Setting up a Service Metadata Publisher for Peppol

Author: Philip Helger, philip@helger.com, Version: 2021-07-08

Status: Release Candidate

Table of contents

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Statement of copyright

1   Introduction

This document elaborates the steps necessary to deploy and configure a Service Metadata Publisher (SMP) for the use in the Peppol network. The intended audience for this document are technical experts that want to setup their own SMP for Peppol usage.

This document explains the application "phoss SMP" only. Other implementations are not covered. This document takes into account different approaches for deploying an SMP, but there might be others ways as well.

This document is meant to be read top to bottom, so it is not recommended to jump between the different chapters, unless you know what you are doing.

1.1   Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.2   Bibliography

The following normative references are used in this document:

[CODELIST]   Peppol Code Lists,
                https://docs.peppol.eu/edelivery/codelists/

[PFUOI]   Peppol Policy for the use of identifiers, v4.0.0, 2019-01-28,
                https://docs.peppol.eu/edelivery/policies/PEPPOL-EDN-Policy-for-use-of-
                identifiers-4.0-2019-01-28.pdf

[RFC2119]   RFC 2119, Key words for use in RFCs to Indicate Requirement Levels, March 1997,
                https://datatracker.ietf.org/doc/html/rfc2119

[TRANSSEC]   Peppol Policy for Transport Security, v1.0.0, 2019-0-30,
                https://docs.peppol.eu/edelivery/policies/PEPPOL-EDN-Policy-for-Transport-
                Security-1.0-2019-01-31.pdf

2 Prerequisites

This chapter contains the prerequisites that need to be fulfilled to operate an instance of phoss SMP.

2.1 Hardware minimum requirements

The hardware requirements are generally low for phoss SMP. The system will idle most of the time, as it is only involved for one request per exchanged message.

- A machine with Windows or Linux on it (Linux is preferred)
- At least 4GB of RAM
- The amount of storage depends on your setup, but assume at least 5 GB for the SMP itself – the exact amount depends on your system
- Internet connectivity

2.2 Software minimum requirements

- Java 1.8 or newer – it was tested with AdoptOpenJDK, Open JDK and Oracle JDK
- Tomcat 8.5 or 9.0 is recommended as the JavaEE application server – the SMP also works knowingly with Jetty and WildFly
- httpd or nginx as reverse proxy for TLS handling
- A TLS certificate, following the rules described in [TRANSSEC] – wildcard certificates are okay, as long as they follow the rules.
- Depending on the data storage system that you chose (see below in chapter 3.2) an additional database system might be needed

2.3 Certificates

- A specific Peppol SMP certificate is needed.
- The certificate should look similar to the following one, as a chain three certificates:

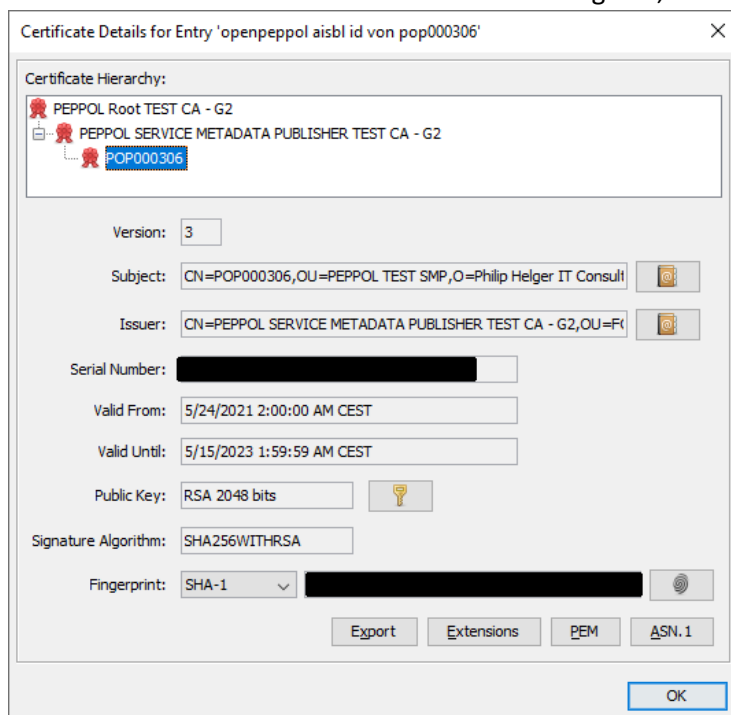*Figure 1: Example screenshot of an SMP test certificate in KeyStore Explorer*

- Keep this keystore private and don't share it with anyone.

- The keystore format can be either JKS (Java Key Store) or PKCS12 (PKCS #12)

2.4 Monitoring

To include an instance of phoss SMP in your infrastructure monitoring, it is recommended to use the status API for that. See https://github.com/phax/phoss-smp/wiki/Status-API for details.

3   phoss SMP introduction

phoss SMP is an Open Source SMP application developed by Philip Helger for usage in the Peppol and other eDelivery systems. The project website is https://github.com/phax/phoss-smp/ and a public Wiki is available at https://github.com/phax/phoss-smp/wiki. The application is a multi-user web application that needs to be run in a JavaEE application server. It offers a web-based GUI as well as a REST API for interacting with it. The current version, at the time of writing of this document, is v5.3.2.

3.1   Existing phoss SMP users

If you are already using phoss SMP for a different project, you anyway need to setup a new instance of it, because the DNS zone it needs to serve is specific to the Peppol project and not compatible with other projects like TOOP or DE4A.

3.2   Version/variant selection

phoss SMP supports three different kinds of data storage systems (called "backend" in the documentation):

- The file system (using a built-in XML database)
- A relational database system (MySQL or PostgreSQL)
- A non-relational database system (MongoDB)

For each of these system, different artefacts are available, that indicate the data storage (`xml`, `sql` and `mongodb`). The functionality of the SMP itself is identical, independent of the backend technology used.

My personal recommendation is to use the "XML" backend, since it is the easiest to setup. Since the amount of data will usually be quite small, there is (in the author's opinion) no need for a fully-fledged database system.

The rest of the document assumes you are using the "XML" version and will not mention any database details.

3.3   Installation variants

To operate an instance of phoss SMP, you have basically two options:

- Run it directly in an existing JavaEE application server (like Tomcat or Jetty). In this case you need to download the binary WAR file from https://github.com/phax/phoss-smp/releases (pick `phoss-smp-webapp-xml-5.3.2.war` for the XML backend).
- Run it in a Docker container – either directly or as part of a Kubernetes cluster. A description of the coordinates can be found at https://github.com/phax/phoss-smp/tree/master/docker (use `phelger/phoss-smp-xml:latest` for the XML backend)

Independent of the deployment option, the SMP needs a writable directory where it stores all it's data. When using a JavaEE application, the directory should be outside the JavaEE server directory (e.g. using `/var/smp` would be an option). When running as a Docker image, please make sure that the volume is mounted from the host system (see the Docker related website for details).

Application server specific detail configurations are described in the Wiki at https://github.com/phax/phoss-smp/wiki/Running.

Additional documentation on how a system can be hardened can be found at https://github.com/phax/phoss-smp/wiki/Security.

3.3.1   Path selection

It is highly recommended, to install the SMP as the `ROOT`[1] application, so that it is accessible via the path `/` on the server.

If you choose to not use the root path, make sure that the configuration item `smp.forceroot` in file `smp-server.properties` to the value `true`. Than you MUST configure a reverse proxy that enforces the usage of the "/" path to the outside world.

3.3.2   Integrating with a reverse proxy

phoss SMP MUST NOT[2] use https when run in Peppol mode. Details on how to integrate phoss SMP with a reverse proxy server like httpd, nginx and IIS are also described in the Wiki at https://github.com/phax/phoss-smp/wiki/Running.

3.3.3   Security considerations

From a security perspective, the recommended scenario is to additionally configure the SMP to run on HTTPS (on any port other than 80), and do the modifying actions only via HTTPS. BasicAuth is required anyway but the data is not readable by third-parties because of the underlying transport security. This is something that is currently technically not available but should be used as a convention when running an SMP with this implementation. See https://github.com/phax/phoss-smp/wiki/Security for further details on security

3.3.4   Verifying the installation

To verify that the installation was successful, ensure that all necessary components are running (reverse proxy, application server, Docker image, Kubernetes cluster etc.), open a local browser and locate the SMP (e.g. via https://my-smp.example.org) – since the exact name is installation dependent, this URL is denoted by the placeholder "{server}" in the rest of the document. If you get redirected to `{server}/public` and see a screen very similar to the one below the installation seems to be okay:
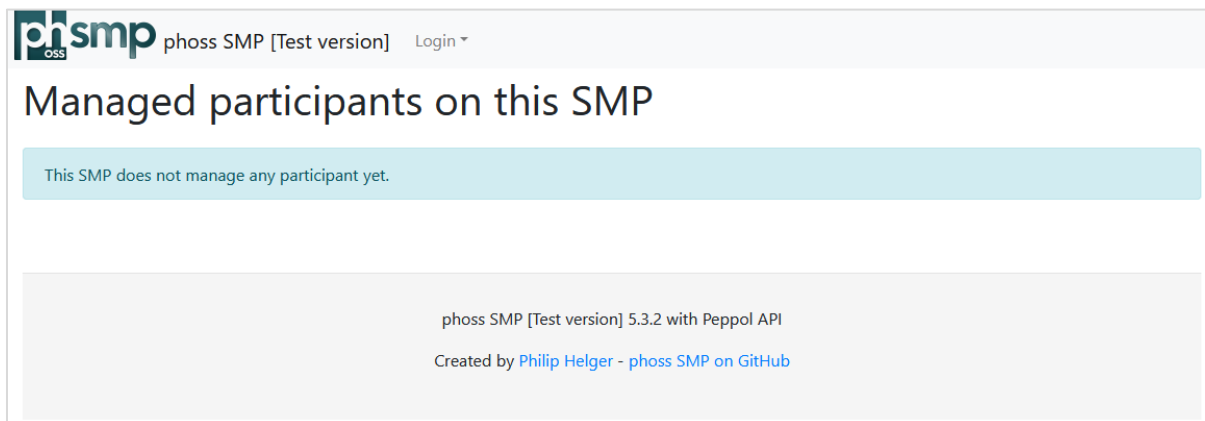

Figure 2: phoss SMP public start screen

Alternatively, you can also open `{server}/secure` in your browser to directly navigate to the management GUI. Then you should see a screen like this:

---

[1] The name „ROOT" is standard value in JavaEE application servers as the alias for the root directory of the server.
[2] See chapter 5.1 for the Peppol SMP specification: "A service implementing the REST profile MUST NOT use TLS (Transport Layer Security) or SSL (Secure Sockets Layer)."
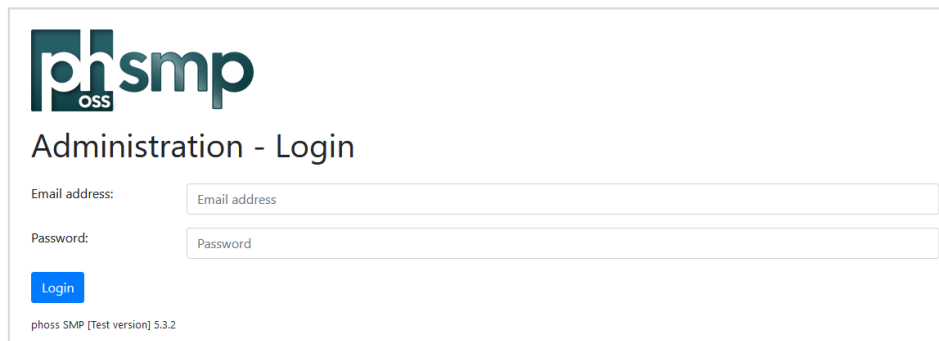
Figure 3: phoss SMP login screen

For error handling, please see the application server log files. For Tomcat these are usually `catalina.out` and `localhost.YYYY-MM-DD.log` (the exact filenames depend on the installation and version of Tomcat used). For the Docker image `docker logs` does not always show the full truth. In case the error is not contained, stepping into the running image (e.g. via `docker exec -it phoss-smp bash`) and then finding the Tomcat log files may be inevitable.

3.4   SMP Configuration

The most difficult thing is to configure the SMP properly. The Wiki elaborates extensively about the different configuration files and the potential values: https://github.com/phax/phoss-smp/wiki/Configuration. This chapter focuses only on the minimum elements that MUST be changed anyway.

You need to have the SMP JKS or PKCS12 key store (see chapter 2.3) to fully finalize the configuration.

Since the phoss SMP configuration mainly consists of three different configuration files, the necessary changes are outlined for each of them separately.

Note: each change to one of the configuration files requires a restart of the application.

Note: placeholders or variables cannot be used in the phoss SMP properties files.

3.4.1   webapp.properties

This file contains settings mainly related to the web frontend.

- `global.debug` SHOULD be set to `false`
- `global.production` SHOULD be set to `true`
- `global.debugjaxws` SHOULD be set to `false`
- `webapp.datapath` MUST be set to the absolute directory, where all data will be stored.

3.4.2   smp-server.properties

This file contains settings mainly related to the SMP functionality itself

- `smp.backend` MUST be set to `xml`
- `smp.keystore.type` MUST be set to `JKS` or `PKCS12` depending on your keystore
- `smp.keystore.path` MUST be set to the absolute path where the SMP certificate resides on your server
- `smp.keystore.password` MUST be set to the plaintext password of the keystore. Mind trailing spaces!

- `smp.keystore.key.alias` MUST be set to the name of the entry/alias in the keystore.
- `smp.keystore.key.password` MUST be set to the plaintext password of the key. Usually this is the same as for the full keystore. Mind trailing spaces!
- `sml.smpid` SHOULD be set to `SMP-XX` where `XX` denotes your company details, in all uppercase chars. This ID MUST be unique within the whole Peppol network.
- `smp.publicurl` MUST be set to the public URL of your SMP server (including an eventual application path – e.g. `https://smp.example.org/smp`)
- `smp.identifiertype` MUST be set to `peppol`
- `smp.rest.type` MUST be set to `peppol`

See also the remark in chapter 3.3.1.

3.4.3   application.properties

For the SMP this file contains the Directory client configuration. Even though we don't need the Directory at the moment, it is strongly recommended to configure it already. The values in this file are the same as for the SMP in the previous chapter.

- `keystore.type` MUST be set to `JKS` or `PKCS12` depending on your keystore
- `keystore.path` MUST be set to the absolute path where the SMP certificate resides on your server
- `keystore.password` MUST be set to the plaintext password of the keystore. Mind trailing spaces!
- `keystore.key.alias` MUST be set to the name of the entry/alias in the keystore.
- `keystore.key.password` MUST be set to the plaintext password of the key. Usually this is the same as for the full keystore. Mind trailing spaces!

3.5   Initial setup

Once the configuration is finished and the application is running again, it's time to perform the initial setup. Open `{server}/secure` in the browser and login for the first time.

The default username is `admin@helger.com` and the default password is `password`.

3.5.1   General user interface elements

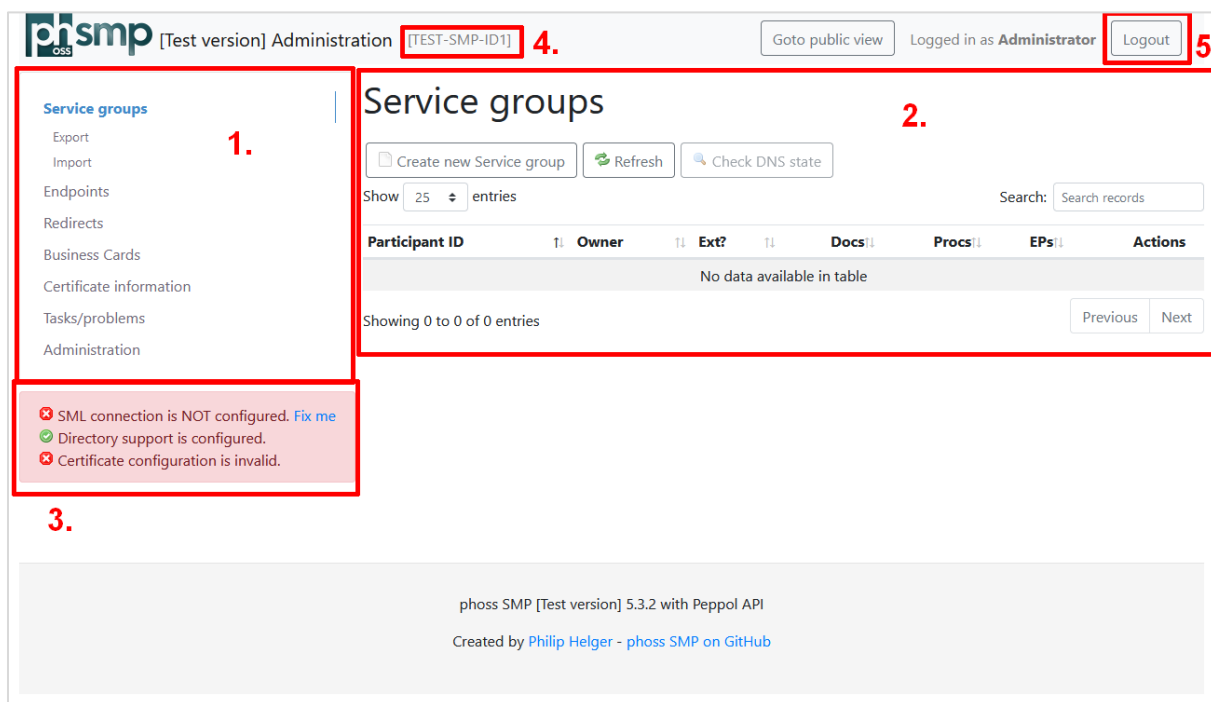After you successfully logged in, the screen looks like depicted in the following figure:

Figure 4: phoss SMP management GUI

The main layout consists of:

- a menu on the left side (denoted by "1.") where all the functionalities can be selected
- a content area right of the menu (denoted by "2."), where the main page content is displayed
- the information area (denoted by "3.") gives you are brief glimpse, if the key settings are okay (green background) or if actions are needed (red background)
- the small text denoted by "4." shows the SMP ID as configured – please make sure it is NOT `TEST-SMP-ID1`.
- the logout button (denoted by "5.") ends the current session for the current user, as each user can only be logged in once. If you forget to logout, the session expires automatically after 30 minutes of inactivity.

Note: some of the screenshots in this document contain a red information area, because no certificate was configured at that point in time. In practise it should be green, once you finished all the necessary steps.

3.5.2   Change default password (highly recommended)

The first thing to do is to change the password of the user. The menu item to use is "Administrator | Change password".

Figure 5: Change password

3.5.3   Change default email address (optional)

If you optionally also want to change the email address of the Administrator, follow these steps:

a)   Navigate to the menu item "Administration | Security | User management" and edit the "Administrator" user by clicking on the small pencil icon:

Figure 6: User management overview

b)   Change the "Email address" and press "Save":

Figure 7: Edit the Administrator email address

Note: you can also change the name of the user if you like

Note: the "Administrator" user is a special user and modification options are limited compared to other users, to avoid locking him out of the system.

3.5.4   Select the SML for usage

After creating the necessary SML configuration in the previous step, it needs to be selected for usage. Therefore, open the menu item "Administration | SMP Settings" and press the "Edit" button:

Figure 8: SMP settings page

Then alter the values like this:

- The checkbox for field "SML connection enabled?" should be enabled (checked)
- For the field "SML configuration:" select the "SMK" entry
- Set the field "Peppol Directory hostname" to https://directory.peppol.eu/ for the Peppol production system, and https://test-directory.peppol.eu for the Peppol test system.

Afterwards, press the "Save" button.

Figure 9: Editing the SMP settings

If you configured everything correctly, the information area should now have turned green:

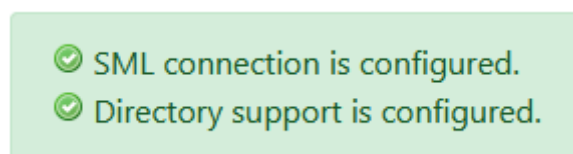*Figure 10: Information area without any issues*

3.5.5 Verify the configuration

As a last check before finalizing the registration, open the "Tasks/problems" menu item and check, that the page contains no error. The below figure shows three warnings, but no error meaning, we're good to continue. In general, you can check on this page at any time if there are severe configuration errors identified.

Figure 11: Tasks/problems page

3.5.6   Register the SMP to the SML

Now the configuration is completed and the SMP is ready to be registered to the SML. To do this, open the menu item "Administration | SML | SML registration".

Figure 12: Register SMP to SML

The field "Physical address" SHOULD contain the public IP address of your server. Since this value is not used, if the "Logical address" is provided, it is okay to use the value "1.1.1.1" in here. The field "Logical address" MUST contain the fully qualified URL of the server, including the protocol and an eventually present application path (not present in the above figure).

Afterwards press the "Register SMP at SML" button. If everything works as expected, you should see a screen similar to the following figure:

Figure 13: Successful registration to SML message

4 Operating the SMP

Once the configuration is completed, the regular operations of an SMP starts. The most common things that will be done, is to create "Participants" which are called "Service Groups" in SMP terminology. Each participant is ensured to be registered by exactly one SMP.

This chapter does not explain all the details of operating an SMP as it should be straight forward, it just gives a very brief overview.

4.1 Participant / Service Group management

To create a new Service Group, open the menu item "Service groups" and press the button "Create new Service group".
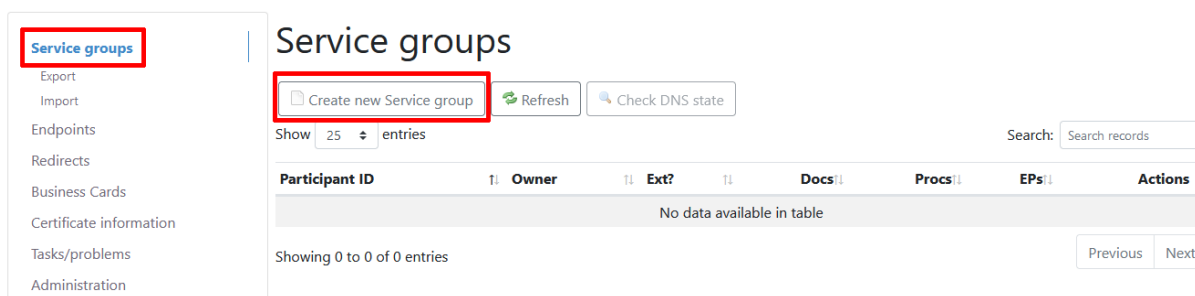
Figure 14: Service Group list

The only relevant field to be filled out is the "Participant ID" field.

- The "Identifier scheme" subfield MUST be set to the value iso6523-actorid-upis as specified in [PFUOI].
- The "Identifier value" subfield MUST contain the participant identifier value, also following the rules of [PFUOI].
- The "Extension" field MUST stay empty

Afterwards press the "Save button".

Figure 15: Create a new Service Group

This operations writes into the SML. If everything worked, you see a green success box, if not a red error box is shown. If an error occurred this usually means one of the following:

- that your certificate configuration is invalid
- or that the identifier value is syntactically not according to the rules
- or that the identifier value was already taken by a different SMP in the network

You can also create and delete a Service Group via the REST API – see
https://github.com/phax/phoss-smp/wiki/REST-API for details.

4.2   Endpoint management

An SMP Endpoint is always linked to a single Service Group. It is identified by the quadruple of
Service Group, Document Type identifier, Process Identifier and Transport Profile. To manage
Endpoints, open the menu item "Endpoints | Endpoint List". Use the button "Create new Endpoint"
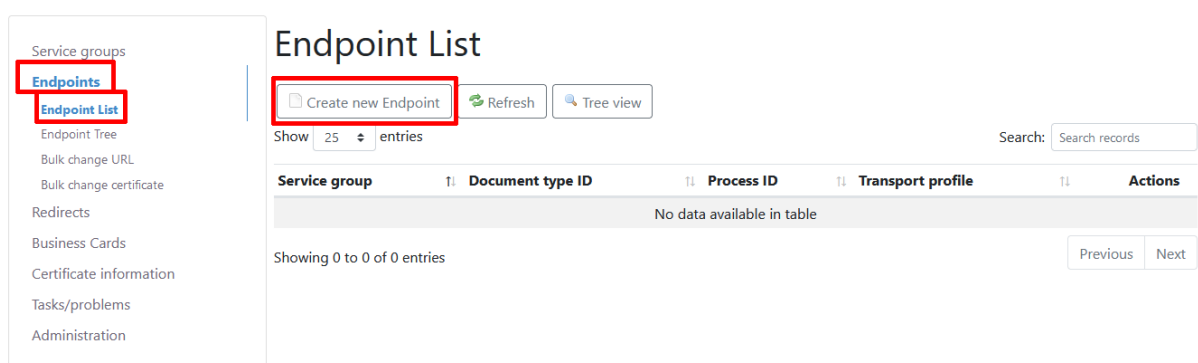to create a new endpoint.

Figure 16: SMP Endpoint list

When creating a new Endpoint, the following fields are relevant:

- The "Service group" MUST be selected from the drop-down select box.
- The "Document type ID" scheme MUST be `busdox-docid-qns` according to [PFUOI]
- The "Document type ID" value MUST be one of the code list values according to [CODELIST]
- The "Process ID" scheme MUST be `cenbii-procid-ubl`
- The "Process ID" value MUST be one of the code lists values according to [CODELIST].
- The "Transport Profile" MUST be one of the code lists values according to [CODELIST].
  Currently the only allowed value is `peppol-transport-as4-v2_0` which by default has
  the name "PEPPOL AS4 v2" in phoss SMP.
- The field "Endpoint Reference" MUST contain the URL of the AS4 endpoint to be used,
  including the protocol and the path. This is the URL where the AS4 message is send to.
- The field "Certificate" MUST contain the PEM encoded certificate of your AS4 certificate. This
  must be text content that starts with `-----BEGIN CERTIFICATE-----` and ends with
  `-----END CERTIFICATE-----`.
- The field "Service Description" MUST be filled – the content is for humans only.
- The field "Technical Contact" MUST be filled – the content is for humans only.

Afterwards press the "Save" button.

Endpoint List

Create new endpoint

Service group*: iso6523-actorid-upis::9915:helger [admin@helger.com (Administrator)]

Document type ID*: busdox-docid-qns
urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##urn:cen.eu:en16931:2017#

Process ID*: cenbii-procid-ubl
urn:fdc:peppol.eu:2017:poacc:billing:01:1.0

Transport Profile*: PEPPOL AS4 v2 (peppol-transport-as4-v2_0)

Endpoint Reference: https://www.helger.com/phase4/as4
The URL where messsages of this type should be targeted to.

☐ Requires Business Level Signature:
Check the box if the recipient requires business-level signatures for the message, meaning a signature applied to the business message before the message is put on the transport. This is independent of the transport-level signatures that a specific transport profile, such as the START profile, might mandate. This flag does not indicate which type of business-level signature might be required. Setting or consuming business-level signatures would typically be the responsibility of the final senders and receivers of messages, rather than a set of APs.

Minimum Authentication Level: Minimum Authentication Level
Indicates the minimum authentication level that recipient requires. The specific semantics of this field is defined in a specific instance of the BUSDOX infrastructure. It could for example reflect the value of the "urn:eu:busdox:attribute:assurance-level" SAML attribute defined in the START specification.

Not before: 📅
Activation date of the service. Senders should ignore services that are not yet activated.

Not after: 📅
Expiration date of the service. Senders should ignore services that are expired.

Certificate*:
```
-----BEGIN CERTIFICATE-----
MIIF2DCCA8CgAwIBAgIQBcAefsdQIk7jCCLVCAicizANBgkqhkiG9w0BAQsFADBr
MQswCQYDVQQGEwJCRTEZMBcGA1UEChMQT3BlblBFUFBPTCBBSVNCTDEWMBQGA1UE
```
Holds the complete signing certificate of the recipient AP, as a PEM base 64 encoded X509 DER formatted value.

Service Description*: Philip Helger Test endpoint
A human readable description of the service.

Technical Contact*: peppol-support@helger.com
Represents a link to human readable contact information. This might also be an email address.

Technical Information: Technical Information
A URL to human readable documentation of the service format. This could for example be a web site containing links to XML Schemas, WSDLs, Schematrons and other relevant resources.

Extension: Extension
Optional extension to the endpoint. If present it must be valid XML content!

[💾 Save] [❌ Cancel]

Figure 17: Create a new SMP Endpoint

Hint: if you want to create a new Endpoint for the same participant, just for a different document type, it is recommended to use the "Copy endpoint" functionality, instead of entering the data manually over and over again. The "copy" action can be interpreted as "Create a new Endpoint using the data of an existing Endpoint". See the below figure for how to do this.
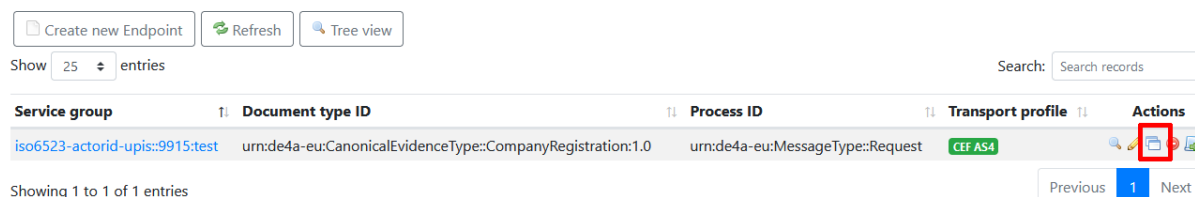
Endpoint List

Figure 18: The "Copy" button

You can also create and delete Endpoints via the REST API – see https://github.com/phax/phoss-smp/wiki/REST-API for details.

4.3 Maintenance tasks

4.3.1 Change an AS4 endpoint URL

Sometimes it is necessary to change the endpoint URL of the AS4 gateway. To avoid editing all Endpoints manually, a functionality to bulk change all endpoint URLs in all SMP endpoint can be found at the menu item "Endpoints | Bulk change URL".

4.3.2 Change/update an AS4 certificate

If it is necessary to update the public AS4 certificate part, there is also a bulk change functionality available. You can find it at the menu item "Endpoints | Bulk change certificate".

Note: see also https://peppol.helger.com/public/locale-en_US/menuitem-docs-peppol-cert-update#ap for additional tasks that need to be done.

4.3.3 Change/update an SMP certificate

If your SMP certificate is expired, multiple activities need to be performed. See https://peppol.helger.com/public/locale-en_US/menuitem-docs-peppol-cert-update#smp for a detailed description.

5   Taking an SMP offline

If you are sure, that you don't need your SMP anymore, you should unregister it from the SML. This implies, that all participants (Service Groups) that you previously registered are made available for registration by other SMPs.

Note: this action cannot be undone!

To do this, open the menu item "Administration | SML | SML registration", select the "Delete SMP from SML" tab, and press the "Delete SMP from SML" button.
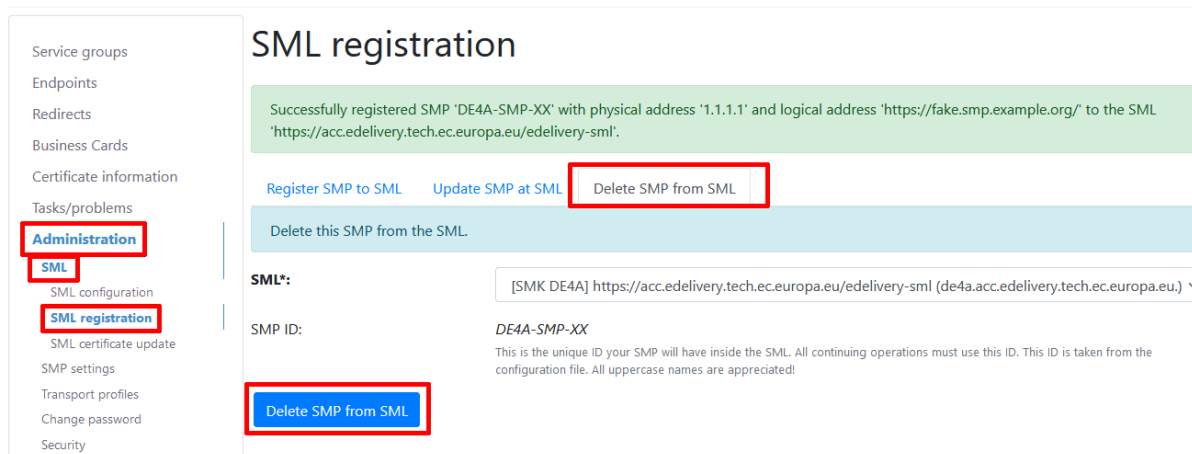

Figure 19: Unregistering an SMP from SML

This should only be done, if you stop supporting Peppol, to free up the used resources if possible.