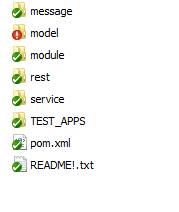
# User component

## Overview of the project structure

The User application was generated following the component archetype: *unionvms-component-archetype*. The module allows integration with different components based on JMS queues. The ***UVMSUserEvent*** queue is used to send requests towards this module and ***UVMSUser*** queue is used for receiving answers.



The ***message*** component includes to the communication of the module with external callers, receiving messages and generating the events. The ***model*** contains the generated data types based on UserCotextElement.xsd and converters from/to these types towards the entities used in actual calls for Information-Module. The ***service*** encapsulates the functionality of calling the USM component (especially Information module) and to retrieve the requested information. ***Rest*** component allows the calls directly to service module and not via JMS queues.

## API definition

The methods exposed by this component are following:

|  |  |
| --- | --- |
| **Method name** | **Description** |
| getUserContext | In case of a given ***application name***, for a specific B, this method returns all contexts along with the role, scope and all associated preferences for the corresponding context. |
| putUserPreferences | For the given user and application name, the listed contexts will have preferences list updated. |
| deployApplication | The given application is deployed under USM database, buy populating tables related to application configuration: features, datasets and options. |
| undeployApplication | This operation removes any configuration details of a specific ***application***, along with his features, datasets, and options. |
| redeployApplication | This operation updates an already installed application on ***USM*** database, by refreshing the information about features, datasets and options. |
| getContactDetails | This operation returns the details related to the person attached to a specific user name account. |
| getOrganisation | This method retrieves the organization details, along with child organisations and their endpoints, channels and contact details. |
| getDeploymentDescriptor | This method retrieves all details of an already configured application in the USM database, along with datasets, features and option lists. |
| ping | This operation is called from other modules that want to verify that ***User*** module is deployed and running. |

The methods are exposed through JMS queues. You can test by using HermesJMS tool.

JMS Messages (requests) are submitted to the queue "UVMSUserEvent".

You can find JMS Messages (responses) for the above requests in the queue "UVMSUser" or whatever other queue you like but in both cases you have to set it in the JMSHeader "JMS ReplyTo" field ("UVMSUser or other).

JMS messages payload are compliant with the requests and responses defined in UserModuleService.wsdl.

You can see some examples:

**getUserContext**

JMS Request Payload:

<mod:getUserContextRequest xmlns:mod="module.user.wsdl.fisheries.ec.europa.eu">

<method>GET\_USER\_CONTEXT</method>

<contextId>

<userName>usm\_admin</userName>

<applicationName>USM</applicationName>

</contextId>

</mod:getUserContextRequest>

JMS Response Payload:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<ns2:getUserContextResponse xmlns:ns2="module.user.wsdl.fisheries.ec.europa.eu">

<context>

<userName>usm\_admin</userName>

<applicationName>USM</applicationName>

<contextSet>

<contexts>

<role>

<roleName>USM-UserBrowser</roleName>

<feature>

<name>viewApplications</name>

</feature>

<feature>

<name>viewUsers</name>

</feature>

<feature>

<name>viewOrganisations</name>

</feature>

<feature>

<name>viewScopes</name>

</feature>

<feature>

<name>viewRoles</name>

</feature>

</role>

</contexts>

<contexts>

<role>

<roleName>USM-UserManager</roleName>

<feature>

<name>manageOrganisations</name>

</feature>

<feature>

<name>copyUserProfile</name>

</feature>

<feature>

<name>manageScopes</name>

</feature>

<feature>

<name>manageUsers</name>

</feature>

<feature>

<name>manageApplications</name>

</feature>

<feature>

<name>manageRoles</name>

</feature>

<feature>

<name>configurePolicies</name>

</feature>

</role>

</contexts>

</contextSet>

</context>

</ns2:getUserContextResponse>

**putUserPreferences**

JMSRequestPayload:

<mod:putUserPreferencesRequest xmlns:mod="module.user.wsdl.fisheries.ec.europa.eu">

<method>PUT\_USER\_PREFERENCES</method>

<context>

<userName>usm\_admin</userName>

<applicationName>USM</applicationName>

<contextSet>

<context>

<role>

<roleName>USM-UserManager</roleName>

</role>

<preferences>

<preference>

<applicationName>USM</applicationName>

<optionName>levelCode</optionName>

<optionValue>option-value1</optionValue>

</preference>

</preferences>

</context>

<context>

<role>

<roleName>USM-UserBrowser</roleName>

</role>

</context>

</contextSet>

</context>

</mod:putUserPreferencesRequest>

JMS Response Payload:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<ns2:putUserPreferencesResponse xmlns:ns2="module.user.wsdl.fisheries.ec.europa.eu">

<response>OK</response>

</ns2:putUserPreferencesResponse>

**deployApplication**

JMS Request Payload:

<mod:deployApplicationRequest xmlns:mod="module.user.wsdl.fisheries.ec.europa.eu">

<method>DEPLOY\_APPLICATION</method>

<application>

<name>hello-app</name>

<!--Optional:-->

<description>Hello app desc</description>

<!--Optional:-->

<parent>USM</parent>

<!--Zero or more repetitions:-->

<dataset>

<name>hello-dataset-1</name>

<!--Optional:-->

<description>Hello dataset 1 description</description>

<category>Dataset Category 1</category>

<!--Optional:-->

<discriminator>Hello dataset 1 discriminator</discriminator>

</dataset>

<!--Zero or more repetitions:-->

<option>

<name>hello option 1</name>

<!--Optional:-->

<description>hello option 1 description</description>

<!--Optional:-->

<dataType>String</dataType>

<!--Optional:-->

<defaultValue>default value</defaultValue>

<!--Optional:-->

<group>Hello option group</group>

</option>

<!--Zero or more repetitions:-->

<feature>

<name>hello feature 1</name>

<!--Optional:-->

<description>hello feature 1 description</description>

<!--Optional:-->

<group>hello feature group</group>

</feature>

</application>

</mod:deployApplicationRequest>

**undeployApplication**

JMS Request Payload:

<mod:undeployApplicationRequest xmlns:mod="module.user.wsdl.fisheries.ec.europa.eu">

<method>UNDEPLOY\_APPLICATION</method>

<applicationName>adoun-app</applicationName>

</mod:undeployApplicationRequest>

**redeployApplication**

JMS Request Payload:

<mod:redeployApplicationRequest xmlns:mod="module.user.wsdl.fisheries.ec.europa.eu">

<method>REDEPLOY\_APPLICATION</method>

<application>

<name>hello-app</name>

<!--Optional:-->

<description>Hello app desc</description>

<!--Optional:-->

<parent>USM</parent>

<!--Zero or more repetitions:-->

<dataset>

<name>hello-dataset-1</name>

<!--Optional:-->

<description>Hello dataset 1 description</description>

<category>Dataset Category 1</category>

<!--Optional:-->

<discriminator>Hello dataset 1 discriminator</discriminator>

</dataset>

<!--Zero or more repetitions:-->

<option>

<name>hello option 1</name>

<!--Optional:-->

<description>hello option 1 description</description>

<!--Optional:-->

<dataType>String</dataType>

<!--Optional:-->

<defaultValue>default value</defaultValue>

<!--Optional:-->

<group>Hello option group</group>

</option>

<!--Zero or more repetitions:-->

<feature>

<name>hello feature 1</name>

<!--Optional:-->

<description>hello feature 1 description</description>

<!--Optional:-->

<group>hello feature group</group>

</feature>

</application>

</mod:redeployApplicationRequest>

**getDeploymentDescriptor**

JMS Request Payload:

<mod:getDeploymentDescriptorRequest xmlns:mod="module.user.wsdl.fisheries.ec.europa.eu">

<method>GET\_DEPLOYMENT\_DESCRIPTOR</method>

<applicationName>USM</applicationName>

</mod:getDeploymentDescriptorRequest>

**ping**

JMS Request Payload:

<mod:pingRequest xmlns:mod="module.user.wsdl.fisheries.ec.europa.eu">

<method>PING</method>

</mod:pingRequest>

## Generate the entities, building project and deploy application

The entities located under ***model*** are not committed on svn. They must be always generated.

This generation can be done running either of the below commands:

* inside ***model*** directory: mvn clean install –Pgenerate-from-wsdl
* inside APP upper parent directory (above the list mentioned in the previous picture)

**mvn clean install -U -Pgenerate-from-wsdl,generate-rest-doc -Dhostname=XXX -Dport=yyy**

For installing this component automatically after it is build the following command must be executed by changing the hostname, port, username and password:

**mvn clean install wildfly:deploy --projects module -Pwildfly-deploy -Dhostname=XXX -Dport=9990 -Dusername=YYY -Dpassword=ZZZZ**

## Module dependencies

The User module performs its actions by calling Information-Service component of USM. All database calls and interactions with USM entities are encapsulated into this module. The calls are directly performed by injecting the EJB and not via a REST service.