EUROPEAN COMMISSION

DIGIT
Connecting Europe Facility

# Service Metadata Publisher

# Administration Guide

# SMP 4.X

Version [3.2]

Status [Final]



Date: 11/04/2022

Document Approver(s):

| Approver Name | Role |
|---|---|
| Bogdan DUMITRIU | Project Manager |
| | |
| | |

Document Reviewers:

| Reviewer Name | Role |
|---|---|
| Joze Rihtarsic | Developer |
| Caroline AEBY | CEF 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<br>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<br>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 | Joze Rihtarsic | SMP 4.1 Updates |
| 2.4 | 15/11/2018 | Joze Rihtarsic<br>Chaouki BERRAH | Oracle Script Changes for Migration purposes |
| 2.5 | 28/11/2018 | Joze Rihtarsic<br>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<br>Gowtham VAITHIAM | Document Update |
| 2.8 | 08/10/2019 | Joze Rihtarsic | SMP 4.1.1 Release updates |
| 2.9 | 06/12/2019 | Caroline AEBY | Typo |
| 3.0 | 03/03/2020 | Chaouki BERRAH<br>Gowtham VAITHIAM | MySql Connector V8 and default keystore and trustore comments. |
| 3.1 | 23/03/2021 | Joze Rihtarsic | Tested on Oracle 19c |

| | | | |
|---|---|---|---|
| 3.2 | 11/04/2022 | Caroline AEBY | No more CEF |

# 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.X on either a WebLogic or Tomcat 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:**

**jdbc.driver = com.mysql.jdbc.Driver**

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

**jdbc.user =** smp_user

**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 now (JRE) 8 **only**: http://www.oracle.com/technetwork/java/javase/downloads/index.html
- **One** of the supported Database Management Systems:
  - MySQL (tested on version 5.7)
  - Oracle (tested on database version 11.2, XE edition and on database version 19c, SE2 edition)
- **One** of the supported Application Servers:
  - WebLogic 12.2 (WebLogic 12.1 is not supported from 4.1.1 release on)
  - Tomcat 8

## 3.1. Binaries repository

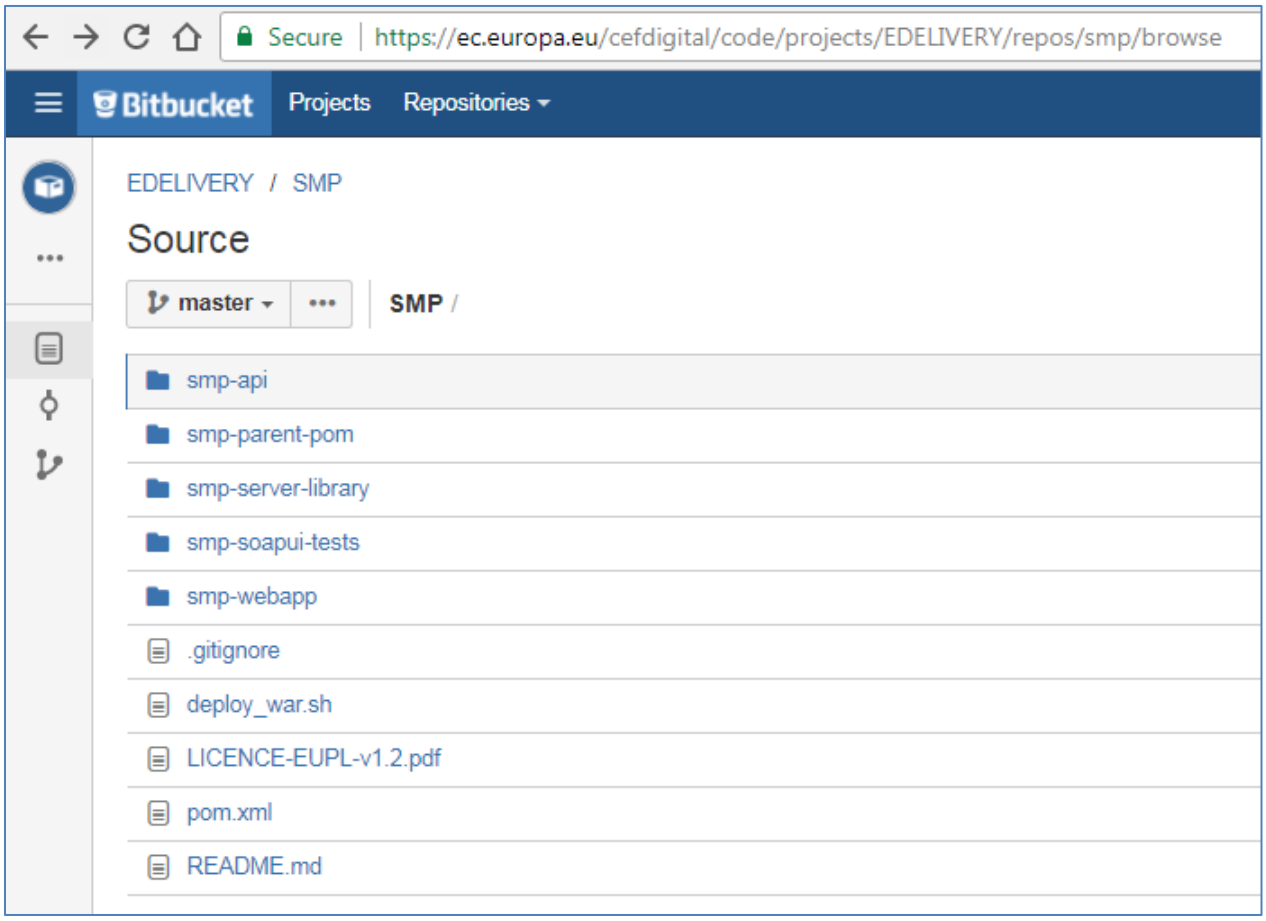
The SMP artefacts can be downloaded from the Digital site[1].

## 3.2. Source Code Repository

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

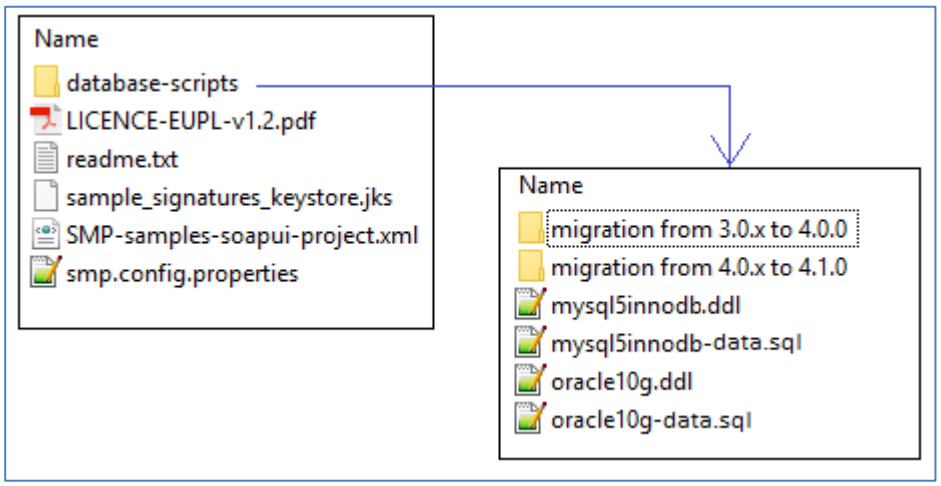
https://ec.europa.eu/digital-building-blocks/code/projects/EDELIVERY/repos/smp/browse

[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-4.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, **cef_edelivery_path**, refers to the name of the folder where the SMP package is installed.*
>
> *For Tomcat, it refers to **CATALINA_HOME**.*
>
> *For Oracle WebLogic, it refers to **DOMAIN_HOME**.*

# 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. Download and copy the mysql5innodb-4.x.ddl script to cef_edelivery_path/sql-scripts
2. Open a command prompt and navigate to the cef_edelivery_path/sql-scripts folder
3. 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 -proot_password smp_schema < mysql5innodb.ddl
```

Execute the following command to fill initial test data:

```
mysql -h localhost -u root_user -proot_password smp_schema < mysql5innodb-
data.sql
```

## 5.2. Oracle Database

1. Download and copy the oracle10g.ddl **script** to *cef_edelivery_path***/sql-scripts**
2. Navigate to *cef_edelivery_path***/sql-scripts** directory
3. 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

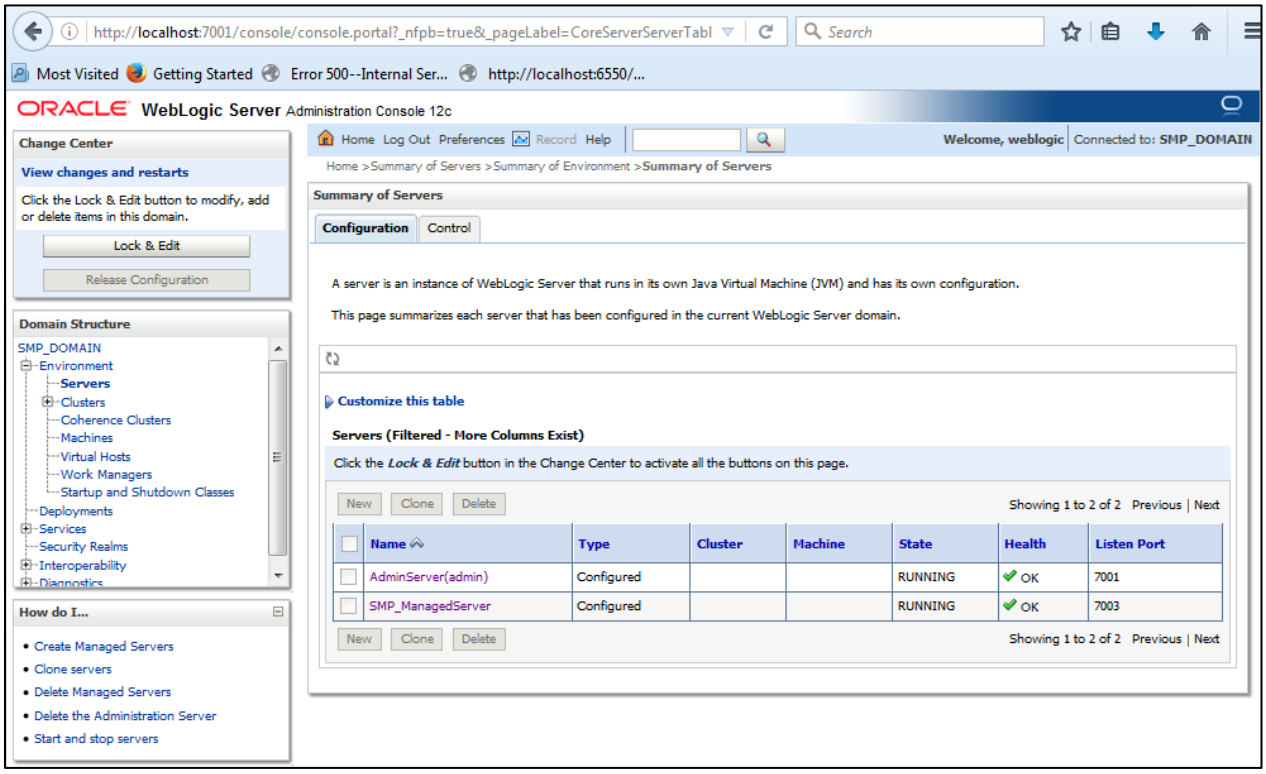
This section does not include the installation of a WebLogic application server. It is assumed that the WebLogic Server is installed and a WebLogic domain is created with an administration server and a managed server on which the SMP will be deployed.

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:




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

- Disabling the 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

The eDelivery SMP has its own authentication mechanism that makes the WebLogic authentication redundant. 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

Under the DOMAIN_HOME directory, create the following sub-directories:

- smp
- logs

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

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

## 7.1. Configuring the Extra CLASSPATH for Tomcat

In this Tomcat example, a directory called **smp** will be created in the root path of the Tomcat installation (**CATALINA_HOME**) and the **CLASSPATH** modified to include this new directory using an existing Tomcat batch file (CATALINA_HOME/bin/setenv.[sh|bat]).

Create a **smp** directory in the **CATALINA_HOME** directory.

**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 : http://www.oracle.com/technetwork/apps-tech/jdbc-112010-090769.html
- For Mysql : https://www.mysql.com/products/connector/

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

# 8. SMP CONFIGURATION

Since **SMP 4.1.0.FR**, configuration properties (except database connection configuration) are stored in the Database **SMP_CONFIGURATION** table. In order to make the migration easier for existing deployments, the initial database configuration is imported from the existing smp configuration file. Therefore, for a new installation, the initial procedure stays the same.

For this step, use the following resources delivered within the zip file downloaded in section §3.3:

- An smp config file named **smp.config.properties** located in the CLASSPATH

The **smp.config.properties** file must be copied to the CLASSPATH folder configured in chapter (see $7.1).

## 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 configured on the application server

```
../
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 configured on the application server

```
../
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

Note that from SMP 4.1.0. FR on, the eDelivery SMP application uses only one keystore with the following default name: smp-keystore.jks. Keystore is automatically created by eDelivery SMP 4.1.0.FR. Initialization is triggered at start-up of the eDelivery SMP when the database *SMP_CONFIGURATION* table is empty. Managing keystore (adding/deleting entries) on SMP 4.1.0. FR is done by the eDelivery SMP Admin console on the Domain page with the SYSTEM_ADMIN role (see also §17).

If there is no keystore and truststore configured in the SMP for the initial deployment, SMP will create a dummy keystore and truststore in the database.User can later change the keystore and truststore via SMP GUI.

### 8.2.1. *Migration of SMP keystore*

Before SMP 4.1.0.FR, the eDelivery SMP used two distinct keystores for different purposes:

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

For migration purposes, please follow the description of deprecated/obsolete keystores described in the next sections.

### 8.2.2. *XMLDSIG response signing Keystore and SML Keystore (DEPRECATED since 4.1.0.FR)*

When initializing SMP 4.1.0.FR with deprecated attributes: **xmldsig.keystore.classpath, xmldsig.keystore.password, bdmsl.integration.keystore.path, bdmsl.integration.keystore.password** in smp.config.properties, keys and certificates are automatically imported to new smp-keystore.jks.

### 8.2.3. *Initializing and merging new keystore.*

The initial deployment of the eDelivery SMP 4.1,0 FR version automatically creates a new keystore with the following filename:

- ***smp-keystore.jks***.

The keystore is created in the folder defined by property: configuration.dir. If property is not defined then is defined as parent folder of the **xmldsig.keystore.classpath or bdmsl.integration.keystore.path**. If even this two parameters are not given then keystore is created in workdir of the application server.

If init attribute encryption.key.filename is not given or file not exists in configuration folder then a random symmetric encryption key (AES 256) is also generated and stored configuration folder with a file name:

- ***encryptionPrivateKey.private***

The Symetric key is used to encrypt the keystore password, which is then stored in the corresponding database field property:

- **smp.keystore.password** ( example value: 7css/uqZRFBeiKavrzzB1A==)

The Decrypted password is also stored in the database. The Decrypted value is not used by the application. You need to store the password in a safe location and remove it from the database.

- **smp.keystore.password.decrypted**

The Application will import the keys and certificate from the existing keystores. First from **xmldsig.keystore.classpath** and then, if it exists, from **bdmsl.integration.keystore.path**, keeping the same aliases for keys used in the old keystore. If the aliases are the same in both keystores, then the aliases from sml keystore are appended with a 3-digit number (starting at 001).

Example:

If the duplicate alias is *myBestCertificate*, then the second imported certificate will have alias: *myBestCertificate_001*.

If the migration fails, the migration/initialization can be relaunched after clearing the ***SMP_CONFIGURATION*** table.

Example:

```
SQL> DELETE FROM SMP_CONFIGURATION;
```

After a successful migration, the old keystores can be backed up and then removed from the property file.

## 8.3. SMP Truststore

From SMP 4.1.1. on, the eDelivery SMP application uses truststore for User Certificate verification. The truststore is automatically created when application starts.
Because of back-compatibility: If truststore is empty then trustiness of certificate is not verified.

Managing keystore (adding/deleting entries) on SMP 4.1.1. FR is done by the eDelivery SMP Admin console on the User page with the SYSTEM_ADMIN role.

If there is no keystore and truststore configured in the SMP for the initial deployment, SMP will create a dummy keystore and truststore in the database.User can later change the keystore and truststore via SMP GUI.

## 8.4. Changing SMP Truststore and Keystore passwords in database

If we change keyststore or truststore filename – than we have to change also the password in database. Because password in database is encrypted we must put clear text password in format:

- {DEC}{password}

Password in second curved brackets is automatically encrypted by the SMP when properties are reloaded.

# 9. SMP .WAR FILE DEPLOYMENT

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

### 9.1.1. *Tomcat*

Copy the smp.war file to the Tomcat **webapps** directory.

### 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:


Important: Context path (example above: */smp-4.1.0*) should be the same as is deployment WAR file. If the war file is called ***smp**-4.1.0**..war*** then the URL will be **http://*[hostname]*:*[port]*/smp**-4.1.0.***

# 10. CONFIGURING THE EDELIVERY SMP FOR USE WITH AN SML

The eDelivery SMP can be registered in an SML, using two identification mechanisms:

- Using HTTP and plain text with metadata embedded into the header of the REST request,
- Using HTTPS/TLS and a keystore containing a certificate.

## 10.1. Configuring the BDMSL Integration

The first step is to configure the SMP so that it can be used with an SML.

This is achieved by setting the database property: **bdmsl.integration.enabled** parameter to **true**, default is **false**.

```
Execute the following sql command for Oracle database:UPDATE
SMP_CONFIGURATION SET VALUE='true', LAST_UPDATED_ON=systemdate WHERE
PROPERTY='bdmsl.integration.enabled';
```

## 10.2. Configuring the SML URL

The configuration of the SML URL endpoint is achieved by configuring the **bdmsl.integration.url** property in the **smp.config.property** file as follows:

```
UPDATE SMP_CONFIGURATION SET VALUE= http://localhost:8080/edelivery-sml/,
LAST_UPDATED_ON=systemdate WHERE PROPERTY= bdmsl.integration.url';
```

## 10.3. SMP authentication to an SML

Once registered in an SML, the SMP needs to authenticate against the SML during normal operations. For authentication, the client key/certificate must be installed in the keystore and an alias defined for the domain. The configuration is either done using the eDelivery SMP Admin console (see also §17) or by entering the values directly into the database.

SML supports two ways of authentication

- BlueCoat: HTTP client certificate header. This is used behind a bluecoat reverse proxy.
- HTTPS/TLS: Standard mutual TLS authentication.

### 10.3.1. *Plain Text HTTP*

**SML_CLIENT_CERT_HEADER** contains the SMP certificate needed if accessing SML directly through HTTP. The configured "Client-Cert" HTTP header will be added to each SML request bypassing SSL certificate verification made normally by SSL terminators.

**SML_CLIENT_CERT_HEADER** is the configuration attribute (column) in the **SMP_DOMAIN** table described in section §16.3.

### 10.3.2. *HTTPS/TLS*

When using HTTPS/TLS, as mentioned in the **bdmsl.integration.url** covered earlier, all information related to the Keystore containing the SMP certificate must be configured under the SMP Keystore and the corresponding certificate alias must be set in column **SML_CLIENT_KEY_ALIAS** in the **SMP_DOMAIN** table described in section §16.3.

# 11. SMP User Management

Only **Admin SMP** and Admin **ServiceGroup** users, who connect to the eDelivery SMP, need to be created in the SMP database.

Anonymous users or public users can access the SMP to retrieve only. They are not registered and therefore not added to the database.

There are no restrictions on the number of users that can be created to access the eDelivery SMP.

The database script creates the following users:

| User name | Role | Default password [2] |
|---|---|---|
| system | SYSTEM_ADMIN | 123456 |
| smp | SMP_ADMIN | 123456 |
| user | SERVICE_GROUP_ADMIN | 123456 |

## 11.1. User Roles

The eDeliery SMP users can be of three types, as briefly described below:

| Actor | UC | Short description | Oper. | Data |
|---|---|---|---|---|
| Admin SMP | Create or Update 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. Information is stored into ServiceGroup table. This same service is used to create and update a ServiceGroup. | PUT | ServiceGroup |
| Admin SMP | Erase Service Group | Erases the service group definition AND the list of services for the specified receiver participant. | DELETE | ServiceGroup |
| Admin Service Group | Create or Update Service | Publish detailed information about one specific document service (multiple processes and endpoints). This same service is used to create and | PUT | ServiceMetadata |

[2] to change immediately for security reasons

| Actor | UC | Short description | Oper. | Data |
|---|---|---|---|---|
| | Metadata | update ServiceMetaData. | | |
| Admin Service Group | Erase Service Metadata | Remove all information about one specific service (i.e. all related processes and endpoints definitions). | DELETE | ServiceMetadata |
| User | Retrieve Service Group | Obtain the list of services provided by a specific receiver participant (collection of references to the ServiceMetaData's). This service provides the information related to the Service Group according to the input duplet participantIdentifier + participantIndentifierScheme. Returns information from the ServiceMetadata table only (references to actual MetaData). | GET | ServiceGroup |
| 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. Returns information from the Endpoint table. | GET | SignedServiceMetadata |
| System admin | | Create, modify and delete users and domains. System admin can be only used in the edelivery SMP 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/cefdigital/wiki/display/CEFDIGITAL/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.

## 11.2. BCRYPT password generation

Since SMP 4.1.x users can be managed by SMP Admin console. Following procedure can be used for creating first system user.

The SMP v4.X 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-4.X.war** file into a temporary directory of your choice.

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

```
jar -xvf smp-4.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
```

## 11.3. 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 ROLE column as follows:

| User Role | Role value |
|---|---|
| Admin SMP | SMP_ADMIN |
| Admin Service Group | SERVICE_GROUP_ADMIN |
| System Administrator | SYSTEM_ADMIN |
| AnonymousUser<br>(Not defined in the SMP User database) | N/A |

In the following two examples, an **Admin SMP** and an **Admin ServiceGroup** users are created.

### 11.3.1. *SYSTEM_ADMIN SMP User creation*

***Remark:***

- In order to logon on the Administration Console **for the first time**, it is necessary to, first 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 (ID, USERNAME, PASSWORD, ROLE, ACTIVE,
CREATED_ON,LAST_UPDATED_ON,PASSWORD_CHANGED) values ((select max(next_val) from
SMP_USER_SEQ), 'smp_admin',
'$2a$10$olcGeWKGEoRia2DPuFqRNeca0IEdRSmOrljLz57BAjf1jlC9SohrS',
'SYSTEM_ADMIN',1,SYSDATE(), SYSDATE(),SYSDATE() );

update SMP_USER_SEQ set next_val=next_val+1;
```

Oracle example:

```
SQL> INSERT into SMP_USER (ID, USERNAME, PASSWORD, ROLE, ACTIVE,
CREATED_ON,LAST_UPDATED_ON) values (SMP_USER_SEQ.NEXTVAL, 'smp_user',
'$2a$10$olcGeWKGEoRia2DPuFqRNeca0IEdRSmOrljLz57BAjf1jlC9SohrS',
'SYSTEM_ADMIN',1,SYSDATE, SYSDATE );

update SMP_USER_SEQ set next_val=next_val+1;
```

### 11.3.2. *Admin SMP User Creation*

Username : smp_user1

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

IsAdmin : SERVICE_GROUP_ADMIN

Example: Oracle database command.

```
SQL> insert into smp_user (ID, username, password, ROLE, ACTIVE,
CREATED_ON,LAST_UPDATED_ON) values (SMP_USER_SEQ.NEXTVAL, 'smp_user1',
'$2a$10$6nytsush2bqfboliycxn8euvibcnn.wcjurwotjlmndodati85zma',
'SMP_ADMIN',1,SYSDATE, SYSDATE);
```

### 11.3.3. *Admin ServiceGroup User Creation*

Username : smp_user2

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

IsAdmin : SERVICE_GROUP_ADMIN

Example: Oracle database command.

```
SQL> insert into smp_user (ID, username, password, ROLE, ACTIVE,
CREATED_ON,LAST_UPDATED_ON) values (SMP_USER_SEQ.NEXTVAL, 'smp_user1',
'$2a$10$6nytsush2bqfboliycxn8euvibcnn.wcjurwotjlmndodati85zma',
'SERVICE_GROUP_ADMIN',1,SYSDATE, SYSDATE);
```

# 12. LOGGING CONFIGURATION

## 12.1. Logging properties

The default SMP logging properties are defined in the ***./WEBINF*/smp-log4j.properties file** embedded in the smp **.war** file.

For changing the log folder we can set a new value to the " log.folder" property in smp.config.propertis.

It is also possible to modify the configuration of the logs by creating custom property log4j file and configuring property: log.configuration.file in the smp.config.propertis.

# 13. SOAPUI TESTING

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

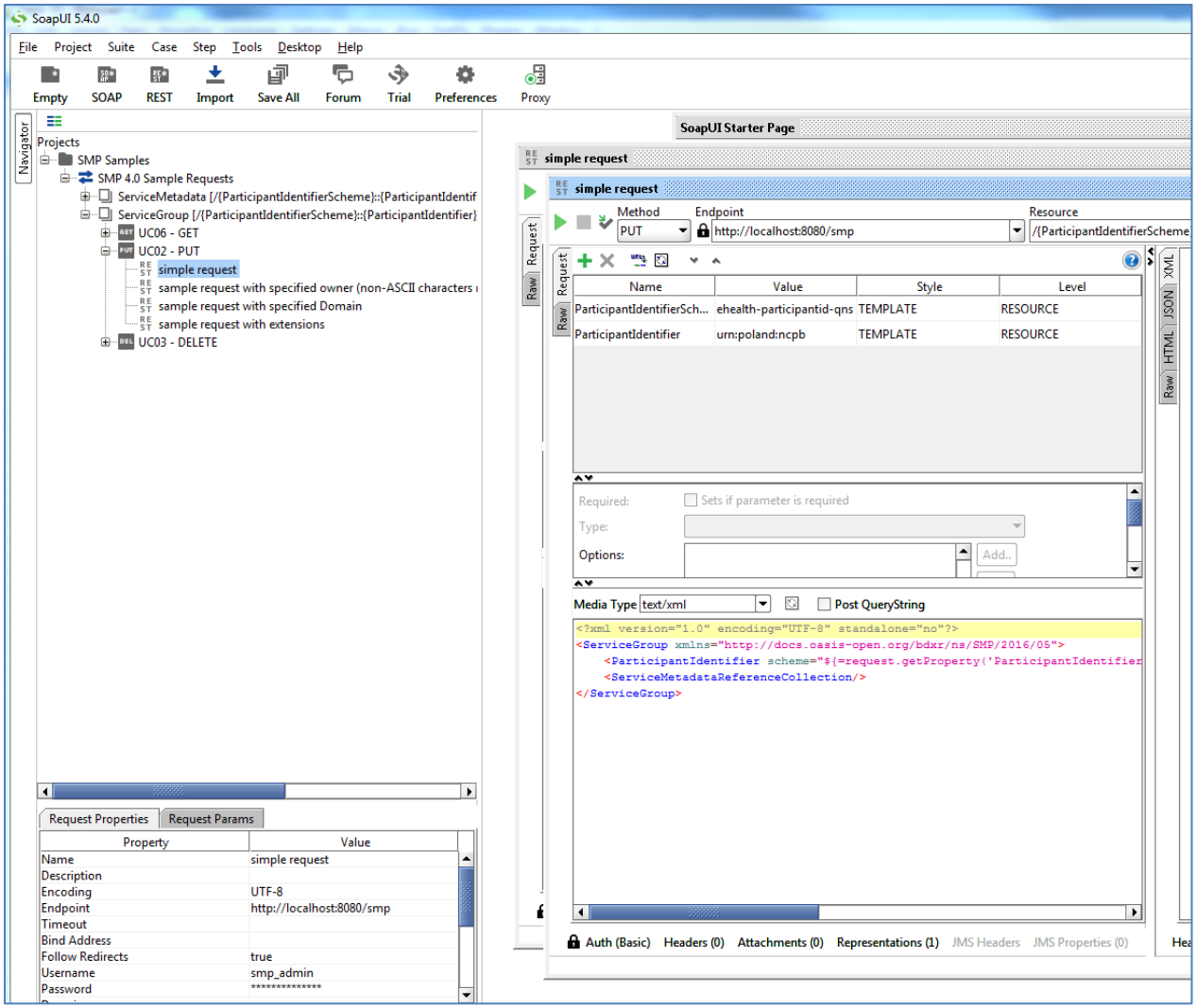
An SMP SoapUI project contains sample requests and is included in the zip file already downloaded.

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

## 13.1. Creation, update and deletion of Service Groups.

### 13.1.1. *Create a Service Group*

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

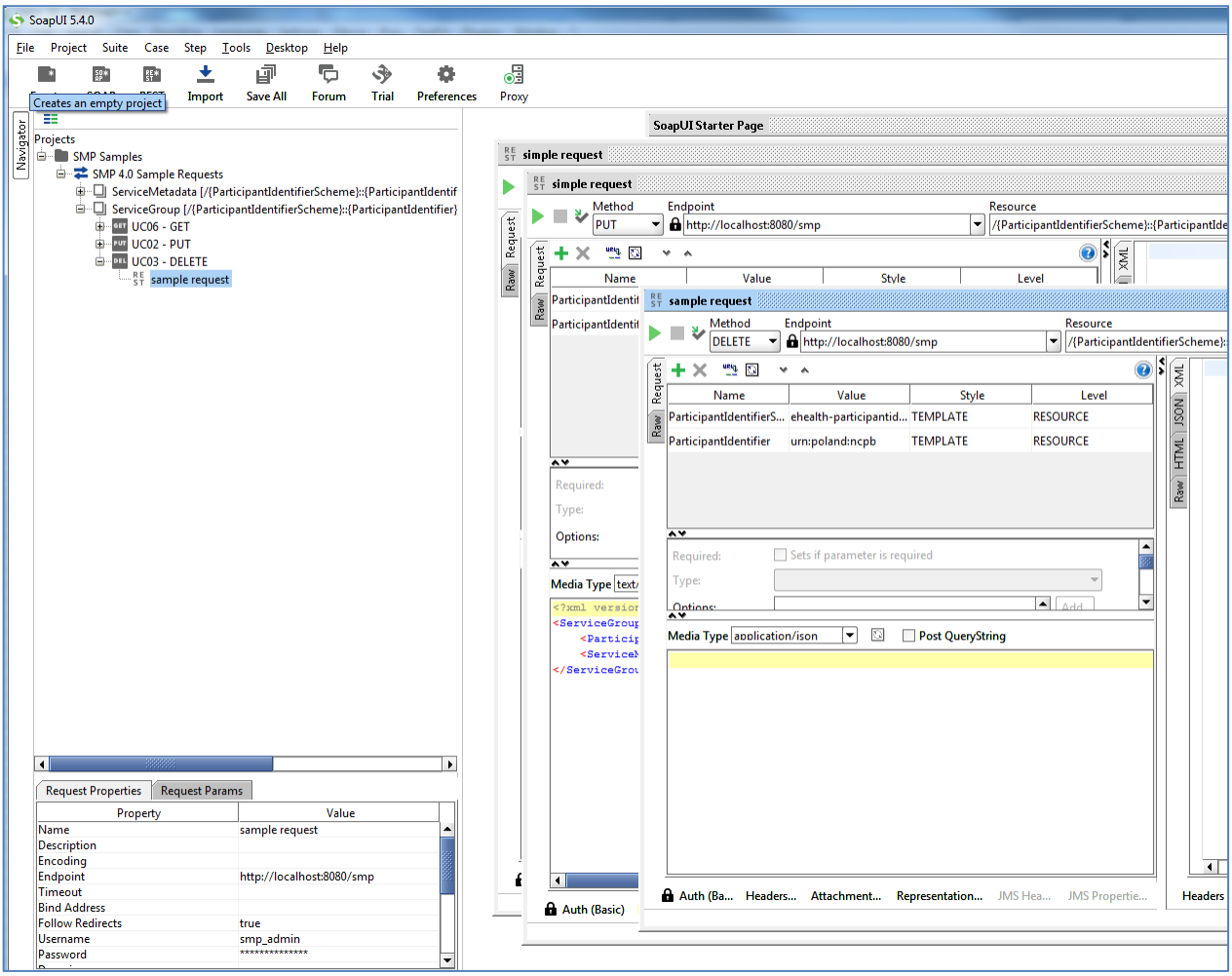



### 13.1.2. *Update a Service Group*

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

### 13.1.3. *Delete a ServiceGroup*

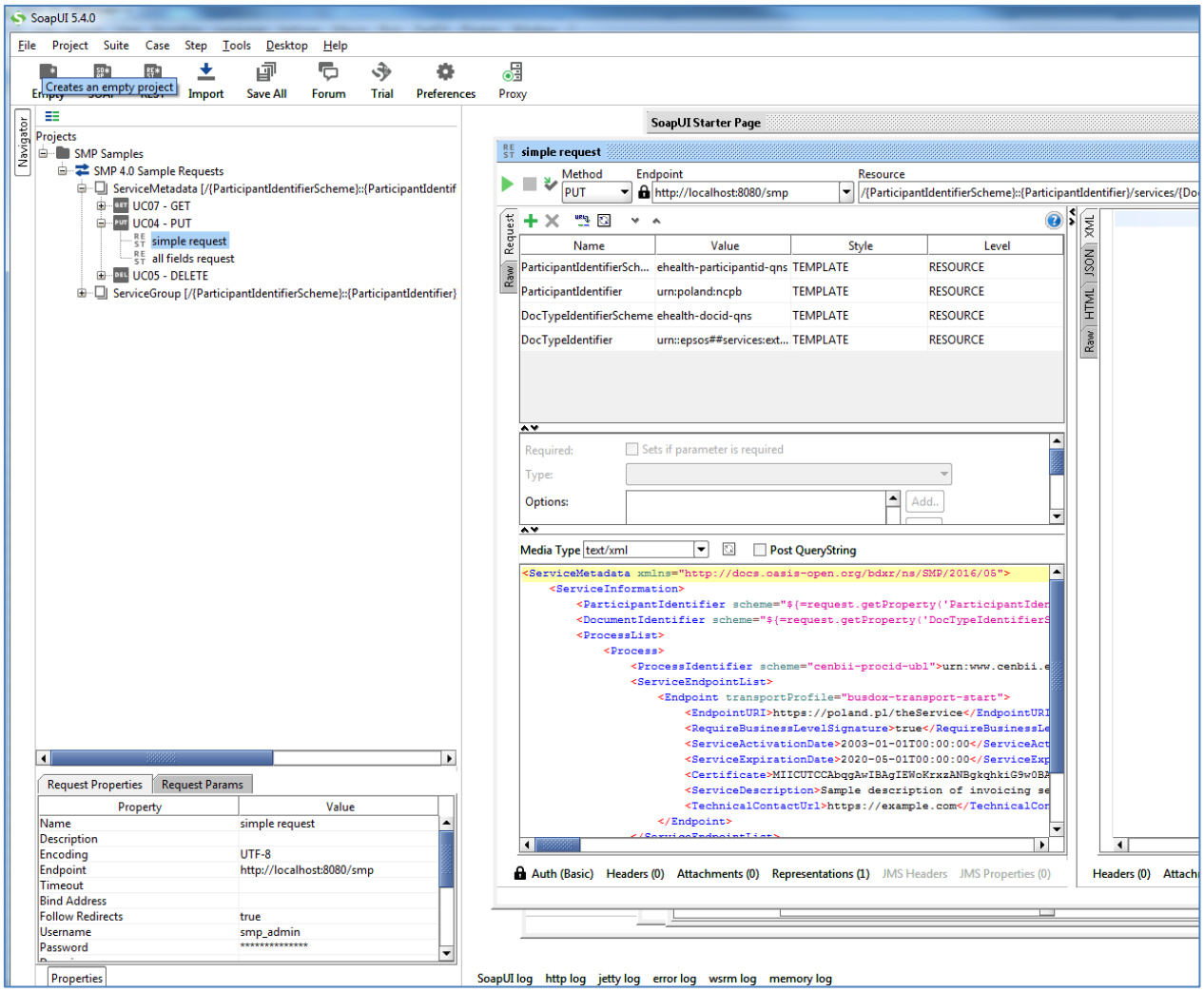
On the SoapUI interface on the left navigation panel, browse to the REST DELETE method as indicated below:




## 13.2. Creation, update and deletion of Service Metadata.

### 13.2.1. *Create a Service Metadata*

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

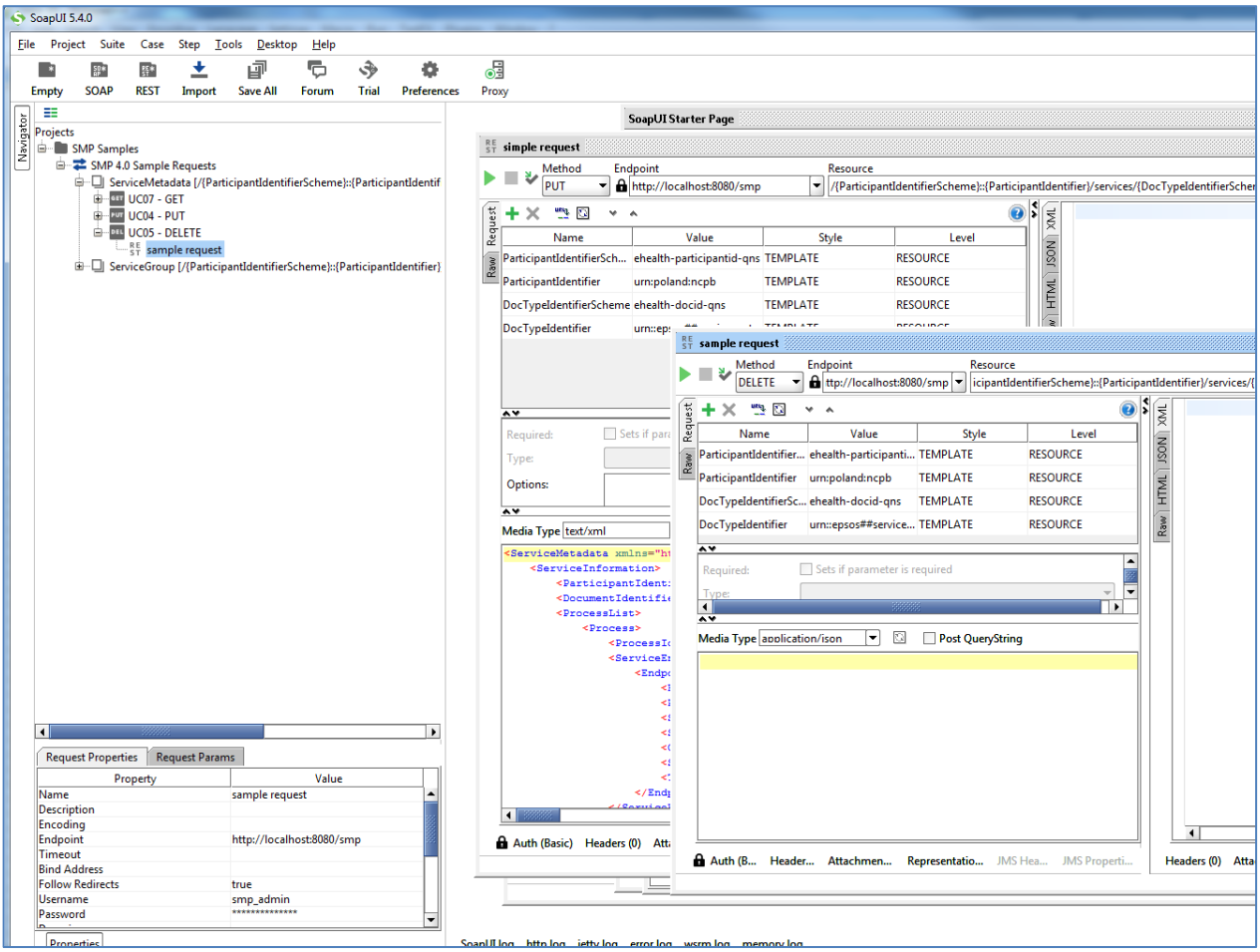



## 13.2.2. *Update Service Metadata*

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

### 13.2.3. *Delete Service Metadata*

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



# 14. THE SWAGGERUI INTERFACE

## 14.1. Introduction

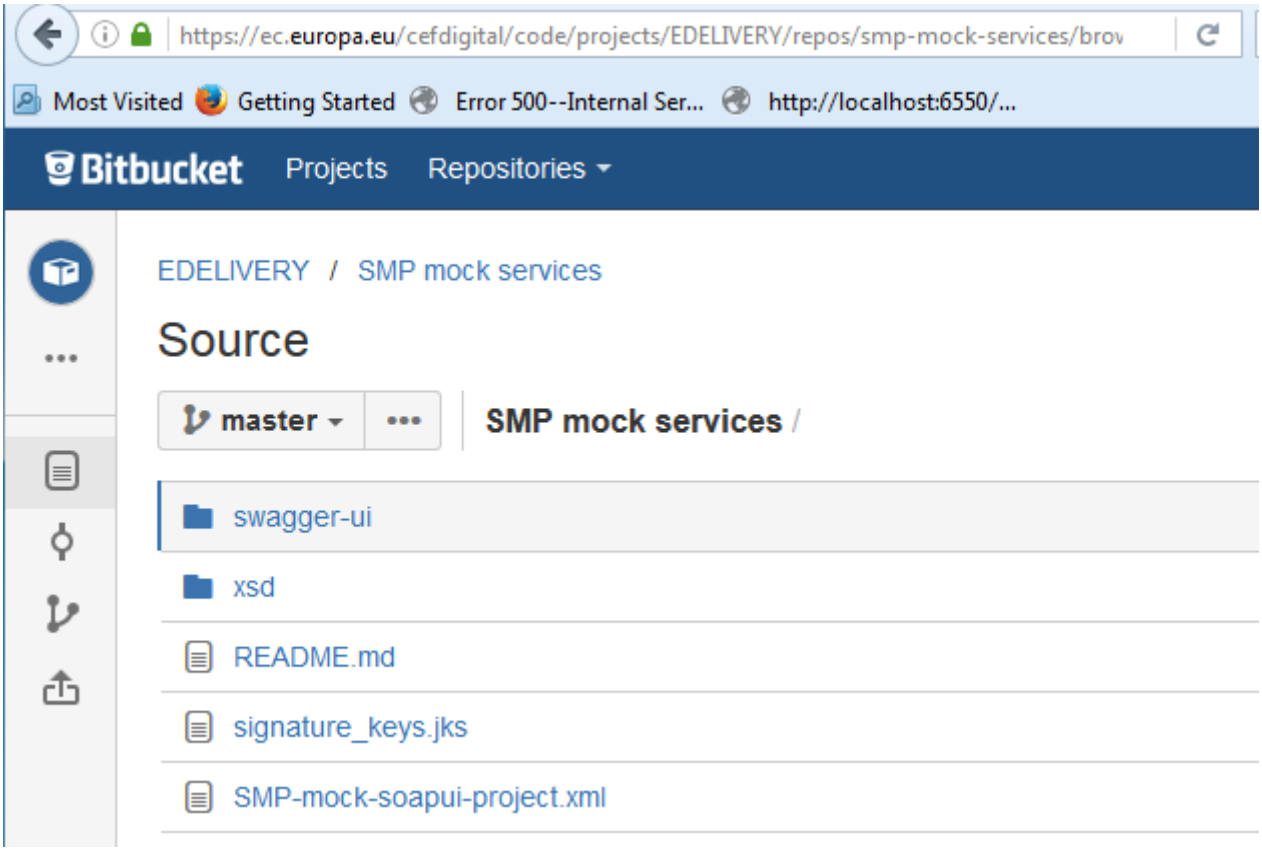
"***Swagger is an API developer tools for the OpenAPI Specification (OAS). It allows anyone (developers or end-users) to interact with the API's resources***"[3].

The SMP Web Client can be tested at: http://130.206.118.4/smp-swagger-ui and, as explained, is a WEB client configured to shoot (PUT, GET or DELETE) at the mocked SMP implementation Metadata.

## 14.2. 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:

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

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

## 14.3. 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-4.X](http://localhost:7003/ smp-4.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 4.X WEB client (based on Swagger-UI)"
  },
  "host": "localhost:7003",
  "basePath": "/ smp-4.X",
  "externalDocs": {
  },
```

## 14.4. 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/*
```

## 14.5. Deploy the SMP SwaggerUI war file

### 14.5.1. *On Tomcat*

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

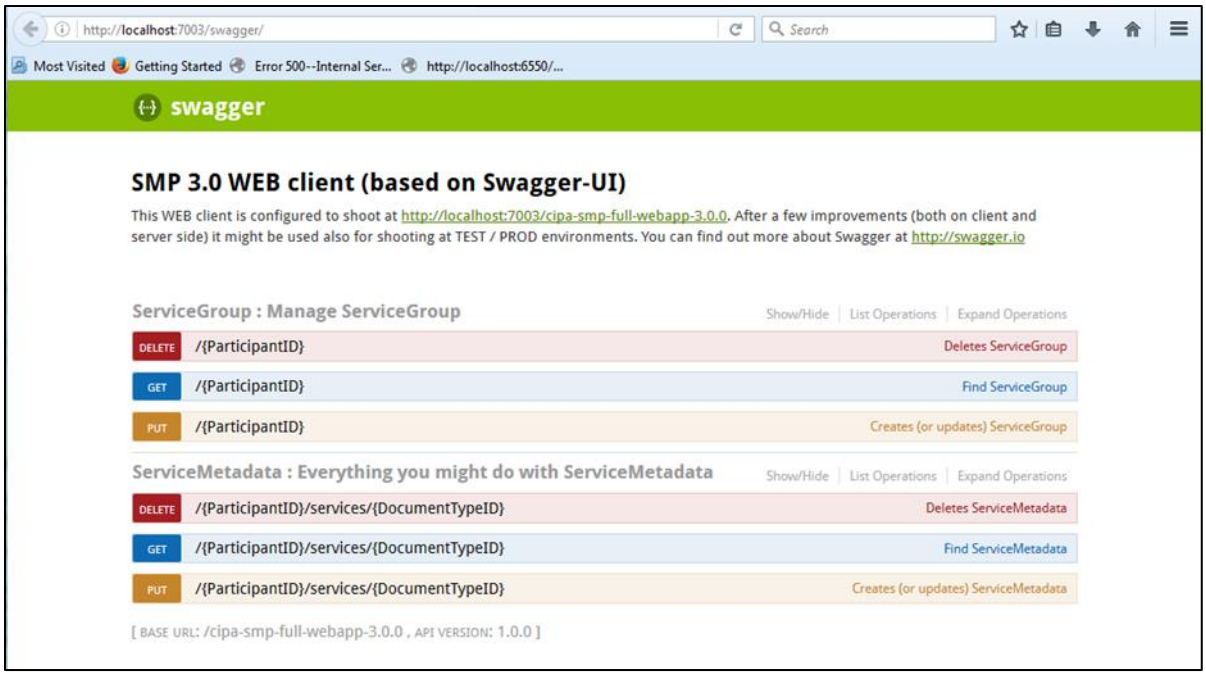
### 14.5.2. *On WebLogic:*

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:



# 15. SMP COMPILATION

## 15.1. Compilation prerequisites

### 15.1.1. *Supported Operating System Platform*

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

- Windows Workstation & Server
- Linux platform

### 15.1.2. *Software Requirements*

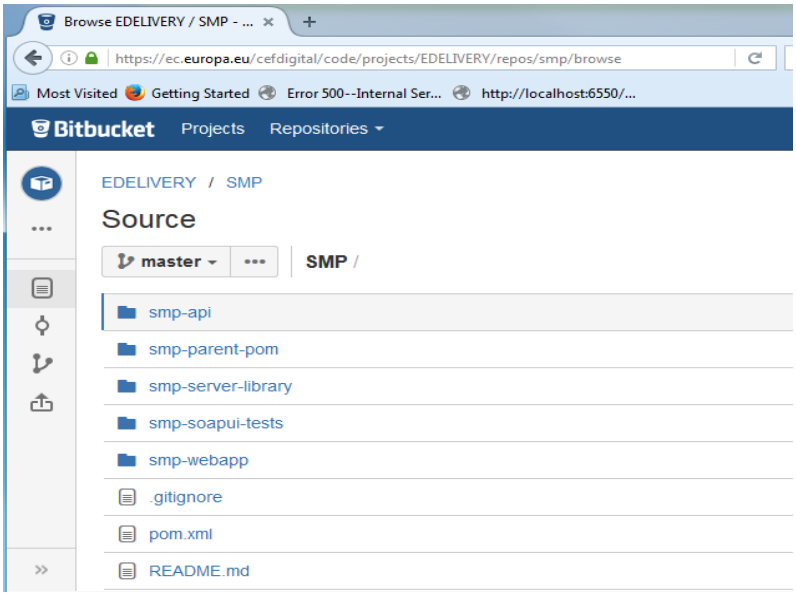
The following software components on the target system:

- Java Development Kit environment (JDK), version 7 or 8: http://www.oracle.com/technetwork/java/javase/downloads/index.html
- Maven 3.0 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).

## 15.2. 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



## 15.3. 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/cefdigital/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-parent-pom ..................................... SUCCESS [  0.120 s]
[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-4.X  smp.war classes  generated-sources  generated-test-sources  maven-status  test-classes  webapp-classes
```

# 16. SMP CONFIGURATION FILE AND TABLE

## 16.1. 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 is able to 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.*

## 16.2. The smp.config.properties file

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

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

```
#
# Copyright 2017 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.
#
```

```
# *********************************
# Datasource JNDI configuration
# *********************************
# weblogic datasource JNDI example
# sml.datasource.jndi=jdbc/cipaeDeliveryDs
# tomcat datasource JNDI example
sml.datasource.jndi=java:comp/env/jdbc/edelivery
hibernate.dialect =org.hibernate.dialect.MySQL5InnoDBDialect


# Datasource can be configured also in the property file
# jdbc.driver = com.mysql.jdbc.Driver
# jdbc.url =
jdbc:mysql://localhost:3306/smpdbdtest?useSSL=false&characterEncoding=UTF-
8&useUnicode=true
# jdbc.user=smptest
# jdbc.password=


# *********************************
# Datasource JNDI configuration
# *********************************
log.folder=../smp-logs/
# Define custom logging properties
# log.configuration.file=/cef/test/smp/smp-log4j.properties
```

### 16.2.1. *Detailed SMP configuration properties*

Attributes are stored in the database and the filesystem property file. If aproperty is located on both locations then the database property value takes precedence.

The following table describes them briefly:

| Parameter | Default Value | Location | Comment |
|---|---|---|---|
| authentication.blueCoat.enabled | false | Database/Property file | Authentication with Blue Coat means that all HTTP requests having **Client-Cert** header will be authenticated as username placed in the header.<br><br># Never expose an SMP to the WEB without properly configured reverse-proxy and active blue coat. |
| contextPath.output | true | Database/Property file | PRODUCTION mode. This variable is used to clear the context path of the SMP |
| encodedSlashesAllowedInUrl | false | Database/Property file | Most Java libraries and J2EE containers block encoded slashes in URL, for security reasons.<br><br>## Theoretically there are no restrictions on slash "/" characters in document or participant identifiers,<br><br>## but by default we block then as well.<br><br>## If slash "/" characters must be supported, than switch this property to "true".<br><br>## Remember that in such case the relevant change should be also applied on J2EE level<br><br>## I.e. for Tomcat it is handled by property: org.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH=true |

| Parameter | Default Value | Location | Comment |
|---|---|---|---|
| bdmsl.integration.enabled | false | Database/Property file | BDMSL (SML) integration ON/OFF switch |
| identifiersBehaviour.ParticipantIdentifierScheme.validationRegex | ^((?!^.{26})([a-z0-9]+-[a-z0-9]+-[a-z0-9]+)\|urn:oasis:names:tc:ebcore:partyid-type:(iso6523\|unregistered)(:.+)?$) | Database/Property file | Participant Identifier Schema of each PUT ServiceGroup request is validated against this schema.<br>SMP has by default implemented validation for PEPPOL scheme type and ebCoreParticipant identifier scheme type. The Peppol scheme 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].<br>The ebCore Participant identifier scheme type must starts with:<br>urn:oasis:names:tc:ebcore:partyid-type:iso6523:<br>or<br>urn:oasis:names:tc:ebcore:partyid- unregistered:<br>This only two scheme types are supported because of current SML implementation. BDMSL 4.0.1 also only supports these two types. When there is no integration with SML scheme type validation can be changed. |
| 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]- | Database/Property file | Error message for UI |

| Parameter | Default Value | Location | Comment |
|---|---|---|---|
| | [identifierType] (ex.: 'busdox-actorid-upis') and may only contain the following characters: [a-z0-9]. | | |
| bdmsl.integration.enabled | false | Database/Property file | BDMSL (SML) integration ON/OFF switch |
| identifiersBehaviour.caseSensitive.ParticipantIdentifierSchemes | casesensitive-participant-scheme1\|casesensitive-participant-scheme2 | Database/Property file | ## All Identifiers by default are CASE-INSENSITIVE.<br>## Specifies schemes of participant/document identifiers that must be considered CASE-SENSITIVE.<br>## List values (delimited by pipe character: "\|" ) placed here are checked against runtime (request) schemes in the CASE-INSENSITIVE way |
| identifiersBehaviour.caseSensitive.DocumentIdentifierSchemes | casesensitive-doc-scheme1\|casesensitive-doc-scheme2 | Database/Property file | |
| bdmsl.integration.url | #https://sml.peppolcentral.org/manageparticipantidentifier<br>http://localhost:8080/manageparticipantidentifier | Database/Property file | The URL of the targeted SML (incl. the service name) |
| bdmsl.integration.logical.address | http://localhost:8080/smp/ | Database/Property file | SMP logical address. Value is used when registering new domain to SML. |
| bdmsl.integration.physical.address | 0.0.0.0 | Database/Property file | SMP physical address. Value is used when registering new domain to SML. |
| bdmsl.integration.tls.disableCNCheck | false | Database/Property | If SML Url is HTTPs - Disable CN check if needed |

| Parameter | Default Value | Location | Comment |
|---|---|---|---|
| | | file | |
| bdmsl.integration.tls.serverSubjectRegex | .* | Database/Property file | Regular expression for server TLS certificate subject verification CertEx. .*CN=acc.edelivery.tech.ec.europa.eu.*. |
| smp.keystore.password | Generated | Database/Property file | Encrypted password for keystore or use format {DEC}{[password}}. Password will be automatically encrypted and properties update.. |
| smp.keystore.password.decrypted | Generated | Database/Property file | Generated password for keystore. Password should be stored on save located and then the entry should be deleted. Purpose of decrypted password is for system administrator to open/edit keystore with other tools if needed. |
| smp.keystore.filename | smp-keystore.jks | Database/Property file | Keystore for SMP. |
| smp.truststore.password | | Database/Property file | Encrypted truststore password or use format {DEC}{[password}}. Password will be automatically encrypted and properties update. |
| smp.truststore.password.decrypted | | Database/Property file | Only for backup purposes when password is automatically created. Store password somewhere save and delete this entry! |

| Parameter | Default Value | Location | Comment |
|---|---|---|---|
| smp.truststore.filename | smp-truststore.jks | Database/Property file | Truststore for SMP |
| smp.certificate.crl.force | false | Database/Property file | If false then if CRL is not reachable ignore CRL validation. |
| configuration.dir | Parent of (xmldsig.keystore.classpath) | Database/Property file | Configuration folder |
| encryption.key.filename | Generated | Database/Property file | Symetric encryption key for keystore password encryption. Password is stored in database (smp.keystore.password) and encryption key is stored on server's file system. |
| smp.property.refresh.cronJobExpression | 0 48 */1 * * * | Database/Property file | Property refresh cron expression (def 12 minutes to each hour). Properties are refreshed if any "update date" in SMP_CONFIGURATION is newer the last load of properties.<br>When changing this property server must be restarted. |
| bdmsl.integration.keystore.path (Deprecated) | | Database/Property file | The location of the keystore. At initialization database propertis keystore is imported to smp-keystore.jks |
| bdmsl.integration.keystore.password(Deprecated) | | Database/Property file | The password of the keystore |
| xmldsig.keystore.classpath (Deprecated) | ../keystore/keystore.jks | Database/Property | The location of the xmldsig keystore (Deprecated) |

| Parameter | Default Value | Location | Comment |
|---|---|---|---|
| | | file | |
| xmldsig.keystore.password(Deprecated) | peppol | Database/Property file | The password of the xmldsig keystore |
| jdbc.driver | com.mysql.jdbc.Driver | Property file | Database Configuration: Driver<br>MySQL:<br>com.mysql.jdbc.Driver<br>Oracle Database:<br>oracle.jdbc.OracleDriver |
| jdbc.url | jdbc:mysql://localhost:3306/smp | Property file | Database Configuration: url<br>MySQL :<br>jdbc:mysql://dbhost:dbport/smp_database<br>Oracle Database:<br>jdbc:oracle:thin:@dbhost:dbport:smp_database<br>jdbc:oracle:thin:@dbhost:dbport/smp_service |
| jdbc.user | | Property file | Database User/Password Configuration: User |
| jdbc.password | | Property file | Database User/password Configuration: Password |
| hibernate.dialect | | Property file | Hibernate dialect for database. |
| datasource.jndi | | Property file | JNDI name for accesing datasource configured on application server. |
| log.folder | logs | Property file | Output for edelivery smp log files. |
| log.configuration.file | | Property file | Custon log4j configuration file. |
| target-database (Deprecated) | MySQL | Property file | Target Database Backend type/Brand: |

| Parameter | Default Value | Location | Comment |
|---|---|---|---|
| | | | For MySQL, use:<br>MySQL<br>For Oracle Database, use:<br>Oracle |
| jdbc.read-connections.max (Deprecated) | 10 | Property file | Database Configuration: Max Read Connection |
| bdmsl.integration.proxy.server (Deprecated) | | Database/Property file | When an SMP calls the SML and the SML is behind a proxy, then standard proxy settings need to be specified.<br>Proxy server hostname |
| bdmsl.integration.proxy.port (Deprecated) | | Database/Property file | Proxy server port |
| bdmsl.integration.proxy.user (Deprecated) | | Database/Property file | Proxy server user |
| bdmsl.integration.proxy.password (Deprecated) | | Database/Property file | Proxy server password |
| smp.proxy.host | | Database/Property file | The http proxy host. If value is null or empty – proxy for SML integration or certificate CRL retriever is not configured. When accessing URLs with scheme HTTPS you should also set appropriate value to application servers system attribute:<br>JAVA_OPTS=-Djdk.http.auth.tunneling.disabledSchemes=<br>From Java version 8_111 on, https scheme is disabled by default. |
| smp.noproxy.hosts | localhost\|127.0.0.1 | Database/Property file | List of no proxy hosts. Ex.: localhost\|127.0.0.1 |
| smp.proxy.port | 80 | Database/Property | The http proxy port |

| Parameter | Default Value | Location | Comment |
|---|---|---|---|
| | | file | |
| smp.proxy.user | | Database/Property file | Optional user for proxy authentication |
| smp.proxy.password | | Database/Property file | Encrypted password for Proxy or use format {DEC}{[password}}. Password will be automatically encrypted and properties update. |

#### *16.2.1.1. bdmsl.integration.url*

Only used if SML/DNS integration has been selected.

In case the integration with the SML/DNS is needed, this value has to be set as the address of the SML or the loadBalancer/proxy tagetting these SML instance(s).

## 16.3. smp_domain table configuration

This table is used to support the multi-tenancy 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 domain. Domain code can be alphanumeric and up to 63 characters long.

# 17. 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”, than application root context might change as well.

Example:

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

Three types of roles are defined in the SMP admin console:

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

- **SMP Administrator**: the SMP administrator can create/delete Service Groups, manage users and domains for service groups and its extensions and metadata. The SMP administrator has access to all Services Groups registered in the SMP.

- **Service Group Administrator**: this user can administer only the metadata for the Service Groups that he owns. He cannot change ownerships or the domains for the existing Service Group.

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 including an SMP Administrator and a Service group Administrator. They, however, cannot change the role of another System Administrator.

# 18. Contact Information

CEF Support Team

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

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