EUROPEAN COMMISSION

DIGIT
Digital Europe Programme

Service Metadata Publisher

Administration Guide

DomiSMP 5.0

Version [3.7]

Status [Final]

Date: 22/06/2023

Document Approver(s):

Approver Name	Role
Bogdan DUMITRIU	Project Manager

Document Reviewers:

Reviewer Name	Role
Jože RIHTARŠIČ	Developer
Caroline AEBY	TESO Support

Summary of Changes:

Version	Date	Created by	Short Description of Changes
0.1	22/02/2016	DHENEIN Christophe	Initial Document Creation
1.0	15/06/2017	Chaouki BERRAH (CEFS), Christophe DHENEIN (CEFS)	Document restructuring and updating.
1.1	16/06/2016	Chaouki BERRAH (CEFS), Christophe DHENEIN (CEFS)	3.0.0-RC2 changed to 3.X Screens updated.
1.2	12/02/2018	Chaouki BERRAH	Version 4.0 RC1 changes added
1.3	16/02/2018	Chaouki BERRAH	Update after test installation
1.4	21/02/2018	Chaouki BERRAH	Pawel changes included.
1.5	26/02/2018	Chaouki BERRAH	Links added
1.6	28/02/2018	Chaouki BERRAH Caroline AEBY	Updates
1.7	20/03/2018	CEF Support	Reuse notice added
1.8	11/06/2018	Chaouki BERRAH	Document update
1.9	28/06/2018	CEF Support	References to nexus updated.
2.0	01/10/2018	Caroline AEBY	No more standby service
2.1	18/10/2018	Chaouki BERRAH	Updates
2.2	23/10/2018	Caroline AEBY	SMP 4.1 RC – SMP admin console
2.3	25/10/2018	Jože RIHTARŠIČ	SMP 4.1 Updates
2.4	15/11/2018	Jože RIHTARŠIČ Chaouki BERRAH	Oracle Script Changes for Migration purposes
2.5	28/11/2018	Jože RIHTARŠIČ Chaouki BERRAH	SMP 4.1 Updates
2.6	04/12/2018	Chaouki BERRAH	Weblogic and user creation Updates
2.7	19/09/2019	Chaouki BERRAH Gowtham VAITHIAM	Document Update
2.8	08/10/2019	Jože RIHTARŠIČ	SMP 4.1.1 Release updates
2.9	06/12/2019	Caroline AEBY	Typo
3.0	03/03/2020	Chaouki BERRAH Gowtham VAITHIAM	MySql Connector V8 and default keystore and trustore comments.
3.1	23/03/2021	Jože RIHTARŠIČ	Tested on Oracle 19c

3.2	11/04/2022	Caroline AEBY	No more CEF
3.3	19/05/2022	Caroline AEBY	CEF eDelivery FMB change to EC-EDELIVERY
3.4	01/06/2022	Caroline AEBY Jože RIHTARŠIČ	SMP 4.2 RC1
3.5	24/06/2022	Caroline AEBY Jože RIHTARŠIČ	SMP 4.2. FR
3.6	11/05/2023	Jože RIHTARŠIČ Caroline AEBY	DomiSMP 5.0 RC1
3.7	20/06/2023	Jože RIHTARŠIČ	Update properties

Table of Contents

1. INTRODUCTION

This Administration Guide is intended for Administrators who are in charge of installing, managing and troubleshooting an eDelivery SMP (Service Metadata Publisher).

1.1. Purpose

The purpose of this guide is to provide detailed information on how to deploy and configure an SMP 4.2.X on either a WebLogic 12.2 c or Tomcat 8.5.x Application Server with either MySQL or Oracle database.

It also provides detailed descriptions of the related Security Configurations (Certificates).

There is also a section on the use of Soap UI to create, update and delete SMP Service Groups and Metadata.

Another section describes an alternative method to perform the creation, update and deletions using Swagger UI.

2. CONVENTION

The Commands and Configuration files listed in this document usually contain a mix of reserved words (commands, instructions and system-related special words), user-defined words (chosen by the user) as well as comments and default/preferred values for some fields or variables.

The conventions used in this document, to distinguish between them, are the following:

- **Bold** is used for "reserved" words and commands.
- *Normal italic* together with a short description of the argument is used for user-defined names (chosen by yourself to designate items like users, passwords, database etc.). It normally contains at least 2 words separated by "_".
- ***Bold and italic*** is used for advisable values which can be changed by the user depending on their infrastructure.
- Comments are sometimes added to describe the purpose of the commands, usually enclosed in brackets ().
- By default, non-OS specific paths will be described using Linux patterns.

Example 1: Sample Oracle Statement:

create user *smp_user* **identified by** *smp_password*;

grant all privileges to *smp_user*;

(Where *smp_user* and *smp_password* are names chosen by the user)

Example 2: Sample Configuration File:

smp.jdbc.driver = com.mysql.jdbc.Driver

smp.jdbc.url = jdbc:mysql://localhost:3306/*smp_database*

smp.jdbc.user = smp_user

smp.jdbc.password = *smp_password*

target-database = MySQL

(Where: *smp_user, smp_database* and *smp_password* are names chosen by the user.

localhost:3306 represents hostname:port parameters of the MySQL database.)

3. PREREQUISITES

Please install the following software on the target system. For further information and installation details, please refer to the software owner's documentation.

- Java runtime environment is (JRE) 8 and 11 **only**: http://www.oracle.com/technetwork/java/javase/downloads/index.html
- **One** of the supported Database Management Systems:
  - MySQL 8.0.x (tested version, future versions might also work)
  - Oracle 11g XE and Oracle 19c (tested version, future versions might also work)
- **One** of the supported Application Servers:
  - WebLogic 12.2.1.4 (tested with JDK 8)
  - WebLogic 14.1c (tested with JDK 11)
  - Tomcat 9.x (tested with JDK 8)

3.1. Binaries repository

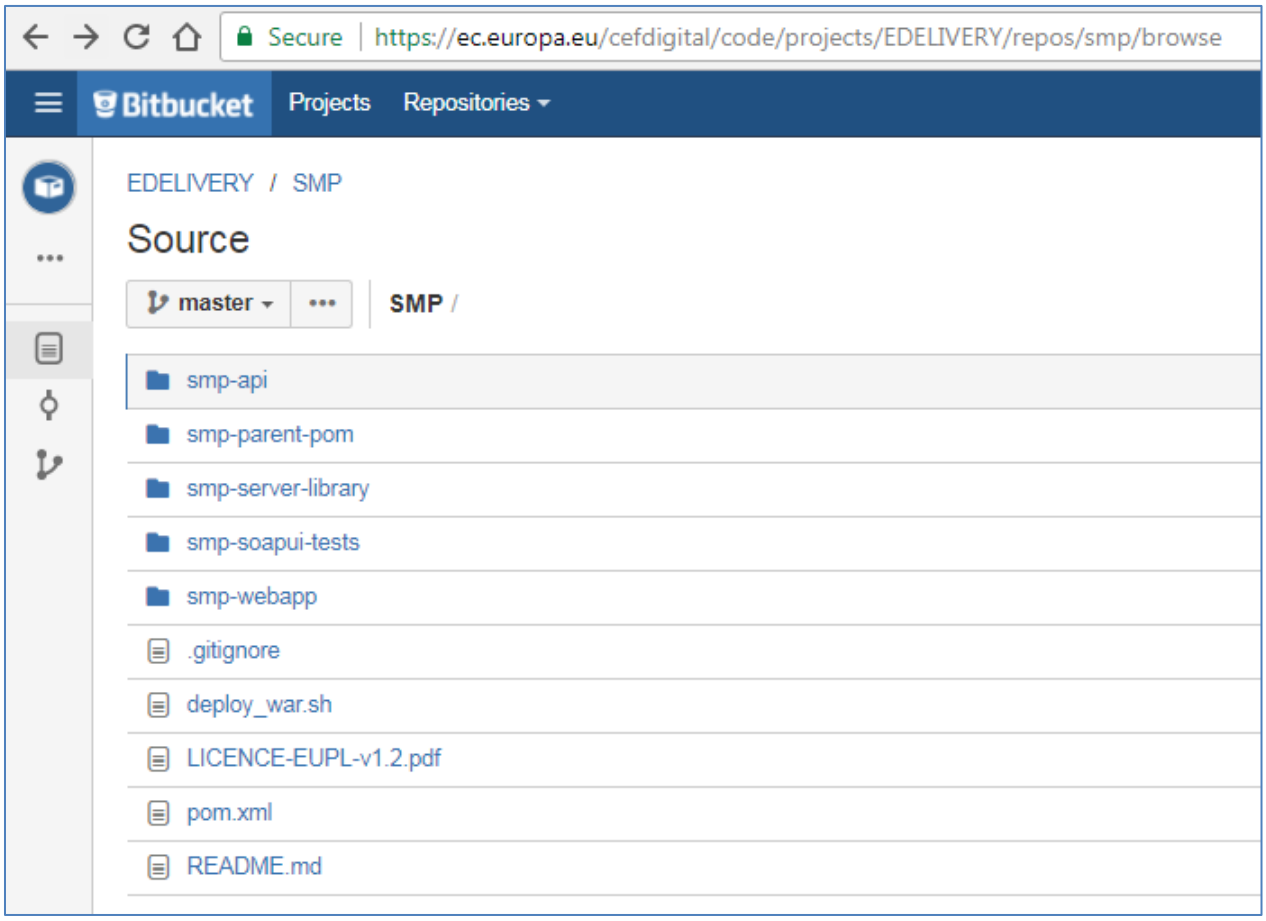
The DomiSMP artefacts can be downloaded from the Digital site[1].

3.2. Source Code Repository

The source code of eDelivery DomiSMP is available in the **GIT** repository at the following location:

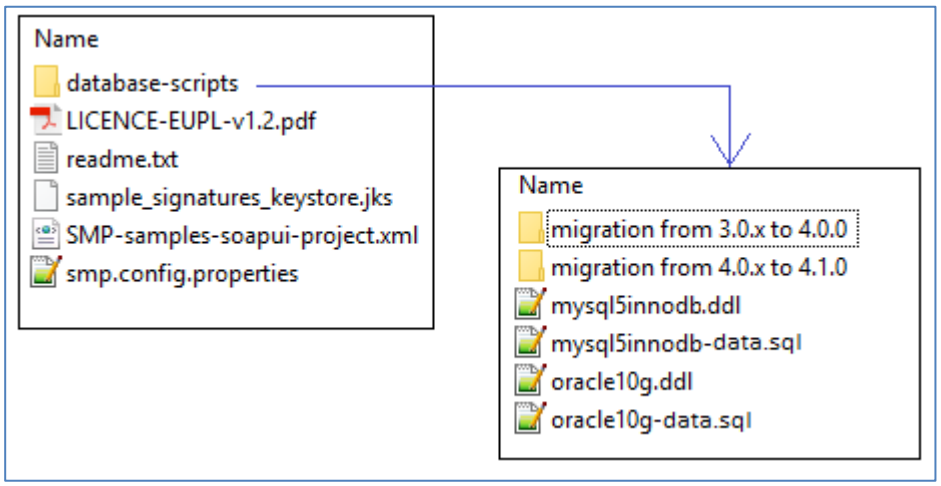
https://ec.europa.eu/digital-building-blocks/code/projects/EDELIVERY/repos/smp/browse

[1] https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/SMP+software



3.3. Database Scripts

The scripts to create (or migrate) the Oracle or MySQL databases are included in the following downloadable zip file from the Digital site (section §3.1): smp-x-setup.zip.



4. DEPLOYMENT

4.1. Deployment overview

As mentioned in the prerequisites, the deployment of the SMP is only supported on Tomcat or WebLogic application servers.

The deployment of the SMP on both platforms is almost identical but minor platform specific changes will be documented in a dedicated section of this manual.

The deployment of the SMP is summarized in the following mandatory steps:

- Database Configuration
- Application Server Preparation (Weblogic and Tomcat) for SMP
- SMP Initial Configuration
- SMP .WAR file Deployment

Remark:

*The environment variable, **AS_HOME**, refers to the application server home folder where the SMP package is installed.*

- *For Tomcat, it refers to **CATALINA_HOME**.*
- *For Oracle WebLogic, it refers to **DOMAIN_HOME**.*

*The environment variable, **SETUP_PATH**, refers to the folder where the deployment SMP package smp-4.x-setup.zip is extracted.*

4.1. Folder structure

The following subdirectories must be created in the AS_HOME directory. The document describes the default folder settings and can be named or created in a location other than the AS_HOME directory.

- AS_HOME/smp: the folder contains the basic SMP settings, and the folder must be configured as a classpath: see sections: for WebLogic §6.2 and for Tomcat §7.1.
- AS_HOME/logs: the purpose of the folder is to contain SMP logs.
- AS_HOME/security: the previous versions of the SMP have security artifacts (truststore, keystore, etc.) under the 'smp' folder. We recommend creating a separate folder for a more transparent handling of the security artifacts. In case of setting SMP in an application server cluster, this folder must be shared among the cluster nodes. The location of the folder must be set in the SMP application property*: smp.security.folder* (before DomiSMP 5.0 version, the application property name was *configuration.dir*).

5. Database Creation

This section describes the steps necessary to create the database, tables and the SMP database user (**dbuser** used for database connection purpose).

It also includes the creation of an initial SMP user account that will be used by REST clients to connect to the SMP.

The SMP uses a direct connection to the database, which removes the need to configure a data source within WebLogic.

For this step you need to use the script included in the zip file downloaded in section §3.3.

5.1. MySQL

1. Open a command prompt and navigate to the *SETUP_PATH/**sql-scripts*** folder
2. Execute the following MySQL commands:

```
mysql -h localhost -u root_user --password=root_password -e "drop schema if
exists smp_schema;create schema smp_schema;alter database smp_schema
charset=utf8; create user smp_dbuser@localhost identified by
'smp_password';grant all on smp_schema.* to smp_dbuser@localhost;"
```

This creates a *smp_schema* and an *smp_dbuser* with (all) privileges to the smp_schema**.**

Execute the following command to create the required objects (tables, etc.) in the database:

```
mysql -h localhost -u root_user --password=root_password smp_schema <
mysql5innodb.ddl
```

Execute the following command to fill initial test data:

```
mysql -h localhost -u root_user --password=root_password smp_schema <
mysql5innodb-data.sql
```

5.2. Oracle Database

1. Navigate to *SETUP_PATH***/sql-scripts** directory
2. Execute the following commands :

```
sqlplus sys as sysdba (password should be the one assigned during the Oracle
installation )
================================================================================
Once logged in Oracle:
create user smp_dbuser identified by smp_dbpassword;
grant all privileges to smp_dbuser;
connect smp_dbuser
```

```
show user;  (should return : smp_dbuser)
@oracle10g.ddl (run the scripts with the @ sign from the location of the scripts)

@oracle10g-data.ddl (Fill initial test data)

exit
================================================================================
```

6. ORACLE WEBLOGIC CONFIGURATION

This section does not include the installation of a WebLogic application server. It is assumed that the WebLogic Server is installed, and a WebLogic domain is created with an administration server and a managed server on which the SMP will be deployed.

Hereafter the domain location will be referred as *DOMAIN_HOME* (user-defined name).

In the examples below, we will use the following Domain and Server names:

- Domain Name : SMPDOMAIN
- Administration Server : AdminServer
- SMP Managed Server : SMP_ManagedServer

As shown below:

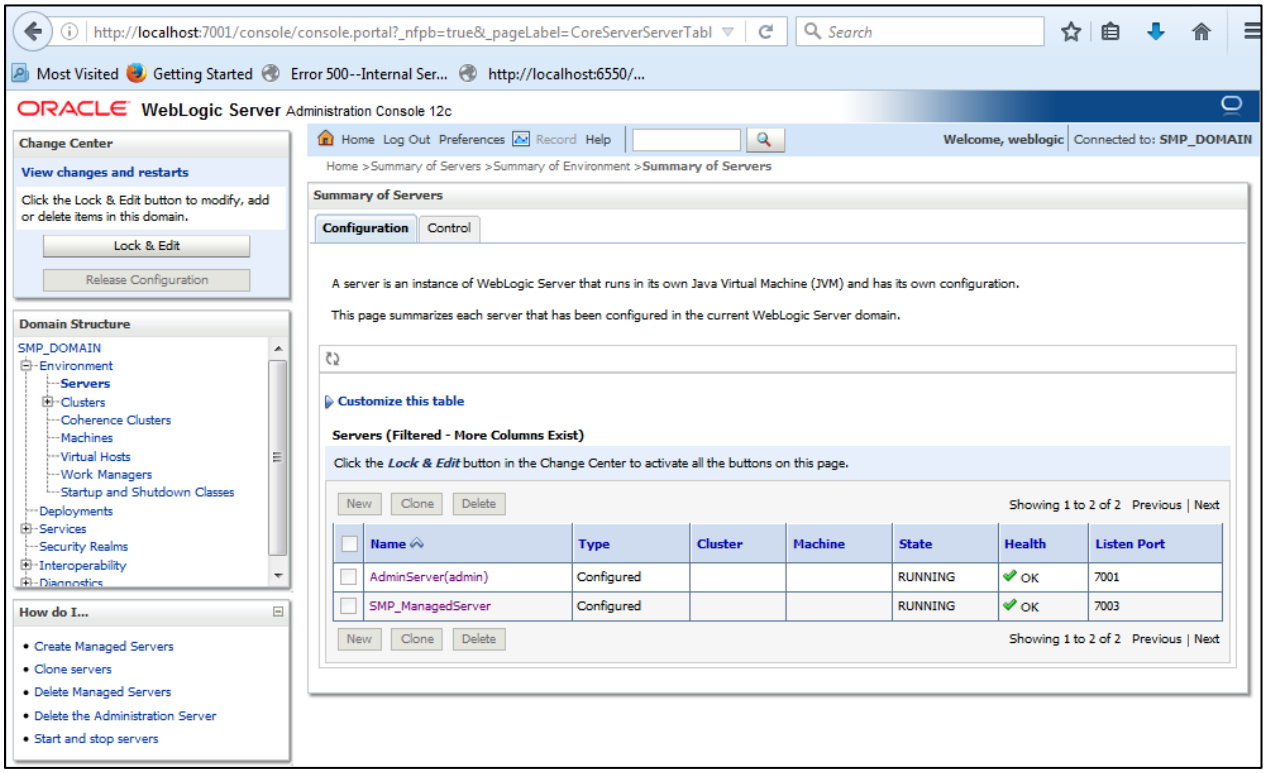


Figure 1 - Weblogic console

To deploy the SMP on the WebLogic Application Server platform, two preliminary steps need to be completed:

- Disabling the application basic Authentication on the Weblogic Server,
- Configuring the Extra CLASSPATH for WebLogic,
- Setup sun HTTP Handler.

This is described in the following 2 sections.

6.1. Disabling the Authentication on the WebLogic

By default, WebLogic performs its own basic authentication checks requests before passing the request to deployed application (e.g eDelivery SMP). The eDelivery SMP has its own authentication mechanism that makes the WebLogic authentication redundant, and it is therefore important to disable the WebLogic Authentication to stop it from interfering with the SMP authentication.

To do so, edit the config.xml file (under SMPDOMAIN/config) by adding the following tag before the `</security-configuration>` closing tag:

```
<enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>
```

Here is an example:

```
../
<enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-
credentials>
</security-configuration>
/..
```

6.2. Configuring the Extra CLASSPATH for WebLogic

The purpose of the section is to describe how to set up folder smp (see §4.1) as a classpath on the WebLogic server.

Edit the WebLogic DOMAIN_HOME/bin/setDomainEnv.sh.

For Linux:

Add the **EXPORT CLASSPATH=${CLASSPATH}:${DOMAIN_HOME}/smp** statement at the end of the CLASSPATH definition as shown below:

```
../
if [ "${PRE_CLASSPATH}" != "" ] ; then
        CLASSPATH="${PRE_CLASSPATH}${CLASSPATHSEP}${CLASSPATH}"
        export CLASSPATH
fi
        CLASSPATH=${CLASSPATH}:${DOMAIN_HOME}/smp
        export CLASSPATH
/..
```

For Windows:

```
../
If NOT "%PRE_CLASSPATH%"=="" (
      set CLASSPATH=%PRE_CLASSPATH%;%CLASSPATH%
)
set CLASSPATH=%CLASSPATH%;%DOMAIN_HOME%\smp
/..
```

6.3. Configuring Sun HTTP handler

Edit the WebLogic DOMAIN_HOME/bin/setDomainEnv.sh and add the following system parameter.

```
../
JAVA_OPTIONS=-DUseSunHttpHandler=true
export JAVA_OPTIONS
/..
```

7. TOMCAT CONFIGURATION

To deploy the SMP on Tomcat, the steps below need to be completed.

7.1. Configuring the Extra CLASSPATH for Tomcat

The purpose of the section is to describe how to set up folder smp (see §4.1) as a classpath on the Tomcat server.

For Linux:

Edit the CATALINA_HOME/bin/setenv.sh file

```
#!/bin/sh
# Set CLASSPATH to include $CATALINA_HOME/smp
# where the smp ‘smp.config.properties’ is located
export CLASSPATH=$CATALINA_HOME/smp
```

For Windows:

Edit the %CATALINA_HOME%/bin/setenv.bat file

```
REM Set CLASSPATH to include $CATALINA_HOME/smp
REM where the ‘smp.config.properties’ is located
set classpath=%classpath%;%catalina_home%\smp
```

7.2. JDBC Driver

The JDBC driver needs to be downloaded from the manufacturer website:

- For Oracle Database : https://www.oracle.com/database/technologies/appdev/jdbc-downloads.html
- For Mysql : https://www.mysql.com/products/connector/

The JDBC driver (.jar file) must be copied to the following directory: AS_HOME/lib.

8. SMP CONFIGURATION

The DomiSMP 5.0.x configuration has two types of properties:

- The system configuration properties: the properties are located in 'smp.config.properties' file and define environment settings such as JDBC connection, logging configuration, SMP extension library folder, etc. Before the first eDelivery SMP startup, the mandatory database connection properties must be set. The complete system property list is described in §15.2.1

- The SMP application properties: the property list with default values is stored at initial startup in the database table **SMP_CONFIGURATION**. System administrators can change most properties during the runtime without application restart. The complete application property list is described in section §15.2.2. In case we want to set different init value for particular property at first SMP startup, the property can be set in the 'smp.config.properties'.

For this step, use the **smp.config.properties** example delivered within the zip file downloaded in section §3.2. The **smp.config.properties** file must be copied to the CLASSPATH folder configured in $7.1 for Tomcat and §6.2 for WebLogic).

8.1. Database configuration

The eDelivery SMP database back-end configuration is performed within the eDelivery SMP configuration file (**smp.config.properties** file).

Depending on the selected database back-end, modify the **smp.config.properties** files as indicated below. Smp database connection can configured in property file or can use application server datasource configuration by JNDI.

8.1.1. *Oracle Database:*

- Datasource configured from property file:

```
../
## Sample for Oracle
jdbc.driver=oracle.jdbc.driver.OracleDriver
jdbc.url=jdbc:oracle:thin:@localhost:1521/xe
jdbc.user=smp
jdbc.password=secret123
hibernate.dialect=org.hibernate.dialect.Oracle10gDialect
/..
```

- Datasource (connection pool) configured on the application server using the JNDI (recommended):

```
../
hibernate.dialect=org.hibernate.dialect.Oracle10gDialect
```

```
# weblogic datasource JNDI example
# datasource.jndi=jdbc/edeliverySmpDS
# tomcat datasource JNDI example
datasource.jndi=java:comp/env/jdbc/edeliverySmpDS
```

8.1.2. *MySQL:*

- Datasource configured from property file

```
../
## Database access
# For mysql connector v8
#jdbc.driver = com.mysql.cj.jdbc.Driver
# For mysql connector v5
jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3306/smp
jdbc.user=smp
jdbc.password=secret123
hibernate.dialect =org.hibernate.dialect.MySQL5InnoDBDialect
/..
```

- Datasource (connection pool) configured on the application server using the JNDI (recommended)

```
../
hibernate.dialect=org.hibernate.dialect.Oracle10gDialect
# weblogic datasource JNDI example
# datasource.jndi=jdbc/edeliverySmpDS
# tomcat datasource JNDI example
datasource.jndi=java:comp/env/jdbc/edeliverySmpDS
/..
```

8.2. SMP Keystore

eDelivery SMP uses keystore for storing keys for the two different purposes:

- One **mandatory** key is used for signing the responses to **GET** requests (XMLDSIG response signing)
- One **optional** key is used to authenticate SMP using 2-way-SSL when it is calling SML via HTTPS.

If the Keystore does not exist when the SMP is started for the first time, it is automatically created with a sample key/certificate 'sample_key.'

The user with a system administrator role can update/manage the Keystore entries using the user interface on the System settings / Keystore page:

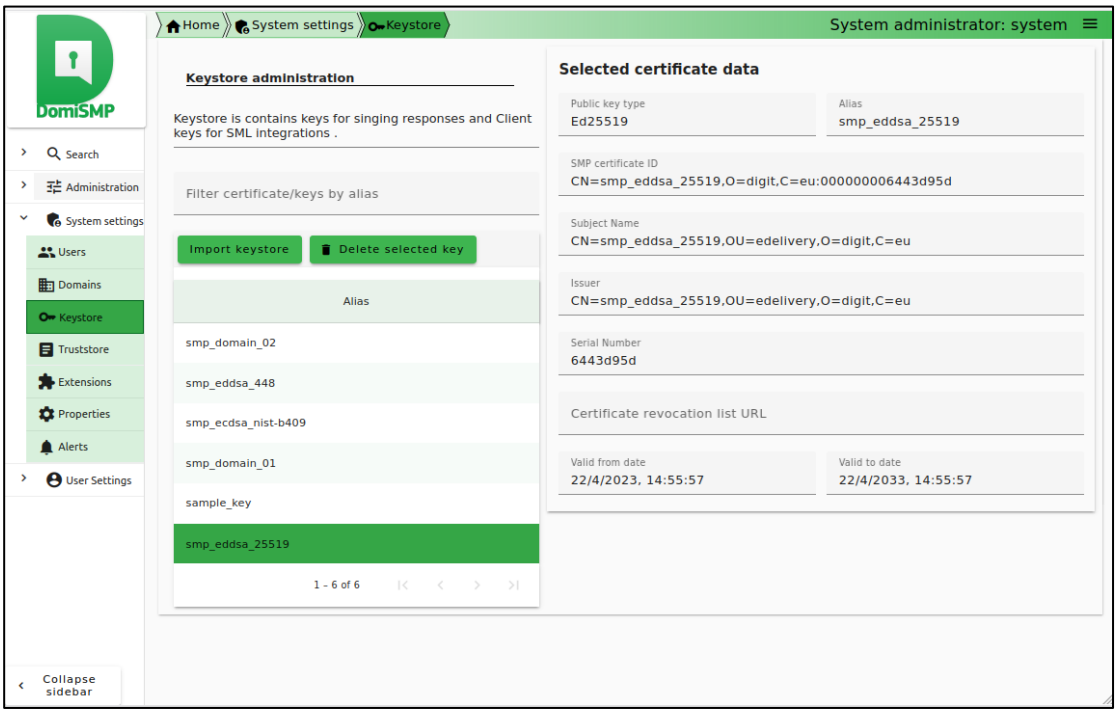
- 

Figure 2 - DomiSMP UI: Keystore administration page

8.3. SMP Truststore

eDelivery SMP uses truststore for storing trusted X509Certificates for the WebService 2-way-SSL authentication and for storing the SML server certificate. The truststore is automatically created at the initial SMP start-up. The truststore can be managed with a System admin account using the UI tools under the page System settings / Truststore.

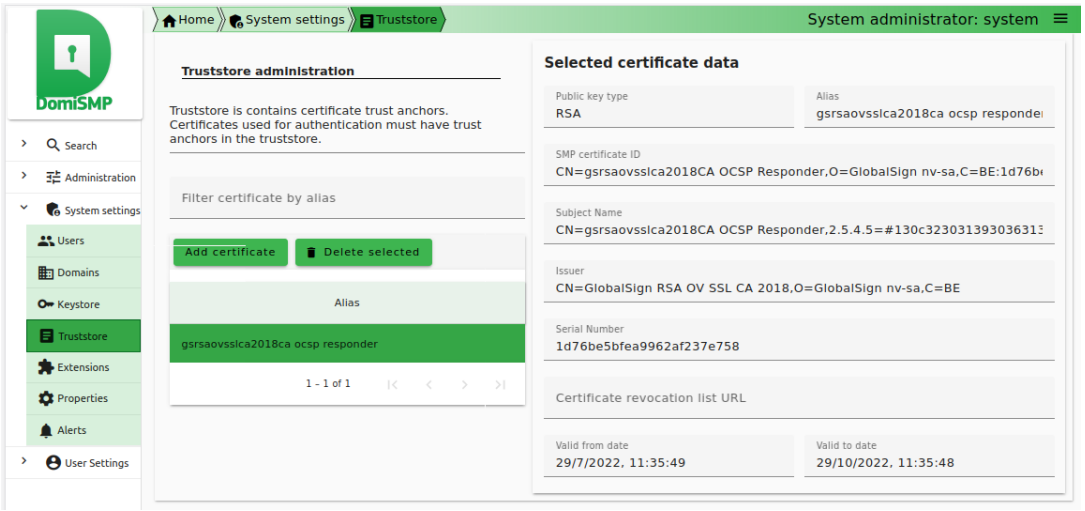


Figure 3 - DomiSMP UI: Truststore administration page

8.4. Custom Keystore and Truststore

On some systems, generating new passwords and keys can take a long time. To speed-up the initial startup, consider the following option:

- Install faster system random generators.
- In case of development or local testing, set property: **smp.mode.development=true** to **smp.config.properties.** To ensure high security, this option MUST NOT be enabled in production.
- Use custom/preprepared keystore and truststore as described below.

Users can configure eDelivery SMP to use prepared keystores at initial startup. To achieve this, the Keystore must be generated manually and saved in the SMP security folder. If the Keystore already contains the keys/certificates, they must have the same Key password as it was set for accessing the Keystore. The following properties must be set in **smp.config.properties:**

- smp.security.folder: the security folder of the SMP where the keystore must be located
- smp.keystore.filename: filename of the keystore
- smp.keystore.password: password for accessing the keystore and the keys. Please note that decrypted password must be wrapped in {DEC}{[PASSWORD]} as example: {DEC}{testPASSkeystore1234}
- smp.truststore.filename: filename of the keystore
- smp. truststore.password: password for accessing the keystore and the keys. Please note that decrypted password must be wrapped in {DEC}{[PASSWORD]} as example: {DEC}{testPASStruststore1234}

```
../
smp.security.folder=/opt/tomcat/security/
smp.keystore.filename=smp-keystore.p12
smp.keystore.password={DEC}{testPASSkeystore1234}
smp.keystore.type=PKCS12
smp.truststore.filename=smp-keystore.p12
smp.truststore.password={DEC}{testPASStruststore1234}
smp.truststore.type=PKCS12

/..
```

- After initial startup, the properties are stored (and the password encrypted) inside the SMP_CONFIGURATION table and they should be removed from the **smp.config.properties** file.

9. SMP .WAR FILE DEPLOYMENT

The eDelivery SMP is deployed using the steps described in the next sections.

9.1.1. *Tomcat*

Download and copy smp-X.war file in the Tomcat **webapps** directory (AS_HOME/webapps/smp.war).

Remark: The application context path is the same as the first part of the smp.war filename. For example, if we deploy the file "smp.war", then the application will be accessible on http://localhost:8080/smp/. If the deployed file is "smp-X.war", the application URL will be: http://localhost:8080/smp-X/.

9.1.2. *Oracle WebLogic*

Deploy the **.war** file within WebLogic using the Oracle Weblogic deployer feature or using the Weblogic Administration Console.

An example of using the Oracle the **weblogic.deployer,** is shown below:

```
java weblogic.Deployer -adminurl
t3://${WebLogicAdminServerListenAddress}:${WebLogicAdminServerPort} \
-username ${WebLogicAdminUserName} \
-password ${WebLogicAdminUserPassword} \
-deploy -name smp.war \
-targets ${SMP_ManagedServer} \
-source $TEMP_DIR/ smp.war
```

9.1.3. *Verification of the Installation*

Verify the installation by navigating with your browser to the following address:
http://*[hostname]*:*[port]*/smp

If the deployment is successful, the following page is displayed:

Figure 4 – DomiSMP: hello page

10. CONFIGURING THE EDELIVERY SMP FOR USE WITH AN BDMSL

The eDelivery SMP can establish BDMSL integration using two identification mechanisms:

- Using HTTP and plain text with metadata embedded into the HTTP header Client-Cert of the REST request Using HTTP and plain text with metadata embedded into the HTTP header Client-Cert of the REST request. This approach should **be used only for testing purposes** and only if both BDMSL and eDelivery SMP are located in the same network where the BDMSL web services are **not** exposed to the internet.

- Using 2-way HTTPS/TLS (**recommended**).

The BDMSL integration configuration has two parts:

- Configuration of the BDMSL integration data as: BDMLS URL, SMPs URL, etc.

- Configuration of the SMP domain credentials/X509Certificate and unique SMP identifier.

10.1. Configuring the BDMSL Integration

The BDMSL integration data can be set using the UI Property tool:

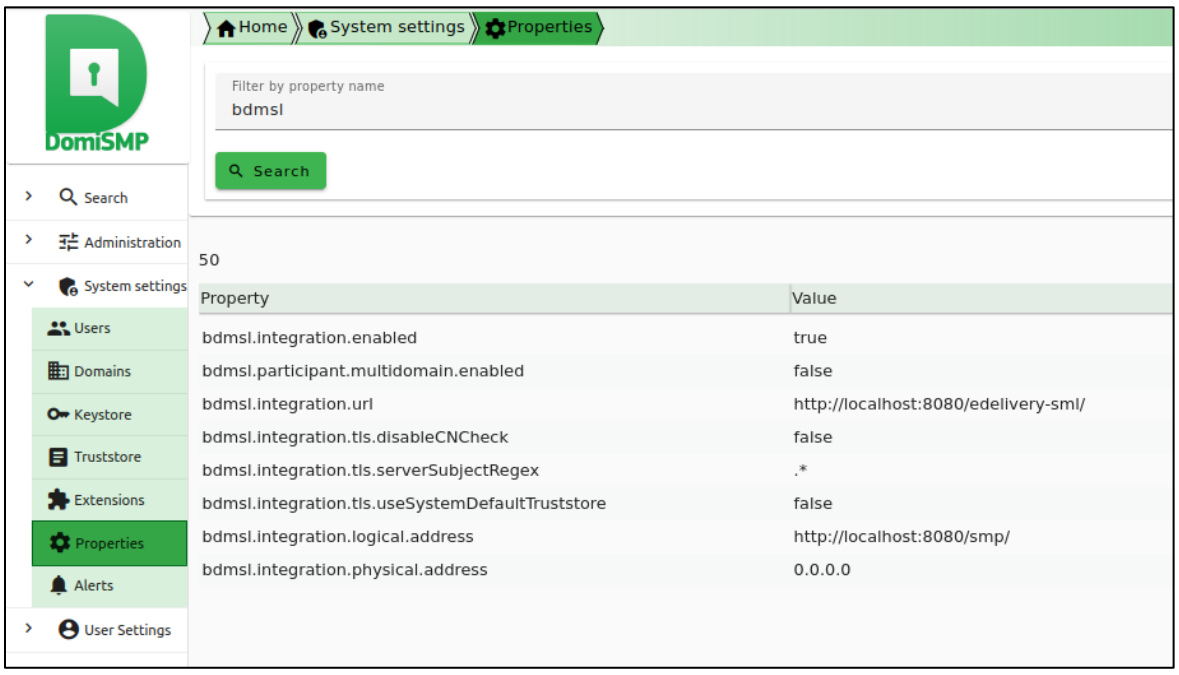


Figure 5 - DomiSMP UI: Property settings page

The following values should be defined for the properties:

- **bdmsl.integration.enabled:** set value to true to enable BDMSL (SML) integration.

- **bdmsl.integration.url:** set the URL where BDMSL is located.
  Ex: https://acc.edelivery.tech.ec.europa.eu/edelivery-sml/

- **bdmsl.integration.logical.address:** set the public SMP URL address. The URL is used by the BDMSL when generating DNS records for the SMP. Do not change this property once the SMP domain is registered to BDMSL. Ex: https://smp.domain.eu/smp

- **bdmsl.integration.physical.address**: ip4 address of the SMP server. The value is informative and can be 0.0.0.0

When using the 2-Way TLS authentication the following parameters should be configured:

- **bdmsl.integration.tls.disableCNCheck:** if set to true, the BDMSL server domain and Certificate CN value must match with the BDMSL certificate to be trusted.

- **bdmsl.integration.tls.useSystemDefaultTruststore**: if set to true, the system default truststore is used to verify the BDMSL truststore. The system default truststore usually points to $JAVA_HOME/lib/security/cacerts truststore, or is configured on the application server using the javax.net.ssl.trustStore system parameter. If the property is set to false, the SMP truststore is used to verify the BDMSL server certificate trust.

- **bdmsl.integration.tls.serverSubjectRegex**: regular expression for BDMSL server TLS certificate subject verification.
  Example: CertEx. .*CN=acc.edelivery.tech.ec.europa.eu.*.

10.2. Configuration of the SMP domain credentials for BDMSL

Once BDMSL integration data is configured, the next step is to configure the SMP client certificate and ID for the BDMSL authentication. Because SMP 4.2 can handle multiple domains, each domain can have its X509Certificate to login to the correct BDMSL DNS domain. The configuration of the SMP domain credentials is described below:

1. Register BDMSL client key/certificate to the SMP Keystore.

2. Create or edit Domain in the UI/Domain tool, enter the SMP ID and choose the client certificate.

3. Choose the authentication type. SML supports two ways of authentication

   - ClientCert: HTTP Client-Cert certificate header. This must be used only behind a reverse proxy. The BDMSL should NOT allow this type of authentication from the internet. In practice, the HTTP Client-Cert should be generated only by the reverse proxy.

   - HTTPS/TLS: standard mutual TLS authentication (recommended).

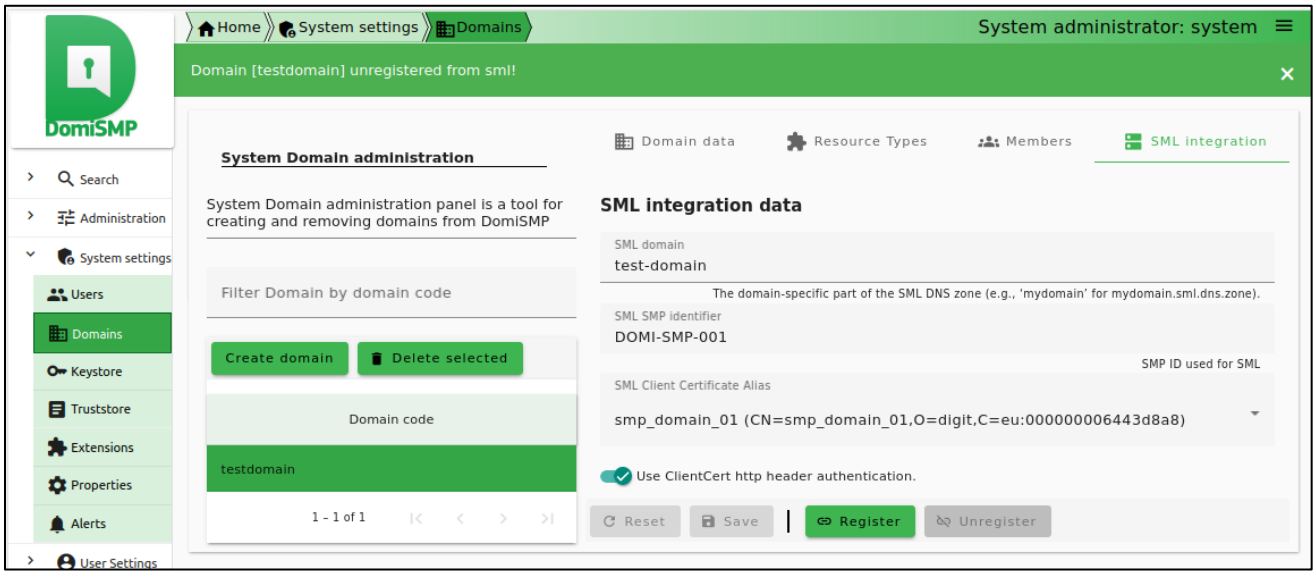


Figure 6 - DomiSMP UI: The Domain SML integration settings

11. SMP User Management

The DomiSMP has two user application roles:

- System Admin: the role allows user to modify DomiSMP system management and settings such as: Domain management, User management, Truststore management, Key management, DomiSMP configuration, etc.

- User: this role allows the user to login to the DomiSMP.

The user can have additional permissions on editing DomiSMP entities when they are assigned to the Resource, Groups, and Domains. Please read the following chapter for more details.

11.1. Domain, Group and Resources

The DomiSMP supports 3-layer security realms.

- The most basic unit is the **Resource**. The Resource is identified by the unique ID, which is part of the URL of the resource as example:

  http://localhost/smp/***resource-identifier***

  An example of the Resource is the "Service Group" document from the Oasis SMP specification. The user can be a Resource member with **Admin** or **Viewer** membership roles. If the user has an Admin membership role, it can modify resource document(s) and manage the resource memberships. If the user has role Viewer, it can view/read the Resource if the Resource has visibility set to: "Private".

- The **Group** is a cluster of resources managed by the dedicated group administrators. The group admin(s) can create and delete the resource, but **only** the resource admins can modify data/documents for the resource. The user can be a Group member with **Admin** or **Viewer** membership roles. With Admin group membership, the user can create and delete group resources. If the user has group role Viewer, it can view/read the Resources if the Group has visibility set to: "Private".

- The top layer is the **Domain**. It indicates the business purpose of the network of participants, such as invoice exchange, Health Records message exchanges, etc. The Domain usually has a domain owner who handles participant interoperability, defining message types, network authentication, and authorization methods such as Certificate PKI, Identity Service providers, etc. In DomiSMP 5.0, the user with a Domain Admin role can create domain groups and assign users to them.

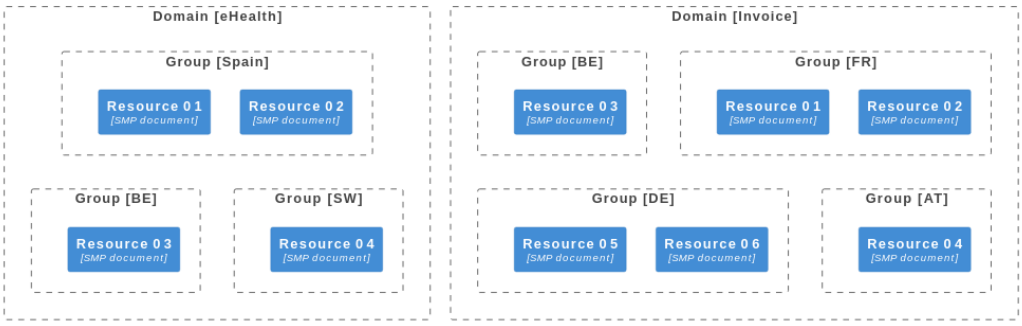


Figure 7 - Example of Domain/Group/Resource overwiew

The provided database script creates the following users:

User name	Role	Default password [2]
system	SYSTEM_ADMIN	123456
user	USER	123456

11.2. User Roles

The following DomiSMP users can be of three types, as briefly described below:

Actor	UC	Short description	Oper.	Data
Group Admin	Create or Update resource: Service Group	Create a new ServiceGroup for a new receiver participant. This service stores the Service Group and links it to the specified duplet participantIdentifier + participantIndentifierScheme the resource identifier. Information is stored into Resource table. This same service is used to create and update a ServiceGroup.	PUT	ServiceGroup
Group SMP	Erase Service Group	Erases the resource (service group definition) AND the list of sub-resources such as servicemetada for the specified receiver participant.	DELETE	ServiceGroup
Resource Admin	Create or Update Resource such as: Service group document and subresources: Service Metadata	Publish detailed information about one specific document service (multiple processes and endpoints). This same service is used to create and update ServiceMetaData.	PUT	ServiceMetadata
Resource Admin	Erase Service Metadata	Remove all information about one specific service (i.e. all related processes and endpoints definitions).	DELETE	ServiceMetadata
Anonymous	Retrieve Service	Obtain the list of public services provided by a specific receiver participant	GET	ServiceGroup

[2] to change immediately for security reasons

Actor	UC	Short description	Oper.	Data
User	Group	(collection of references to the ServiceMetaData's). This service provides the information related to the Service Group according to the input duplet participantIdentifier + participantIndentifierScheme.		
Anonymous User	Retrieve Service Metadata	Obtain detailed definition about one specific service of a specific participant for all supported transports. This service retrieves the SignedServiceMetadata according to the input quadruplet participantIdentifier+participantIndentifierScheme+documentIdentifier+documentIdentifierScheme.	GET	SignedServiceMetadata
System admin		Create, modify, and delete users and domains. System admin can be only used in the DomiSMP UI.		

Note: For a complete description of the SMP user management, please consult the SMP Interface Control Document (ICD) document available at https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/SMP.

Users can be added, modified and deleted using the SMP Admin console or directly by executing sql commands. Below are instructions on how to modify users in the database.

11.3. BCRYPT password generation

The DomiSMP users can be managed by DomiSMP UI console. Following procedure can be used for creating first system admin user (alternative is to use provided SQL init scripts and replace passwords at first login).

The DomiSMP uses the BCRYPT algorithm to hash users' passwords. A BCRYPT-hashing tool is bundled into the SMP WAR file. To get the hashing code, follow the steps below.

Place a copy of the **smp-X.war** file into a temporary directory of your choice.

Extract the war file using the **jar** command:

```
jar -xvf smp-X.war
```

Obtain one or multiple hashes at once, using the following command:

```
java -cp "WEB-INF/lib/*"
eu.europa.ec.edelivery.smp.utils.BCryptPasswordHash password_to_be_hashed
```

The result will be a BCRYPT hash of the specified password (listed below in italic):

```
java -cp "WEB-INF/lib/*"
eu.europa.ec.edelivery.smp.utils.BCryptPasswordHash 123456
Gives:
$2a$10$6nYTSUSh2BQfbOLIyCXn8eUViBcnn.WcjUrWOtJlMNDOdAtI85zMa
```

The next command shows the hashing of several passwords at once, separated by a space in the command.

```
java -cp "WEB-INF/lib/*" eu.europa.ec.edelivery.smp.BCryptPasswordHash
password_to_be_hashed_1 password_to_be_hashed_2
$2a$10$6nYTSUSh2BQfbOLIyCXn8eUViBcnn.WcjUrWOtJlMNDOdAtI85zMa
$2a$107zNzSeZpxiHeqY2BRKkHE.HknfIe3aiu6XzU.qHHnnPbUHKtfcmDG
```

11.4. SMP Database User Creation

Adding an SMP user is done by adding a new entry in the SMP database **SMP_USER** table either directly or via the Administration console.

The User role is set in the SMP_USER table APPLICATION_ROLE column as follows:

User Role	Role value
System Administrator	SYSTEM_ADMIN
Domi SMP user	USER
AnonymousUser (Not defined in the SMP User database)	N/A

In the following examples, an **System Admin** user is created.

11.4.1. *SYSTEM_ADMIN SMP User creation*

Remark:

- In order to logon on the Administration Console **for the first time**, it is necessary to, create a user with **SYSTEM_ADMIN** privileges by entering the details directly into the **SMP_USER** table. This initial user's password is generated using the **BCRYPT** utility described previously.
- *If PASSWORD_CHANGED is not set, the user will be asked to change the password at first logon.*

Example of a SYSTEM_ADMIN user creation:

Username : smp_admin

Password (Hashed) : $2a$10$6nYTSUSh2BQfbOLIyCXn8eUViBcnn.WcjUrWOtJlMNDOdAtI85zMa

Role : SYSTEM_ADMIN

Execute the following database command using the database user/password created in the Database Configuration section of this guide.

MySql example:

```
insert into SMP_USER (USERNAME, ACTIVE, APPLICATION_ROLE, EMAIL, CREATED_ON, LAST_UPDATED_ON) values
(1,'smp_admin', 1, 'SYSTEM_ADMIN', 'system@mail-example.local',  NOW(),  NOW());

insert into SMP_CREDENTIAL (FK_USER_ID, CREDENTIAL_ACTIVE, CREDENTIAL_NAME, CREDENTIAL_VALUE,
CREDENTIAL_TYPE, CREDENTIAL_TARGET, CREATED_ON, LAST_UPDATED_ON) values
((select id from SMP_USER where USERNAME='smp_admin'),1,  'smp_admin',
'$2a$10$olcGeWKGEoRia2DPuFqRNeca0IEdRSmOrljLz57BAjf1jlC9SohrS', 'USERNAME_PASSWORD','UI',  NOW(),
NOW());
```

Oracle example:

```
insert into SMP_USER (ID, USERNAME, ACTIVE, APPLICATION_ROLE, EMAIL, CREATED_ON, LAST_UPDATED_ON)
values
(SMP_USER_SEQ.NEXTVAL, 'smp_admin', 1, 'SYSTEM_ADMIN', 'system@mail-example.local', sysdate,
sysdate);

insert into SMP_CREDENTIAL (FK_USER_ID, CREDENTIAL_ACTIVE, CREDENTIAL_NAME, CREDENTIAL_VALUE,
CREDENTIAL_TYPE, CREDENTIAL_TARGET, CREATED_ON, LAST_UPDATED_ON) values
((select id from SMP_USER where USERNAME='smp_admin'),1,  'smp_admin',
'$2a$10$olcGeWKGEoRia2DPuFqRNeca0IEdRSmOrljLz57BAjf1jlC9SohrS', 'USERNAME_PASSWORD','UI',  sysdate,
sysdate);
```

Note: The username/password credential is stored in table: SMP_CREDENTIAL.
The record must have the following values set to:

- CREDENTIAL_VALUE: the BCrypted password
- CREDENTIAL_TYPE: value must be set to: 'USERNAME_PASSWORD'
- CREDENTIAL_TARGET: value must be set to: 'UI'
- FK_USER_ID: value must be set to user id.

LOGGING CONFIGURATION

11.5. Logging properties

The SMP logging properties are defined in the ***./WEB-INF/classes/logback.xml*** file embedded in the SMP **.war** file.

It is possible to modify the configuration of the logs by editing the embedded ***logback.xml*** or by defining new logback file in **smp.config.properties** file as example:

```
log.configuration.file=/opt/apache-tomcat-8.5.30/smp/logback.xml
```

In the example below, a ***logback.xml*** file is shown:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <!-- pattern definition -->
    <property name="encoderPattern" value="%d{ISO8601} [%X{smp_user}] [%X{smp_session_id}] [%X{smp_request_id}] [%thread] %5p %c{1}:%L - %m%n" scope="global"/>
    <property name="consolePattern" value="%d{ISO8601} [%X{smp_user}] [%X{smp_session_id}] [%X{smp_request_id}] [%thread]  %5p %c{1}:%L - %m%n" scope="global"/>

     <appender name="file" class="ch.qos.logback.core.rolling.RollingFileAppender">
        <file>${log.folder:-logs}/edelivery-smp.log</file>
        <filter class="ch.qos.logback.core.filter.EvaluatorFilter">
            <evaluator class="ch.qos.logback.classic.boolex.OnMarkerEvaluator">
                <marker>SECURITY</marker>
                <marker>BUSINESS</marker>
            </evaluator>
            <onMismatch>NEUTRAL</onMismatch>
            <onMatch>DENY</onMatch>
        </filter>
        <rollingPolicy class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPolicy">
            <!-- rollover daily -->
            <fileNamePattern>${log.folder:-logs}/edelivery-smp-%d{yyyy-MM-dd}.%i.log</fileNamePattern>
            <!-- each file should be at most 30MB, keep 60 days worth of history, but at most 20GB -->
            <maxFileSize>30MB</maxFileSize>
            <maxHistory>60</maxHistory>
            <totalSizeCap>20GB</totalSizeCap>
        </rollingPolicy>
        <encoder>
            <pattern>${encoderPattern}</pattern>
        </encoder>
    </appender>
    <appender name="stdout" class="ch.qos.logback.core.ConsoleAppender">
        <Target>System.out</Target>
        <encoder>
            <pattern>${consolePattern}</pattern>
        </encoder>
    </appender>

    <logger name="eu.europa.ec.edelivery.smp" level="INFO" />
    <logger name="org.springframework.security.cas" level="DEBUG" />
    <root level="DEBUG">
        <appender-ref ref="file"/>
        <appender-ref ref="stdout"/>
    </root>
</configuration>
```

More details on how to configure logback can be found at:

https://logback.qos.ch/documentation.html

12. SOAPUI TESTING

Soap UI can be used to create, update and delete Service Groups and Metadata.

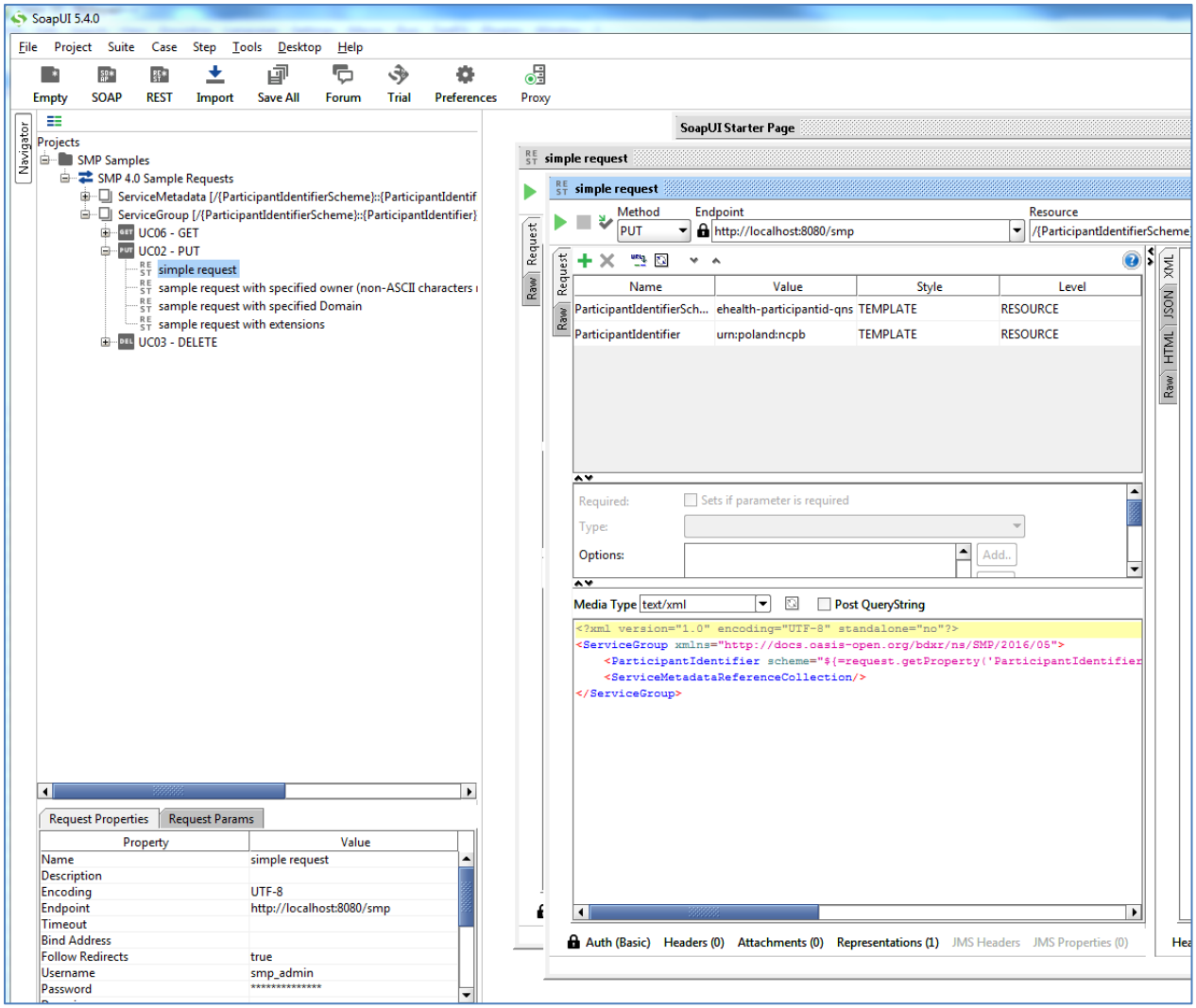
An SMP SoapUI project contains sample requests and is included in the zip file already downloaded.

The procedure to create, update or delete a Service Group is described in the next steps.

12.1. Creation, update and deletion of Service Groups.

12.1.1. *Create a Service Group*

In the left navigation pane of the SoapUI interface, browse to the REST PUT method as shown below:

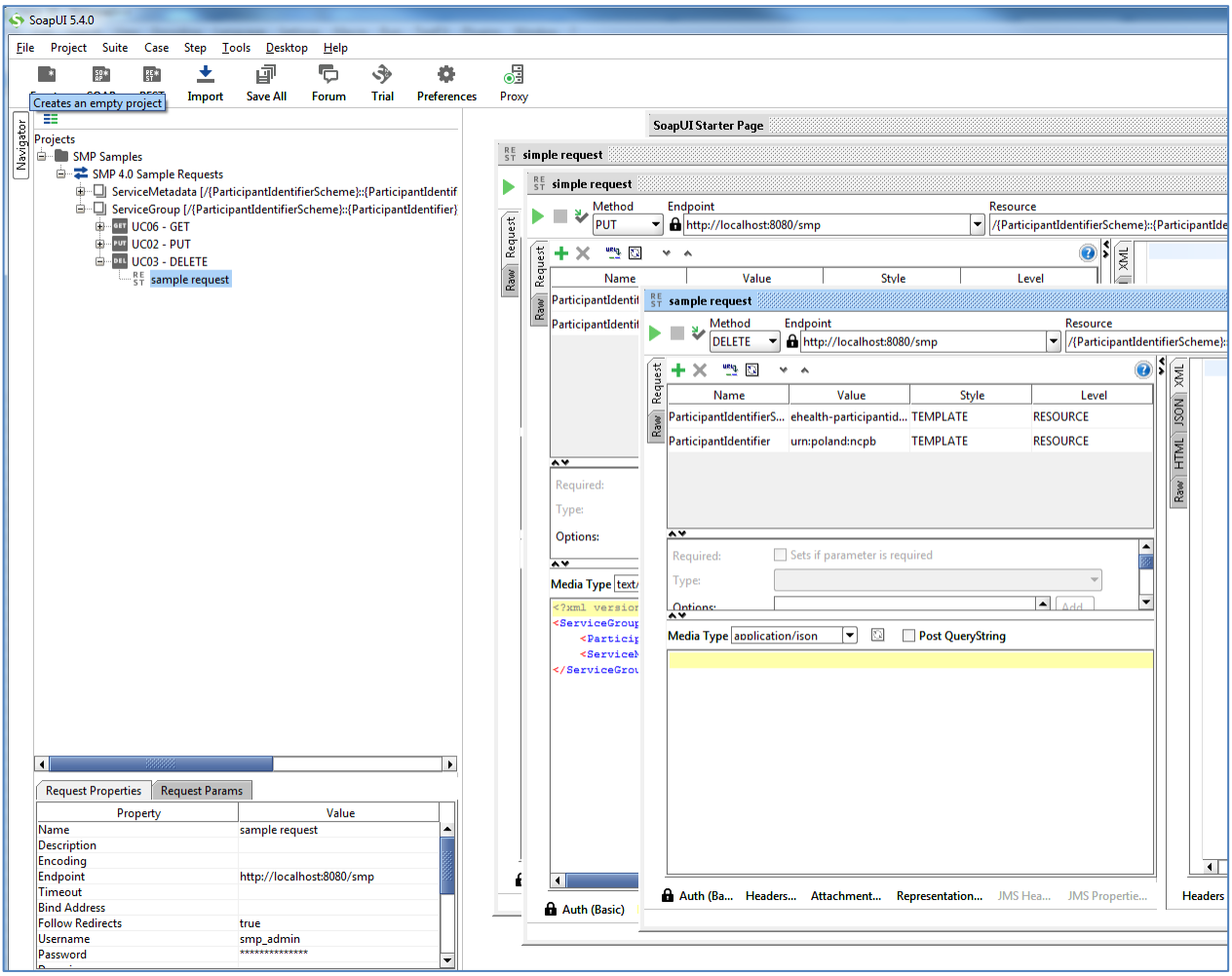


12.1.2. *Update a Service Group*

The REST method to update the **ServiceGroup** is the same as the one used for creating **ServiceGroup** described in the previous section.

12.1.3. *Delete a ServiceGroup*

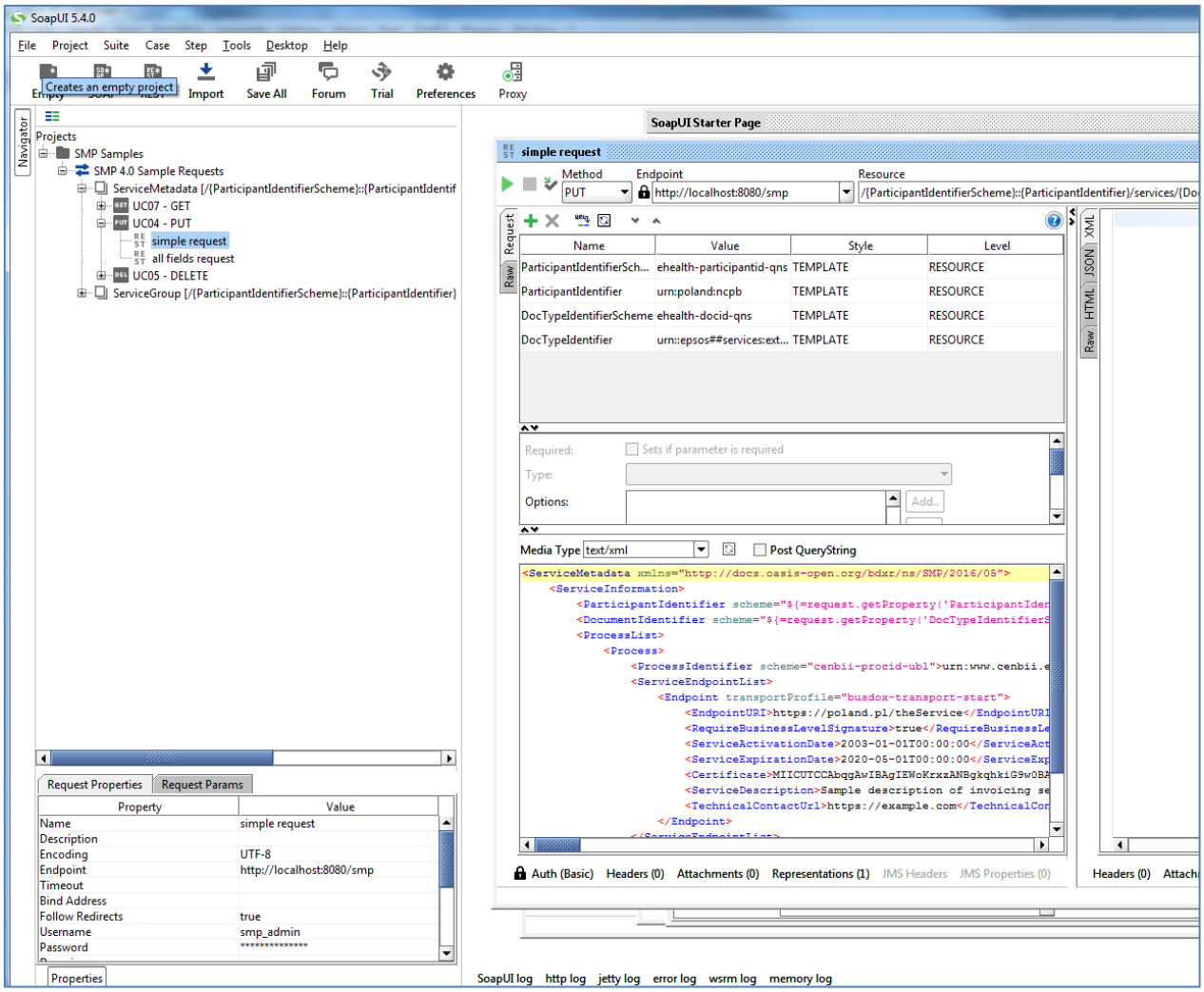
On the SoapUI interface on the left navigation panel, browse to the REST DELETE method as indicated below:



12.2. Creation, update and deletion of Service Metadata.

12.2.1. *Create a Service Metadata*

In the left navigation pane of the SoapUI interface, browse to the REST PUT method as shown below:

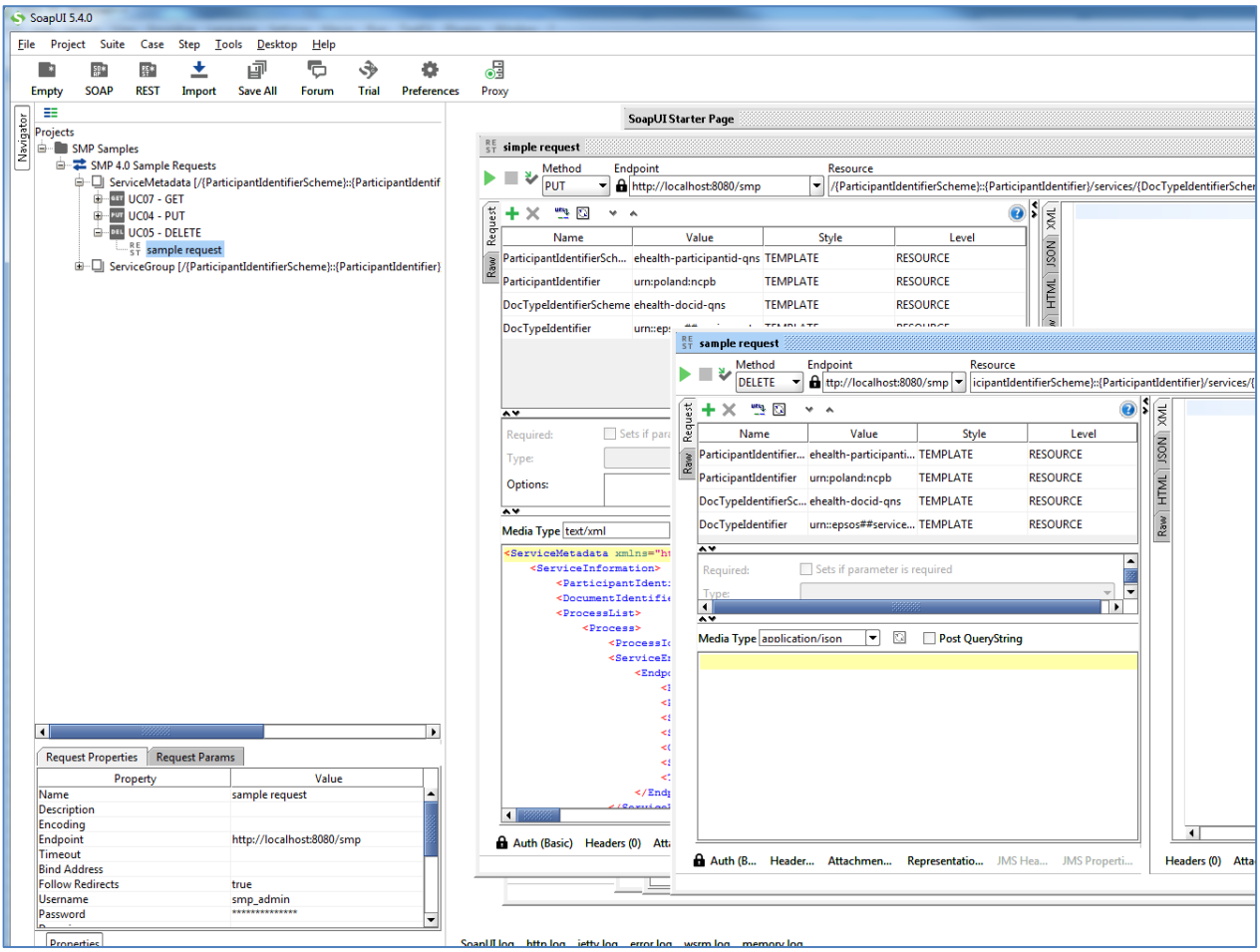


12.2.2. *Update Service Metadata*

The REST method to update **ServiceMetadata** is the same as the one use for creating **ServiceMetadata** as described in the previous section.

12.2.3. *Delete Service Metadata*

In the left navigation pane of the SoapUI interface, browse to the **REST DELETE** method as indicated below:



13. THE SWAGGERUI INTERFACE

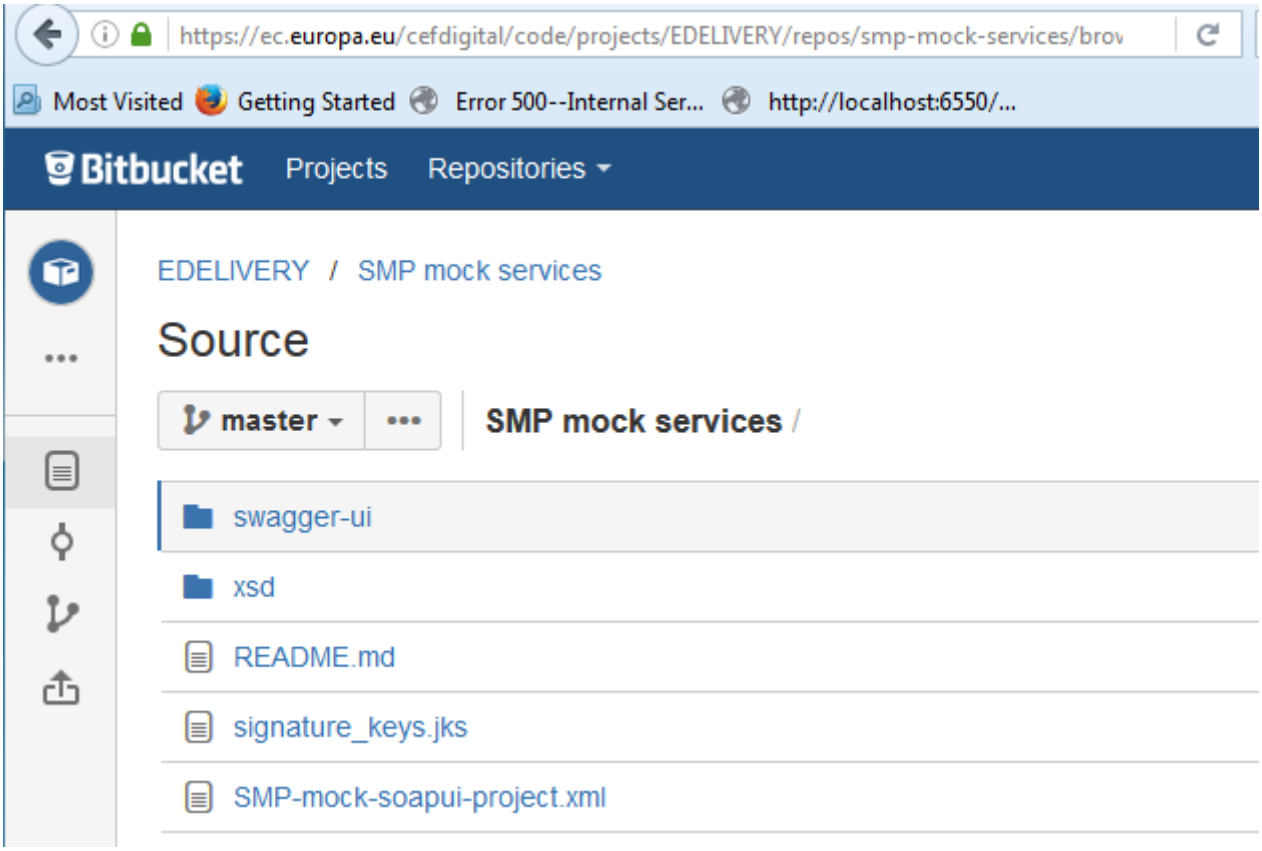
13.1. Introduction

"***Swagger is an API developer tools for the OpenAPI Specification (OAS). It allows anyone (developers or end-users) to interact with the API's resources***"[3].

13.2. Downloading the eDelivery SMP SwaggerUI web application project

The eDelivery SMP SwaggerUI web application project can be freely downloaded from the following location:

https://ec.europa.eu/digital-building-blocks/code/projects/EDELIVERY/repos/smp-mock-services/browse



Create a new **swagger_temp** temporary directory.

Within the previously created **swagger_temp** directory, execute the following command:

```
git clone https://ec.europa.eu/cefdigital/code/scm/edelivery/smp-mock-services.git
```

[3] Quote from: http://swagger.io/.

```
Cloning into 'smp-mock-services'...
remote: Counting objects: 133, done.
remote: Compressing objects: 100% (130/130), done.
remote: Total 133 (delta 50), reused 0 (delta 0)
Receiving objects: 100% (133/133), 823.54 KiB | 0 bytes/s, done.
Resolving deltas: 100% (50/50), Done.
```

The SMP **SwaggerUI** project is downloaded and saved the **smp-mock-services** directory:

```
ls
smp-mock-services
```

13.3. Configuring the SMP SwaggerUI

Navigate to the **swagger-ui** directory located under the **smp-mock-services** directory.

The contents is listed below:

```
ls
css  fonts  images  index.html  lib  smp.json  swagger-ui.js
```

Edit the **smp.json** file and modify it to target your SMP:

Replace:

```
{
  "swagger": "2.0",
  "info": {
    "description": "This WEB client is configured to shoot at the [mocked SMP implementation](http://smp-digit-mock.publisher.ehealth.acc.edelivery.tech.ec.europa.eu/ehealth-actorid-qns%3A%3Aurn%3Apoland%3Ancpb). After a few improvements (both on client and server side) it might be used also for shooting at TEST / PROD environments. You can find out more about Swagger at [http://swagger.io](http://swagger.io)",
    "version": "1.0.0",
    "title": "SMP 3.X WEB client (based on Swagger-UI)"
  },
  "host": "smp-digit-mock.publisher.ehealth.acc.edelivery.tech.ec.europa.eu",
  "basePath": "/",
  "externalDocs": {
    "description": "Find out more about SMP 3.X mock services",
```

With:

```
    "url": "https://ec.europa.eu/cefdigital/code/projects/EDELIVERY/repos/smp-mock-services"
{
  "swagger": "2.0",
  "info": {
    "description": "This WEB client is configured to shoot at [http://localhost:7003/ smp-X](http://localhost:7003/ smp-X).  After a few improvements (both on client and server side) it might be used also for shooting at TEST / PROD environments. You can find out more about Swagger at [http://swagger.io](http://swagger.io)",
    "version": "1.0.0",
    "title": "SMP X WEB client (based on Swagger-UI)"
  },
  "host": "localhost:7003",
  "basePath": "/ smp-X",
  "externalDocs": {
  },
```

13.4. Generating the Web Application Archive (.war file)

To generate the eDelivery SMP SwaggerUI Web Application archive (**.war** file), jus create a zip file of the content of the swagger-ui directory and rename it as **swagger.war**.

This can be performed using any **zip** utility (**winzip** on Windows or **zip** on Linux).

Example on Linux:

```
zip -r swagger.war swagger-ui/*
```

13.5. Deploy the SMP SwaggerUI war file

13.5.1. *On Tomcat*

Copy the **swagger.war** file to *AS_HOME* **/webapps.**

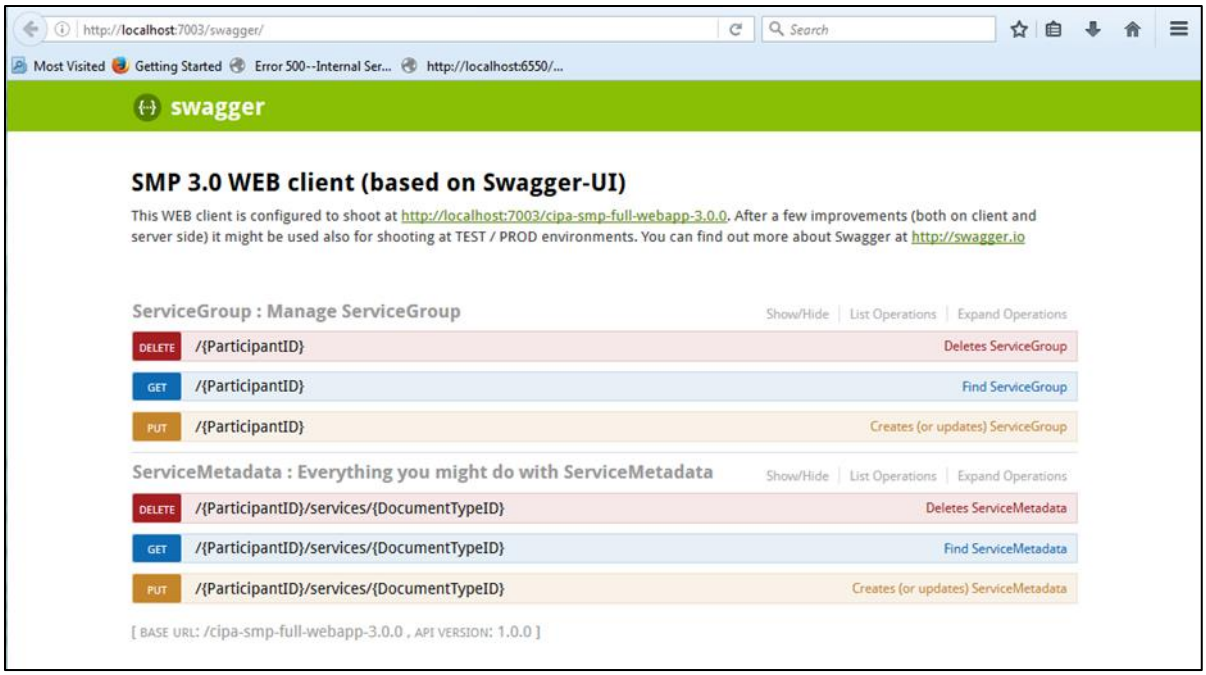
13.5.2. *On WebLogic:*

Deploy the **.war** file within WebLogic:

```
java weblogic.Deployer -adminurl
t3://${WebLogicAdminServerListenAddress}:${WebLogicAdminServerPort} \
-username ${WebLogicAdminUserName} \
-password ${WebLogicAdminUserPassword} \
-deploy -name swagger.war \
-targets ${SMP_ManagedServer} \
```

After starting the application, connect to http://locxalhost:7003/swagger.

A successful deployment should display the following page:



14. SMP COMPILATION

14.1. Compilation prerequisites

14.1.1. *Supported Operating System Platform*

The eDelivery SMP can be built on the following OS platforms:

- Windows Workstation & Server
- Linux platform

14.1.2. *Software Requirements*

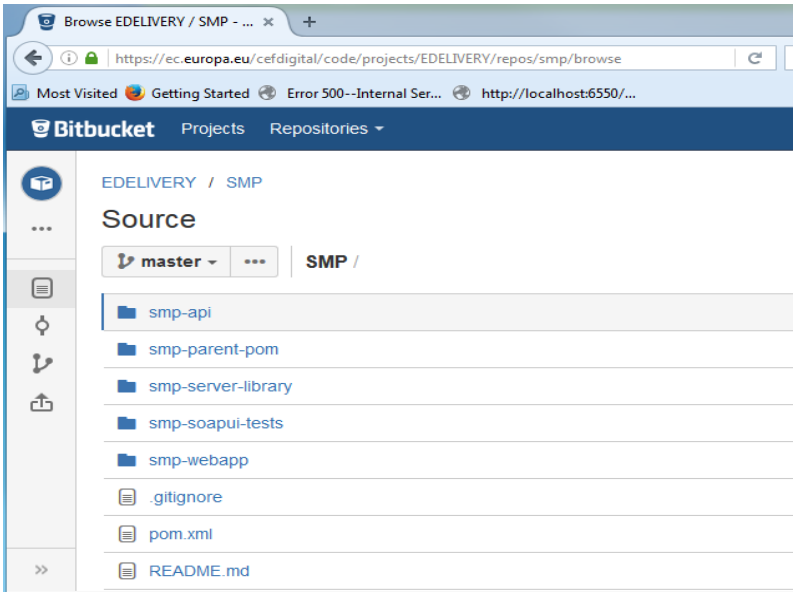
The following software components on the target system:

- Java Development Kit environment (JDK), version 8: http://www.oracle.com/technetwork/java/javase/downloads/index.html
- Maven 3.6 and above (https://maven.apache.org/download.cgi)
- GIT (optional: Git is only used to download the project sources but these sources can be downloaded from any system having Git installed and then just copied manually on the compilation platform).

14.2. Downloading the source code

The source code of SMP is freely available and can be downloaded from the following location:

https://ec.europa.eu/digital-building-blocks/code/projects/EDELIVERY/repos/smp/browse



14.3. Compilation

Create a new **comp_dir** temporary directory.

Within the previously created **comp_dir** directory, execute the following command:

```
git clone  https://ec.europa.eu/digital-building-blocks/code/scm/edelivery/smp.git
Cloning into 'smp'...
remote: Counting objects: 52788, done.
remote: Compressing objects: 100% (15640/15640), done.
remote: Total 52788 (delta 25293), reused 47993 (delta 23387)
Receiving objects: 100% (52788/52788), 637.14 MiB | 2.06 MiB/s, done.
Resolving deltas: 100% (25293/25293), done.
```

Go to the newly created **smp** directory.

The directory contains the following:

```
ls
pom.xml  README.md  smp-api  smp-parent-pom  smp-server-library  smp-soapui-tests  smp-webapp
```

Start the compilation by executing the following command:

```
mvn clean install -DskipTests
```

A successful compilation will result with the following:

```
mvn clean install -DskipTests
[INFO] Scanning for projects…
/..
../
[INFO] Installing /home/smpcomp/smp/smp/pom.xml to /home/smpcomp/.m2/repository/eu/europa/ec/smp/3.X/smp-3.X.pom
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] smp-angular ....................................... SUCCESS [132.375 s]
[INFO] smp-api ........................................... SUCCESS [ 32.375 s]
[INFO] smp-server-library ................................ SUCCESS [02:01 min]
[INFO] smp-webapp ........................................ SUCCESS [ 23.314 s]
[INFO] SMP Builder POM ................................... SUCCESS [  2.222 s]
[INFO] ------------------------------------------------------------------------
```

```
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 03:00 min
[INFO] Finished at: 2017-06-08T11:35:27+02:00
[INFO] Final Memory: 61M/726M
[INFO] ------------------------------------------------------------------------
```

The resulting will be a Web application Archive (.war file) named **smp.war** located in the **smp-webapp/target/** directory:

```
ls ./smp-webapp/target
smp-X  smp.war classes  generated-sources  generated-test-sources  maven-status
test-classes  webapp-classes
```

15. SMP CONFIGURATION FILE AND TABLE

15.1. Multitenancy and Multidomain Support

The SMP is able to support multiple certificates in the same SMP. This is very useful in the Acceptance environment where multiple domains like ISA ITB, eHealth and others are hosted.

The SMP has the capability of keeping a relationship between a particular **Service Group** and its related **domain**.

As a result of this feature, the SMP Administration has the option, if need be, to define extra domains for newly created **Service Groups** meaning that the SMP can handle multiple domains environments.

Remark:

In normal circumstances, when any one SMP is used for only one domain, the domain used is then considered as the "domain by default" (or "default domain") for configuration purposes. The domain, in this case, does not need to be specified in the **Service Group** definitions or other configurations of the SMP as in previous versions of SMP.

The SMP configuration is performed in 2 different locations: in the **smp.config.properties** file as well as in the **smp_configuration** table. The following section describes the details of the parameters that are included in the configuration.

15.2. The smp.config.properties file

The eDelivery SMP configuration is performed via the **smp.config.properties** file.

The initial eDelivery SMP configuration is performed via the **smp.config.properties** file. The file contains basic configuration for defining the database connection, logging file configuration and smp folder for deploying the extensions.

This file is delivered by default embedded within the SMP war file.

```
#
# Copyright 2018 European Commission | CEF eDelivery
#
# Licensed under the EUPL, Version 1.2 or - as soon they will be approved
by the European Commission - subsequent versions of the EUPL (the
"Licence");
# You may not use this work except in compliance with the Licence.
#
# You may obtain a copy of the Licence attached in file: LICENCE-EUPL-
v1.2.pdf
#
```

```
# Unless required by applicable law or agreed to in writing, software
distributed under the Licence is distributed on an "AS IS" basis,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the Licence for the specific language governing permissions and
limitations under the Licence.
#


#
********************************************************************************
# Database connection can be achieved using custom datasource
configuration
# or reusing application server datasource.
#
********************************************************************************
## set database hibernate dialect
# smp.database.hibernate.dialect=org.hibernate.dialect.Oracle10gDialect
smp.database.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect


# *********************************
#  Custom defined datasource
# *********************************
# mysql database example
smp.jdbc.driver = com.mysql.jdbc.Driver
smp.jdbc.url = jdbc:mysql://localhost:3306/smp
smp.jdbc.user = smp
smp.jdbc.password=secret123


# Oracle database example
# smp.jdbc.driver = oracle.jdbc.driver.OracleDriver
# smp.jdbc.url=jdbc:oracle:thin:@localhost:1521/xe
# smp.jdbc.user=smp
# smp.jdbc.password=secret123



# *********************************
# Datasource JNDI configuration alternative
# *********************************
# weblogic datasource JNDI example
# smp.datasource.jndi=jdbc/eDeliverySmpDs
```

```
# tomcat datasource JNDI example
# smp.datasource.jndi=java:comp/env/jdbc/eDeliverySmpDs



# ********************************
# Logging properties
# ********************************
# smp log folder
smp.log.folder=../logs/

# custom log4j configuration file
# smp.log.configuration.file=smp-logback.xml

# ********************************
# Extension folder
# ********************************
# path where SMP extensions are located. The Folder is loaded by the SMP classloader at startup.
smp.libraries.folder=/cef/test/smp/apache-tomcat-8.5.73/smp/ext-lib
```

15.2.1. *SMP configuration properties (smp.config.properties)*

The **WEB-INF/classes/smp.config.properties** file is used to configure the initial SMP properties needed for the SMP startup.

The following table describes them briefly:

Parameter	Default Value	Comment
smp.configuration.file	smp.conf.properties	Configuration property file path.
smp.init.configuration.file	smp.init.properties	Init configuration property file path.
smp.security.folder	smp	Security folder for storing the keystore and the truststore.
smp.jdbc.driver	com.mysql.jdbc.Driver	Database Configuration: Driver MySQL: com.mysql.jdbc.Driver Oracle Database: oracle.jdbc.OracleDriver
smp.jdbc.url	jdbc:mysql://localhost:3306/smp	Database Configuration: url MySQL : jdbc:mysql://dbhost:dbport/smp_database Oracle Database: jdbc:oracle:thin:@dbhost:dbport:smp_database jdbc:oracle:thin:@dbhost:dbport/smp_service
smp.jdbc.user	smp	Database User/Password Configuration: User
smp.jdbc.password	The_password	Database User/password Configuration: Password
smp.datasource.jndi	jdbc/eDeliverySmpDs	If the data source is configured on the application server (recommended),

Parameter	Default Value	Comment
		the property defines the JNDI name of the database connection.
smp.database.show-sql	false	Print generated sql queries to logs. The property is effective only when smp.mode.development=true.
smp.database.create-ddl	false	Auto create/update database objects. The property is effective only when smp.mode.development=true. NOTE: Do NOT use this in production: this feature is only for test, demo and development purposes.
smp.log.folder	/var/logs/smp	The provided logback.xml configuration defines logging file as <file>${log.folder:-logs}/edelivery-smp.log</file> With the property we can define the folder for the logging files.
smp.log.configuration.file	/opt/logging/smp-logback.xml	Custom logback configuration file (filepath can be absolute or relative to smp configuration.dir).
smp.libraries.folder	/opt/smp/extension-libs	Path where SMP extensions are located. The folder is loaded by the SMP classloader at startup.
smp.smp.mode.development	false	The development mode uses semi-random generators for password and key generation. Setting the property value to ‘true‘ makes the first startup and access token generation faster. To ensure high security, this option MUST NOT be enabled in production.

15.2.2. *SMP application configuration (database table SMP_CONFIGURATION)*

eDelivery SMP application configuration values are stored in the database table ***SMP_CONFIGURATION***. If the table is empty (usually at first SMP startup), edelivery SMP populates the table at startup with all properties and default values. When updating properties via the user interface, the property values are taken into account immediately if the server starts in non-cluster mode (property***: smp.cluster.enabled = false***). Otherwise, each node refreshes the properties on all cluster nodes at the same time) according to the property refreshes the cron expression in the property: ***smp.property.refresh.cronJobExpression.***

Parameter	Default Value	Comment	Restart needed	Value type
contextPath.output	true	This property controls pattern of URLs produced by SMP in GET ServiceGroup responses.	true	BOOLEAN
encodedSlashesAllowedInUrl	true	Allow encoded slashes in context path. Set to true if slashes are are part of identifiers.	true	BOOLEAN
smp.http.forwarded.headers.enabled	false	Use (value true) or remove (value false) forwarded headers. There are security considerations for forwarded headers since an application cannot know if the headers were added by a proxy, as intended, or by a malicious client.	false	BOOLEAN
smp.http.httpStrictTransportSecurity.maxAge	31536000	How long (in seconds) should HSTS last in the browser cache (default one year).	true	INTEGER
smp.http.header.security.policy		Content Security Policy (CSP) default-src 'self'; script-src 'self'; connect-src 'self'; img-src 'self'; style-src 'self' 'unsafe-inline'; frame-ancestors 'self'; form-action 'self';	true	STRING
smp.proxy.host		The http proxy host.	false	STRING
smp.noproxy.hosts	localhost|127.0.0.1	list of nor proxy hosts. Ex.: localhost|127.0.0.1	false	STRING
smp.proxy.password		Base64 encrypted password for Proxy.	false	STRING
smp.proxy.port	80	The http proxy port.	false	INTEGER

Parameter	Default Value	Comment	Restart needed	Value type
smp.proxy.user		The proxy user.	false	STRING
identifiersBehaviour.ParticipantIdentifierScheme.validationRegex	^$\|^(?!^.{26})([a-z0-9]+-[a-z0-9]+-[a-z0-9]+)$|^urn:oasis:names:tc:ebcore:partyid-type:(iso6523|unregistered)(:.+)?$	Participant Identifier Schema of each PUT ServiceGroup request is validated against this schema.	false	REGEXP
identifiersBehaviour.ParticipantIdentifierScheme.validationRegexMessage	Participant scheme must start with:urn:oasis:names:tc:ebcore:partyid-type:(iso6523:|unregistered:) OR must be up to 25 characters long with form [domain]-[identifierArea]-[identifierType] (ex.: 'busdox-actorid-upis') and may only contain the following characters: [a-z0-9].	Error message for UI.	false	STRING
identifiersBehaviour.scheme.mandatory	true	Scheme for participant identifier is mandatory.	false	BOOLEAN
identifiersBehaviour.ParticipantIdentifierScheme.ebCoreId.concatenate	false	Concatenate ebCore party id in XML responses <ParticipantIdentifier>urn:oasis:names:tc:ebcore:partyid-type:unregistered:test-ebcore-id</ParticipantIdentifier>	false	BOOLEAN

Parameter	Default Value	Comment	Restart needed	Value type
identifiersBehaviour.caseSensitive.ParticipantIdentifierSchemes	sensitive-participant-sc1|sensitive-participant-sc2	Specifies schemes of participant identifiers that must be considered CASE-SENSITIVE.	false	LIST_STRING
identifiersBehaviour.caseSensitive.DocumentIdentifierSchemes	casesensitive-doc-scheme1|casesensitive-doc-scheme2	Specifies schemes of document identifiers that must be considered CASE-SENSITIVE.	false	LIST_STRING
identifiersBehaviour.splitPattern	^(?i)\\s*?(?<scheme>urn:oasis:names:tc:ebcore:partyid-type:(iso6523:[0-9]{4}|unregistered(:[^:]+)?))::?(?<identifier>.+)?\\s*$	Regular expression with groups <scheme> and <identifier> for splitting the identifiers to scheme and identifier part.	false	REGEXP
identifiersBehaviour.ParticipantIdentifierScheme.urn.concatenate		Regular expression to detect URN party identifiers. If the party identifier schema matches the regexp, then the party identifier is concatenated with a single colon in XML responses. Else it is handled as OASIS SMP party identifier. Example: ^(?i)(urn:)|(mailto:).*$	false	REGEXP
bdmsl.integration.enabled	false	BDMSL (SML) integration ON/OFF switch.	false	BOOLEAN
bdmsl.integration.url	http://localhost:8080/edelivery-sml	BDMSL (SML) endpoint.	false	URL
bdmsl.integration.tls.disableCNCheck	false	If SML Url is HTTPs - Disable CN check if needed.	false	BOOLEAN
bdmsl.integration.tls.serverSubjectRegex	.*	Regular expression for server TLS certificate subject verification CertEx. .*CN=acc.edelivery.tech.ec.europa.eu.*.	false	REGEXP
bdmsl.integration.logical.address	http://localhost:8080/smp/	Logical SMP endpoint which will be registered on SML when registering new domain.	false	URL

Parameter	Default Value	Comment	Restart needed	Value type
bdmsl.integration.physical.address	0.0.0.0	Physical SMP endpoint which will be registered on SML when registering new domain.	false	STRING
bdmsl.integration.tls.useSystemDefaultTruststore	false	If the value is 'true', the system default truststore is used for trusting TLS server certificate (Legacy behaviour to SMP 4.1 version), if the value is 'false', the SMP truststore is used.	false	BOOLEAN
bdmsl.integration.naptr_service.map	edelivery-oasis-cppa-3.0-cpp:meta:cppa3	DNS NAPTR service value for the resource types. Property values are "key:value" strings separated with '|', where key is document type identifier and value is NAPTR service value. Ex.: edelivery-oasis-cppa3-extension:meta:cppa3|custom-doc-type:META:ctd.	False	MAP_STRING
smp.keystore.password		Encrypted keystore (and keys) password.	false	STRING
smp.keystore.filename	smp-keystore.p12	Keystore filename.	false	FILENAME
smp.keystore.type	PKCS12	Keystore type as JKS/PKCS12.	false	STRING
smp.truststore.password		Encrypted truststore password.	false	STRING
smp.truststore.filename	smp-truststore.p12	Truststore filename.	false	FILENAME
smp.truststore.type	PKCS12	Truststore type as JKS/PKCS12.	false	STRING
smp.certificate.crl.force	false	If false, then if CRL is not reachable ignore CRL validation.	false	BOOLEAN
encryption.key.filename	encryptionPrivateKey.private	Key filename to encrypt passwords.	true	FILENAME
smp.keystore.password.decrypted		Only for backup purposes when password is automatically created. Store the password somewhere and delete this entry.	false	STRING
smp.truststore.password.decrypted		Only for backup purposes when password is automatically	false	STRING

Parameter	Default Value	Comment	Restart needed	Value type
		created. Store password somewhere and delete this entry.		
smp.certificate.validation.allowedCertificatePolicyOIDs		List of certificate policy OIDs separated by | where at least one must be in the CertifictePolicy extension.	false	STRING
smp.certificate.validation.subjectRegex	.*	Regular expression to validate subject of the certificate.	false	REGEXP
smp.property.refresh.cronJobExpression	0 48 */1 * * *	Property refresh cron expression (def 12 minutes to each hour). Property change is refreshed at restart.	true	CRON_EXPRESSION
smp.ui.session.secure	false	Cookie is only sent to the server when a request is made with the https: scheme (except on localhost), and therefore is more resistant to man-in-the-middle attacks.	false	BOOLEAN
smp.ui.session.max-age		Number of seconds until the cookie expires. A zero or negative number will expire the cookie immediately. Empty value will not set parameter.	false	INTEGER
smp.ui.session.strict	Lax	Controls whether a cookie is sent with cross-origin requests, providing some protection against cross-site request forgery attacks. Possible values are: Strict, None, Lax (cookies with SameSite=None require a secure context/HTTPS).	false	STRING
smp.ui.session.path		A path that must exist in the requested URL, or the browser will not send the Cookie header. Null/Empty value sets the authentication requests context by default. The forward slash (/) character is interpreted as a directory separator, and subdirectories will be matched as well: for Path=/docs, /docs, /docs/Web/, and /docs/Web/HTTP will all match.	false	STRING
smp.ui.session.idle_timeout.admin	300	Specifies the time, in seconds, between client requests before the SMP will invalidate session for ADMIN users (System).	false	INTEGER
smp.ui.session.idle_timeout.user	1800	Specifies the time, in seconds, between client requests	false	INTEGER

Parameter	Default Value	Comment	Restart needed	Value type
		before the SMP will invalidate session for users (Service group, SMP Admin).		
smp.cluster.enabled	false	Define if application is set in cluster. In not cluster environment, properties are updated on set Property.	false	BOOLEAN
smp.passwordPolicy.validationRegex	^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[~`!@#$%^&+=\-_<>.,?:;*/()\|\[\]{}'"\\]).{16,32}$	Password minimum complexity rules.	false	REGEXP
smp.passwordPolicy.validationMessage	Minimum length: 16 characters;Maximum length: 32 characters;At least one letter in lowercase;At least one letter in uppercase;At least one digit;At least one special character	The error message shown to the user in case the password does not follow the regex put in the domibus.passwordPolicy.pattern property	false	STRING
smp.passwordPolicy.validDays	90	Number of days password is valid.	false	INTEGER
smp.passwordPolicy.warning.beforeExpiration	15	How many days before expiration should the UI warn users at login.	false	INTEGER
smp.passwordPolicy.expired.forceChange	true	Force change password at UI login if expired.	false	BOOLEAN
smp.user.login.fail.delay	1000	Delay response in ms on invalid username or password.	false	INTEGER

Parameter	Default Value	Comment	Restart needed	Value type
smp.user.login.maximum.attempt	5	Number of console login attempt before the user is deactivated.	false	INTEGER
smp.user.login.suspension.time	3600	Time in seconds for a suspended user to be reactivated. (If 0 the user will not be reactivated).	false	INTEGER
smp.accessToken.validDays	60	Number of days access token is valid is valid.	false	INTEGER
smp.accessToken.login.maximum.attempt	10	Number of accessToken login attempt before the accessToken is deactivated.	false	INTEGER
smp.accessToken.login.suspension.time	3600	Time in seconds for a suspended accessToken to be reactivated. (If 0 the user will not be reactivated).	false	INTEGER
smp.accessToken.login.fail.delay	1000	Delay in ms on invalid token id or token.	false	INTEGER
smp.ui.authentication.types	PASSWORD	Set list of '|' separated authentication types: PASSWORD|SSO.	false	LIST_STRING
smp.automation.authentication.types	TOKEN|CERTIFICATE	Set list of '|' separated application-automation authentication types (Web-Service integration). Currently supported TOKEN, CERTIFICATE: ex. TOKEN|CERTIFICATE.	false	LIST_STRING
smp.automation.authentication.external.tls.clientCert.enabled	false	Authentication with external module as: reverse proxy. Authenticated data are sent to application using 'Client-Cert' HTTP header. Do not enable this feature without a properly configured reverse-proxy.	false	BOOLEAN
smp.automation.authentication.external.tls.SSLClientCert.enabled	false	Authentication with external module as: reverse proxy. Authenticated certificate is sent to application using ‘SSLClientCert' HTTP header. Do not enable this feature without properly a configured reverse-proxy.	false	BOOLEAN
smp.sso.cas.ui.label	EU Login	The SSO service provider label.	true	STRING
smp.sso.cas.url	http://localhost:808	The SSO CAS URL endpoint.	true	URL

Parameter	Default Value	Comment	Restart needed	Value type
	0/cas/			
smp.sso.cas.urlPath.login	login	The CAS URL path for login. Complete URL is composed from parameters: ${smp.sso.cas.url}/${smp.sso.cas.urlpath.login}.	true	STRING
smp.sso.cas.callback.url	http://localhost:8080/smp/ui/public/rest/security/cas	The URL is the callback URL belonging to the local SMP Security System. If using RP, make sure it target SMP path '/ui/public/rest/security/cas'.	true	URL
smp.sso.cas.smp.urlPath	/smp/ui/public/rest/security/cas	SMP relative path which triggers CAS authentication.	true	STRING
smp.sso.cas.smp.user.data.urlPath	userdata/myAccount.cgi	Relative path for CAS user data. Complete URL is composed from parameters: ${smp.sso.cas.url}/${smp.sso.cas.smp.user.data.urlpath}.	true	STRING
smp.sso.cas.token.validation.urlPath	laxValidate	The CAS URL path for login. Complete URL is composed from parameters: ${smp.sso.cas.url}/${smp.sso.cas.token.validation.urlpath}.	true	STRING
smp.sso.cas.token.validation.params	acceptStrengths:BASIC,CLIENT_CERT|assuranceLevel:TOP	The CAS token validation key:value properties separated with '|'. Ex: 'acceptStrengths:BASIC,CLIENT_CERT|assuranceLevel:TOP'	true	MAP_STRING
smp.sso.cas.token.validation.groups	DIGIT_SMP|DIGIT_ADMIN	'|' separated CAS groups user must belong to.	true	LIST_STRING
mail.smtp.host		Email server - configuration for submitting the emails.	false	STRING
mail.smtp.port	25	Smtp mail port - configuration for submitting the emails.	false	INTEGER
mail.smtp.protocol	smtp	smtp mail protocol- configuration for submitting the emails.	false	STRING
mail.smtp.username		smtp mail protocol- username for submitting the emails.	false	STRING
mail.smtp.password		smtp mail protocol - encrypted password for submitting the	false	STRING

Parameter	Default Value	Comment	Restart needed	Value type
		emails.		
mail.smtp.properties		key:value properties separated with '|'. Ex: mail.smtp.auth:true|mail.smtp.starttls.enable:true|mail.smtp.quitwait:false.	false	MAP_STRING
smp.alert.user.login_failure.enabled	false	Enable/disable the login failure alert of the authentication module.	false	BOOLEAN
smp.alert.user.login_failure.level	LOW	Alert level for login failure.	false	STRING
smp.alert.user.login_failure.mail.subject	Login failure	Login failure mail subject. Values: {LOW, MEDIUM, HIGH}	false	STRING
smp.alert.user.suspended.enabled	true	Enable/disable the login suspended alert of the authentication module.	false	BOOLEAN
smp.alert.user.suspended.level	HIGH	Alert level for login suspended. Values: {LOW, MEDIUM, HIGH}	false	STRING
smp.alert.user.suspended.mail.subject	Login credentials suspended	Login suspended mail subject.	false	STRING
smp.alert.user.suspended.mail.moment	WHEN_BLOCKED	#When should the account disabled alert be triggered. Values: AT_LOGON: An alert will be triggered each time a user tries to login to a disabled account. WHEN_BLOCKED: An alert will be triggered once when the account got suspended.	false	STRING
smp.alert.password.imminent_expiration.enabled	true	Enable/disable the imminent password expiration alert.	false	BOOLEAN
smp.alert.password.imminent_expiration.delay_days	15	Number of days before expiration as for how long before expiration the system should send alerts.	false	INTEGER
smp.alert.password.imminent_expiration.frequency_days	5	Interval between alerts.	false	INTEGER

Parameter	Default Value	Comment	Restart needed	Value type
smp.alert.password.imminent_expiration.level	LOW	Password imminent expiration alert level. Values: {LOW, MEDIUM, HIGH}	false	STRING
smp.alert.password.imminent_expiration.mail.subject	Password imminent expiration	Password imminent expiration mail subject.	false	STRING
smp.alert.password.expired.enabled	true	Enable/disable the password expiration alert.	false	BOOLEAN
smp.alert.password.expired.delay_days	30	Number of days after expiration as for how long the system should send alerts.	false	INTEGER
smp.alert.password.expired.frequency_days	5	Frequency in days between alerts.	false	INTEGER
smp.alert.password.expired.level	LOW	Password expiration alert level. Values: {LOW, MEDIUM, HIGH}	false	STRING
smp.alert.password.expired.mail.subject	Password expired	Password expiration mail subject.	false	STRING
smp.alert.accessToken.imminent_expiration.enabled	true	Enable/disable the imminent accessToken expiration alert.	false	BOOLEAN
smp.alert.accessToken.imminent_expiration.delay_days	15	Number of days before expiration as for how long before expiration the system should send alerts.	false	INTEGER
smp.alert.accessToken.imminent_expiration.frequency_days	5	Frequency in days between alerts.	false	INTEGER
smp.alert.accessToken.imminent_expiration.level	LOW	AccessToken imminent expiration alert level. Values: {LOW, MEDIUM, HIGH}	false	STRING
smp.alert.accessToken.imminent_expiration.mail.subject	Access token imminent expiration	accessToken imminent expiration mail subject.	false	STRING
smp.alert.accessToken.expired.enabled	true	Enable/disable the accessToken expiration alert.	false	BOOLEAN
smp.alert.accessToken.expired.delay_day	30	Number of days after expiration as for how long the system	false	INTEGER

Parameter	Default Value	Comment	Restart needed	Value type
s		should send alerts.		
smp.alert.accessToken.expired.frequency_days	5	Frequency in days between alerts.	false	INTEGER
smp.alert.accessToken.expired.level	LOW	Access Token expiration alert level. Values: {LOW, MEDIUM, HIGH}	false	STRING
smp.alert.accessToken.expired.mail.subject	Access token expired	Password expiration mail subject.	false	STRING
smp.alert.certificate.imminent_expiration.enabled	true	Enable/disable the imminent certificate expiration alert.	false	BOOLEAN
smp.alert.certificate.imminent_expiration.delay_days	15	Number of days before expiration as for how long before expiration the system should send alerts.	false	INTEGER
smp.alert.certificate.imminent_expiration.frequency_days	5	Frequency in days between alerts.	false	INTEGER
smp.alert.certificate.imminent_expiration.level	LOW	certificate imminent expiration alert level. Values: {LOW, MEDIUM, HIGH}	false	STRING
smp.alert.certificate.imminent_expiration.mail.subject	Certificate imminent expiration	Certificate imminent expiration mail subject.	false	STRING
smp.alert.certificate.expired.enabled	true	Enable/disable the certificate expiration alert.	false	BOOLEAN
smp.alert.certificate.expired.delay_days	30	Number of days after expiration as for how long the system should send alerts.	false	INTEGER
smp.alert.certificate.expired.frequency_days	5	Frequency in days between alerts.	false	INTEGER
smp.alert.certificate.expired.level	LOW	Certificate expiration alert level. Values: {LOW, MEDIUM, HIGH}	false	STRING

Parameter	Default Value	Comment	Restart needed	Value type
smp.alert.certificate.expired.mail.subject	Certificate expired	Password expiration mail subject.	false	STRING
smp.alert.credentials.cronJobExpression	0 52 4 */1 * *	Property cron expression for triggering alert messages.	false	CRON_EXPRESSION
smp.alert.credentials.serverInstance	localhost	If smp.cluster.enabled is set to true then then instance (hostname) to generate report.	false	STRING
smp.alert.credentials.batch.size	200	Max alertes generated in a batch for the type.	false	INTEGER
smp.alert.mail.from	test@alert-send-mail.eu	Alert send mail.	false	EMAIL
authentication.blueCoat.enabled	false	Property was replaced by property: smp.automation.authentication.external.tls.clientCert.enabled	false	BOOLEAN
smp.domain.default		Default domain code. If the domain cannot be determined from the request, the default domain is used.	false	STRING
smp.certificate.validation.allowed.certificate.types		Allowed user certificate types. Empty value means no restrictions, for other values see the java KeyFactory Algorithms for examples: RSA|EC|Ed25519|Ed448",	false	LIST_STRING
authentication.blueCoat.enabled	false	Property was replaced by property: smp.automation.authentication.external.tls.clientCert.enabled	false	BOOLEAN
smp.domain.default		Default domain code. If the domain cannot be determined from the request, the default domain is used.	false	STRING
smp.certificate.validation.allowed.certificate.types		Allowed user certificate types. Empty value means no restrictions, for other values see the java KeyFactory Algorithms for examples: RSA|EC|Ed25519|Ed448",	false	LIST_STRING

15.3. smp_domain table configuration

This table is used to support the multitenancy feature of the SMP. Its parameters/fields are:

- **SML_SMP_ID**: This is the SMP ID that must match the SMP ID registered within the SML.
- **SML_CLIENT_CERT_HEADER:** The SMP's certificate - needed only when accessing BDMSL directly through HTTP. The configured "Client-Cert" HTTP header will be added to each BDMSL request (bypassing SSL certificate verification made normally by SSL terminator) .
- **SML_CLIENT_KEY_ALIAS**: This is the Domain scoped alias of the keystore private key used for authentication with the SML. The password is the same as xmldsig.keystore.password defined in the SMP configuration file.
- **SIGNATURE_KEY_ALIAS:** This field points to the **Domain scoped** alias of the Keystore private key certificate, used by the SMP to sign GET Signed Service Metadata responses.
- **SML_SUBDOMAIN:** This is the informative identifier of SML domain code (eHealth, Peppol, etc). Since SML subdomain is part of DNS domain it must be a valid DNS domain part.
- **DOMAIN_CODE:** The unique domain code that is used as HTTP domain parameter when adding participants true REST service API to particular a domain. Domain code can be alphanumeric and up to 63 characters long.

Example: Update the default single domain smp_domain table record:

update smp_domain **set SML_SMP_ID**='SMP-MCB-ID14', **SML_CLIENT_KEY_ALIAS**= 'smp_mock';

or

update smp_domain **set SML_SMP_ID**='SMP-MCB-ID14', **SML_CLIENT_CERT_HEADER**= 'serial=000000000000000000009A195D2DD88C&subject=CN=SMP_1000000000,O=DG-DIGIT,C=BE&validFrom=Oct 21 02:00:00 2014 CEST&validTo=Oct 21 01:59:59 2016 CEST&issuer=CN=Issuer Common Name,OU=Issuer Organization Unit,O=Issuer Organization,C=BE' where domainId='default';

16. SMP ADMIN CONSOLE

The SMP Admin console has two purposes:

- Enable anonymous users to search and explore published data in the SMP. Anonymous users can search for participants by participant ID, schema, or domain.

- Enable Service Group administrators to manage owned Service groups, SMP administrators to manage Service groups registered on SMP, and System Administrators to manage users and domains.

The administration dashboard is reachable via the following URLs:

http://[host]:[port]/smp[-version]/iu/

If the deployment package (war file) filename changed in order to simply upgrade the old SMP version as for example “smp-4.0.0.war” to “cipa-smp-full-webapp.war”, then the application root context might change as well.

Example:

http:// [host]:[port]/cipa-smp-full-webapp/ui/ .

Two types of application roles are defined in the SMP admin console:

- **System Administrator**: this is a “super admin” who can manage SMP users and domains

- **User**: a regular user of the DomiSMP: the user can administer Domains, Groups and Resources according to membership roles described in § 11-SMP User Management.

When users are logged, their role is displayed in read-only mode (as a label). Only the System Administrator can change the role of another user.

17. CONTACT INFORMATION

eDelivery Support Team

By email: **EC-EDELIVERY-SUPPORT@ec.europa.eu**

Support Service: 8am to 6pm (Normal EC working Days)