fieldname_SECOND_DATE

method-definition

```
public int addSubscriber(java.lang.String username,  
                        java.lang.String password,  
                        boolean doubleCheck,  
                        java.lang.String keyColumn,  
                        boolean overwrite,  
                        StringArrayType paramNames,  
                        StringArrayType paramValues)
```

```
return customerID of added or updated recipient
```

Please note, that for paramNames the field names email, gender and mailtype are mandatory. You should set gender in paramValues to 2 (unknown) and mailtype to 0 (text only) or 1

(HTML).

updateSubscriber

Method for updating a subscriber in the database.

<i>Parameter</i>	<i>Description</i>
username	Username for webservice.
password	Password for webservice.
customerID	ID of the subscriber to be updated.
paramNames	Name of columns that will be filled with data (semicolon separated list), non-listed columns will be left unchanged.
paramValues	Values for paramNames (semicolon separated list).

To transfer parameter fields with dates , you have to split the content into several sub fields like this:

Necessary sub fields:

- fieldname_DAY_DATE
- fieldname_MONTH_DATE
- fieldname_YEAR_DATE

Optional sub fields:

- fieldname_HOUR_DATE
- fieldname_MINUTE_DATE
- fieldname_SECOND_DATE

method-definition

```
public int addSubscriber(java.lang.String username,  
                          java.lang.String password,  
                          int customerID,  
                          StringArrayType paramNames,  
                          StringArrayType paramValues)  
  
return true (successful update) or false (failed update)
```

findSubscriber

Method for searching for a subscriber in the database.

<i>Parameter</i>	<i>Description</i>
username	Username for webservice.
password	Password for webservice.
keyColumn	Column by which recipient should be retrieved
value	Value to be searched for.

method-definition

```
public int findSubscriber(java.lang.String username,  
                           java.lang.String password,  
                           java.lang.String keyColumn,  
                           java.lang.String value)
```

```
return 0 or customerID of the first match.
```

getSubscriber

Method for getting the data of a subscriber from the database.

<i>Parameter</i>	<i>Description</i>
username	Username for webservice.
password	Password for webservice.
customerID	ID of the subscriber.

method-definition

```
public SubscriberData getSubscriber(java.lang.String username,  
                                     java.lang.String password,  
                                     int customerID)
```

```
return class SubscriberData;  
        contains String Arrays (semicolon separated list): paramNames,  
paramValues
```

deleteSubscriber

Method for deleting the data of a subscriber from the database.

<i>Parameter</i>	<i>Description</i>
username	Username for webservice.
password	Password for webservice.
customerID	ID of the subscriber.

method-definition

```
public SubscriberData getSubscriber(java.lang.String username,  
                                     java.lang.String password,  
                                     int customerID)
```

```
return 1 (success) or 0 (failure)
```



```
int exitMailingID)
```

```
return 0 or customerID of the recipient
```

getSubscriberBinding

Method for getting the binding of a subscriber to a certain mailinglist.

<i>Parameter</i>	<i>Description</i>
username	Username for webservice.
password	Password for webservice.
customerID	ID of the customer, whose binding should be retrieved.
mailinglistID	ID of mailinglist for which the binding should be retrieved.
mediatype	Mediatype for which the binding should be retrieved. Possible values: 0: Email 1: SMS

method-definition

```
public String setSubscriberBinding(java.lang.String username,  
                                   java.lang.String password,  
                                   int customerID,  
                                   int mailinglistID,  
                                   int mediatype)
```

```
return status;bindingType;exitMailingID;remark
```

See webservice "setSubscriberBinding" for explanation of return parameters.