SimpleSAMLphp Service Provider QuickStart

This guide will describe how to configure SimpleSAMLphp as a service provider (SP). You should previously have installed SimpleSAMLphp as described in the SimpleSAMLphp installation instructions .

Configuring the SP

The SP is configured by an entry in `config/authsources.php` .

This is a minimal `authsources.php` for a SP:

```
<?php
$config = [

    /* This is the name of this authentication source, and will be used to access it later. */
    'default-sp' => [
        'saml:SP',
    ],
];
```

For more information about additional options available for the SP, see the `saml:SP` reference .

If you want multiple Service Providers in the same site and installation, you can add more entries in the `authsources.php` configuration. If so remember to set the EntityID explicitly. Here is an example:

```
'sp1' => [
    'saml:SP',
  'entityID' => 'https://sp1.example.org/',
],
'sp2' => [
    'saml:SP',
  'entityID' => 'https://sp2.example.org/',
],
```

Enabling a certificate for your Service Provider

Some Identity Providers / Federations may require that your Service Providers holds a certificate. If you enable a certificate for your Service Provider, it may be able to sign requests and response sent to the Identity Provider, as well as receiving encrypted responses.

Create a self-signed certificate in the `cert/` directory.

```
cd cert
openssl req -newkey rsa:3072 -new -x509 -days 3652 -nodes -out saml.crt -keyout saml.pem
```

Then edit your `authsources.php` entry, and add references to your certificate:

```
'default-sp' => [
    'saml:SP',
    'privatekey' => 'saml.pem',
    'certificate' => 'saml.crt',
],
```

Adding IdPs to the SP

The service provider you are configuring needs to know about the identity providers you are going to connect to it. This is configured by metadata stored in `metadata/saml20-idp-remote.php` and `metadata/shib13-idp-remote.php` . This is a minimal example of a `metadata/saml20-idp-remote.php` metadata file:

```
<?php
$metadata['https://example.com'] = [
    'SingleSignOnService'  => 'https://example.com/simplesaml/saml2/idp/SSOService.php',
    'SingleLogoutService'  => 'https://example.com/simplesaml/saml2/idp/SingleLogoutService.php',
    'certificate'          => 'example.pem',
];
```

`example.pem` under your `cert/` directory contains the certificate the identity provider uses for signing assertions.

For more information about available options in the idp-remote metadata files, see the IdP remote reference .

If you have the metadata of the remote IdP as an XML file, you can use the built-in XML to SimpleSAMLphp metadata converter, which by default is available as `/admin/metadata-converter.php` in your SimpleSAMLphp installation.

Note that the idp-remote file lists all IdPs you trust. You should remove all IdPs that you don't use.

Setting the default IdP

An option in the authentication source allows you to configure which IdP should be used. This is the `idp` option.

```
<?php
$config = [

    'default-sp' => [
        'saml:SP',

        /*
         * The entity ID of the IdP this should SP should contact.
         * Can be NULL/unset, in which case the user will be shown a list of available IdPs.
         */
        'idp' => 'https://idp.example.com',
    ],
];
```

Exchange metadata with the IdP

In order to complete the connection between your SP and an IdP, you must exchange the metadata of your SP with the IdP. The metadata of your SP can be found in the *Federation* tab of the web interface. Copy the SAML 2.0 XML Metadata document automatically generated by SimpleSAMLphp and send it to the administrator of the IdP. You can also send them the dedicated URL of your metadata, so that they can fetch it periodically and obtain automatically any changes that you may perform to your SP.

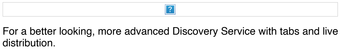
You will also need to add the metadata of the IdP. Ask them to provide you with their metadata, and parse it using the *XML to SimpleSAMLphp metadata converter* tool available also in the *Federation* tab of the web interface. Copy the resulting parsed metadata and paste it with a text editor into the `metadata/saml20-idp-remote.php` file in your SimpleSAMLphp directory.

If you intend to add your SP to a federation, the procedure for managing trust in federations will differ, but the common part is that you would need to provide the *SAML 2.0 metadata of your SP* , and register that with the federation administration. You will probably be required too to consume the federation metadata periodically. Read more about automated metadata management to learn more about that.

Test the SP

After the metadata is configured on the IdP, you should be able to test the configuration. The installation page of SimpleSAMLphp has a link to test authentication sources. When you click the link, you should receive a list of authentication sources, including the one you have created for the SP.

After you click the link for that authentication source, you will be redirected to the IdP. After entering your credentials, you should be redirected back to the test page. The test page should contain a list of your attributes:



For a better looking, more advanced Discovery Service with tabs and live search, you may want to use the `discopower` module contained in the SimpleSAMLphp distribution.

Integrating authentication with your own application

The API is documented in the SP API reference .

For those web resources you want to protect, you must add a few lines of PHP code:

- Register the SimpleSAMLphp classes with the PHP autoloader.
- ```
  Require authentication of the user for those places it is required.
  ```
- ```
  Access the users attributes.
  ```

Example code:

We start off with loading a file which registers the SimpleSAMLphp classes with the autoloader.

```
require_once('../../lib/_autoload.php');
```

We select our authentication source:

```
$as = new \SimpleSAML\Auth\Simple('default-sp');
```

We then require authentication:

```
$as->requireAuth();
```

And print the attributes:

```
$attributes = $as->getAttributes();
print_r($attributes);
```

Each attribute name can be used as an index into $attributes to obtain the value. Every attribute value is an array - a single-valued attribute is an array of a single element.

We can also request authentication with a specific IdP:

```
$as->login([
    'saml:idp' => 'https://idp.example.org/',
]);
```

Other options are also available. Take a look in the documentation for the SP module for a list of all parameters.

If we are using PHP sessions in SimpleSAMLphp and in the application we are protecting, SimpleSAMLphp will close any existing session when invoked for the first time, and its own session will prevail afterwards. If you want to restore your own session after calling SimpleSAMLphp, you can do so by cleaning up the session like this:

```
$session = \SimpleSAML\Session::getSessionFromRequest();
$session->cleanup();
```

If you don't cleanup SimpleSAMLphp's session and try to use $_SESSION afterwards, you won't be using your own session and all your data is likely to get lost or inaccessible.

Note that if your application uses a custom session handler , SimpleSAMLphp will use it as well. This can lead to problems because SimpleSAMLphp's stand-alone web UI uses the default PHP session handlers. Therefore, you may need to unset the custom handler before making any calls to SimpleSAMLphp:

```
// use custom save handler
session_set_save_handler($handler);
session_start();

// close session and restore default handler
session_write_close();
session_set_save_handler(new SessionHandler(), true);

// use SimpleSAML\Session
$session = \SimpleSAML\Session::getSessionFromRequest();
$session->cleanup();
session_write_close();

// back to custom save handler
session_set_save_handler($handler);
session_start();
```

Support

If you need help to make this work, or want to discuss SimpleSAMLphp with other users of the software, you are fortunate: Around SimpleSAMLphp there is a great Open source community, and you are welcome to join! The forums are open for you to ask questions, contribute answers other further questions, request improvements or contribute with code or plugins of your own.

- SimpleSAMLphp homepage
- List of all available SimpleSAMLphp documentation
- Join the SimpleSAMLphp user's mailing list