

Single Sign On. SAML extension for M 2 (frontend, Implemented by Sixto Martin)

Installation

The SAML extension which version is equal or greater to 1.4 requires php-saml 3.X. Otherwise, use php-saml 2.X

Steps to install the Magento 2.0 connector via Composer

1. Set up the correct path for Composer or keep Composer within Magento root.
2. At Magento root folder, run command
`composer require sixtomartin/onelogin-module-saml2`
After it is installed, verify that in the vendor folder you see a sixtomartin folder, but also a onelogin/php-saml folder
3. After the above is successful, run the command
`php bin/magento module:enable Pitbulk_SAML2`
at Magento root folder. This will let Magento know about the module.
4. Run the command
`php bin/magento setup:upgrade`
at Magento root folder. This will update the database schema.
5. Run the command
`php bin/magento setup:di:compile`
if you have a single website and store, or
`php bin/magento setup:di:compile-multi-tenant`
if you have multiple ones.
6. Clear cache from Magento admin.

Steps to install the Magento 2.0 connector via FTP/ZIP File

1. Unzip the ZIP file.
2. Make sure to create the directory structure in your Magento:
'Magento_Root/app/code/Pitbulk/SAML2'.
3. Drop/move all of the content inside the unzipped folder to directory
'Magento_Root/app/code/Pitbulk/SAML2'.
4. The SAML extension has php-saml as dependency (2.x or 3.X),

1. At Magento root folder run 'composer require onelogin/php-saml' to install it.
2. In the uncommon scenario that you are not using composer, you will need to download the code from php-saml repository (<https://github.com/onelogin/php-saml>) manually. Create a onelogin/php-saml folder at the 'vendor' folder and drop there:

- the `_toolkit_loader.php` file that will load php-saml library and its dependency xmlseclibs

- the `lib/src` folder

- the `extlib` folder, if exists in the php-saml version that you are using, if not you will need to copy xmlseclibs `src` folder (<https://github.com/robrichards/xmlseclibs>) as xmlseclibs and edit the previous `_toolkit_loader.php` file and uncomment the 'Load xmlseclibs' block and set as `$xmlseclibsSrcDir` the value 'xmlseclibs', the name you gave to the renamed `src` folder of xmlseclibs repo.

Once all that is done, you will need to load the `_toolkit_loader.php` in Magento code so php-saml and xmlseclibs classes will be available. One way to do that is to manually register this loader in the global Magento vendors loader. Edit the `vendor/autoload.php` and place at the end of the file:

```
require_once __DIR__ . '/onelogin/php-saml/_toolkit_loader.php';
```

but notice that if you execute any composer change, it will modify the `autoload.php` file and changes will disappear.. but since you are not using composer, it shouldn't be a problem

5. Run the command
`php bin/magento module:enable Pitbulk_SAML2`
at Magento root folder. This will let Magento know about the module.
6. Run the command
`php bin/magento setup:upgrade`
at Magento root folder. This will update the database schema.
7. Run the command
`php bin/magento setup:di:compile`
if you have a single website and store, or
`php bin/magento setup:di:compile-multi-tenant`
if you have multiple ones.
8. Clear cache from Magento admin.