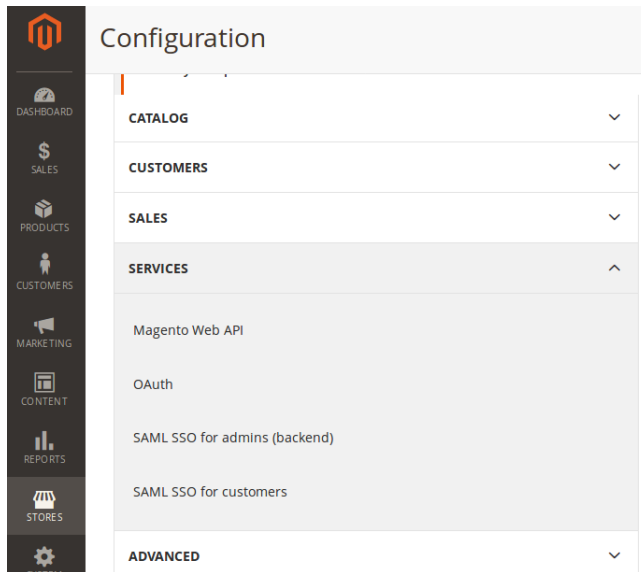


# Single Sign On. SAML extension for M 2 ( frontend + backend, Implemented by Sixto Martin )

The Setting panel contains different sections with fields that need to be filled.

Sections and fields contains a description that contains enough information to understand what info need to be provided.

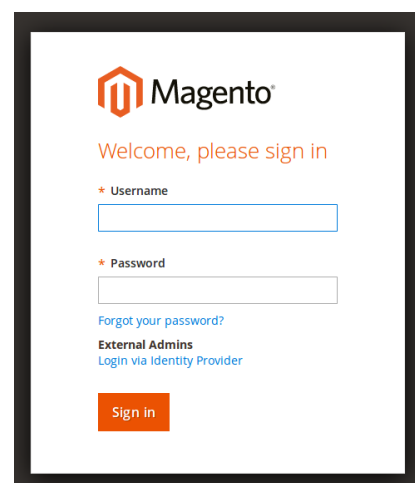
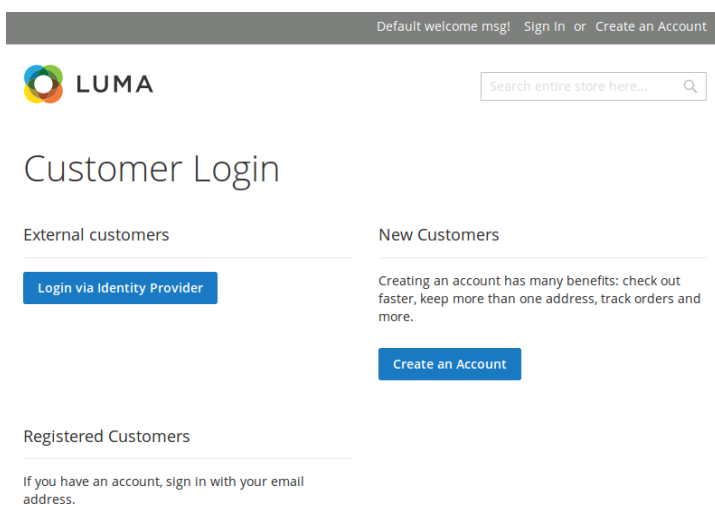


In order to access to the setting panel, log in the admin instance of M2, and got to Stores > Configuration. At the Services tab, the "SAML SSO for customers" link.

The extension support multi-sites, so if you access to the setting panel of each store you will be able to enable SAML on it. (Each store has its own settings).

The Settings for the backend login form are at "SAML SSO for admins (backend)" link.

Once enabled the customer will see the login form modified, with the new "External customer" section. Notice that the title and the text of the button can be customized on the "Custom messages" setting. Similar in the backend area



Those are the sections of the SAML setting panel for the front-end (the backend has similar).

## STATUS

**Enabled**  
[store view]

No

**License KEY**  
[store view]

The Lincese KEY related to the SAML extension

**Metadata of this SP**  
[store view]

<http://pitbulk.no-ip.org/sso/saml2/metadata>

## IDENTITY PROVIDER SETTINGS

Set here some info related to the IdP that will be connected with our Magento. Contact the IdP's administrator and ask him for the IdP metadata

**IdP Entity Id**  
[store view]

Identifier of the IdP entity. ("Issuer URL")

**Single Sign On Service Url**  
[store view]

SSO endpoint info of the IdP. URL target of the IdP where the SP will send the Authentication Request. ("SAML 2.0 Endpoint (HTTP)")

**Single Sign On Service Binding**  
[store view]

urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect

Select if you want to send the AuthNRequest using HTTP-Redirect or HTTP-POST

**Single Log Out Service Url**  
[store view]

SLO endpoint info of the IdP. URL target of the IdP where the SP will send the SLO Request. ("SLO Endpoint (HTTP)")

**X.509 Certificate**  
[store view]

Public x509 certificate of the IdP. ("X.509 certificate")

## OPTIONS

In this section the behavior of the extension is set.

<b>Create user if not exists</b> <small>[store view]</small>	<input type="text" value="Yes"/>	▼
Auto-provisioning. If user not exists, Magento will create a new user with the data provided by the IdP. Review the Mapping section		
<b>Update user data</b> <small>[store view]</small>	<input type="text" value="Yes"/>	▼
Auto-update. Magento will update the account of the user with the data provided by the IdP. Review the Mapping section		
<b>User ID field</b> <small>[store view]</small>	<input type="text" value="Email"/>	▼
Select what field will be used to identify the user: username or email		
<b>Default user role id</b> <small>[store view]</small>	<input type="text" value="9"/>	
If extension is not able to read role from IdP, assign this user role when creating new user account.		
<b>Single Log Out</b> <small>[store view]</small>	<input type="text" value="No"/>	▼
Enable/disable Single Log Out. SLO is a complex functionality, the most common SLO implementation is based on front-channel (redirections), sometimes if the SLO workflow fails a user can be blocked in an unhandled view. If the admin does not controls the set of apps involved in the SLO process maybe is better to disable this functionality due could carry more problems than benefits.		

## ATTRIBUTE MAPPING



Sometimes the names of the attributes sent by the IdP not match the names used by Magento for the customer accounts. In this section we can set the mapping between IdP fields and Magento fields.

<b>Email</b> <small>[store view]</small>	<input type="text"/>
<b>First Name</b> <small>[store view]</small>	<input type="text"/>
<b>Last Name</b> <small>[store view]</small>	<input type="text"/>
<b>Group</b> <small>[store view]</small>	<input type="text"/>

## GROUP MAPPING



The IdP can use its own groups. Set in this section the mapping between IdP and Magento customer groups. Accepts multiple valued comma separated. Example: admin,owner,superuser. There are 5 fields, The Id means that Role id=1 will match the Magento groups that has id=1 if exists. Review the Group list at Stores > Other Settings > Customer Groups

<b>Group id=1</b> <small>[store view]</small>	<input type="text"/>
<b>Group id=2</b> <small>[store view]</small>	<input type="text"/>
<b>Group id=3</b> <small>[store view]</small>	<input type="text"/>
<b>Group id=4</b> <small>[store view]</small>	<input type="text"/>
<b>Group id=5</b> <small>[store view]</small>	<input type="text"/>
<b>Group id=6</b> <small>[store view]</small>	<input type="text"/>
<b>Group id=7</b> <small>[store view]</small>	<input type="text"/>
<b>Group id=8</b> <small>[store view]</small>	<input type="text"/>
<b>Group id=9</b> <small>[store view]</small>	<input type="text"/>
<b>Group id=10</b> <small>[store view]</small>	<input type="text"/>

## ADDRESS MAPPING

If the IdP has address data, set in this section the mapping between IdP address data and Magento.

<b>Company</b> <small>[store view]</small>	<input type="text"/>
<b>Street 1</b> <small>[store view]</small>	<input type="text"/>
<b>Street 2</b> <small>[store view]</small>	<input type="text"/>
<b>City</b> <small>[store view]</small>	<input type="text"/>
<b>Country</b> <small>[store view]</small>	<input type="text"/>
<b>State/Province</b> <small>[store view]</small>	<input type="text"/>
<b>Zip/Postal Code</b> <small>[store view]</small>	<input type="text"/>
<b>Fax</b> <small>[store view]</small>	<input type="text"/>
<b>Telephone</b> <small>[store view]</small>	<input type="text"/>

## CUSTOM MESSAGES

Handle what messages are showed in the login form.

<b>Login Header</b> <small>[store view]</small>	<input type="text"/>	The text that appears as the title, default is: 'External customers'
<b>Login Link</b> <small>[store view]</small>	<input type="text"/>	The text that appears as the link, default is: 'Login via Identity Provider'

---

## ADVANCED SETTINGS

Handle some other parameters related to customizations and security issues.

If sign/encryption is enabled, then x509 cert and private key for the SP must be provided. There are 2 ways:

1. Store them as files named sp.key and sp.crt on the 'certs' folder of the extension. (be sure that the folder is protected and not exposed to internet)
2. Store them at the database, filling the corresponding textareas. (take care of security issues)

<b>Debug Mode</b> <small>[store view]</small>	<input type="text" value="No"/>	Enable it when your are debugging the SAML workflow. Errors and Warnigs will be showed
--	---------------------------------	--

<b>Strict Mode</b> <small>[store view]</small>	<input type="text" value="No"/>	If Strict mode is Enabled, then Magento will reject unsigned or unencrypted messages if it expects them signed or encrypted. Also will reject the messages if not strictly follow the SAML standard: Destination, NameId, Conditions ... are validated too
---	---------------------------------	--

<b>SP Entity Id</b> <small>[store view]</small>	<input type="text"/>	Set the Entity ID for the Service Provider. If not provided, the url where the metadata is published will be used.
--	----------------------	--

<b>NameID Format</b> <small>[store view]</small>	<input type="text" value="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress"/>	Specifies constraints on the name identifier to be used to represent the requested subject.
---	---	---

<b>Encrypt nameID</b> <small>[store view]</small>	<input type="text" value="No"/>	The nameID sent by this SP will be encrypted
--	---------------------------------	--

<b>Sign AuthnRequest</b> <small>[store view]</small>	<input type="text" value="No"/>	The samlp:AuthnRequest messages sent by this SP will be signed
---	---------------------------------	--

<b>Sign LogoutRequest</b> <small>[store view]</small>	<input type="text" value="No"/>	The samlp:logoutRequest messages sent by this SP will be signed
--	---------------------------------	---

<b>Sign LogoutResponse</b> <small>[store view]</small>	<input type="text" value="No"/>	The samlp:logoutResponse messages sent by this SP will be signed
---	---------------------------------	--

<b>Reject Unsigned Messages</b> <small>[store view]</small>	<input type="text" value="No"/>	Reject unsigned samlp:Response, samlp:LogoutRequest and samlp:LogoutResponse received
--	---------------------------------	---

<b>Reject Unsigned Assertions</b> <small>[store view]</small>	<input type="text" value="No"/>	Reject unsigned saml:Assertion received
--	---------------------------------	---

<b>Reject Unencrypted Assertions</b> <small>[store view]</small>	<input type="text" value="No"/>	Reject unencrypted saml:Assertion received
---	---------------------------------	--

### Requested AuthN Context

[\[store view\]](#)

```
urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified
urn:oasis:names:tc:SAML:2.0:ac:classes:Password
urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTrans
urn:oasis:names:tc:SAML:2.0:ac:classes:X509
urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard
urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos
urn:federation:authentication:windows
```

Authentication context. Unselect all to accept any type, otherwise select the valid contexts

### Service Provider X.509 Certificate

[\[store view\]](#)

Public x509 certificate of the SP. Leave this field empty if you gonna provide the cert by the sp.crt

### Service Provider Private Key

[\[store view\]](#)

Private Key of the SP. Leave this field empty if you gonna provide the private key by the sp.key

### Signature Algorithm

[\[store view\]](#)

Algorithm that the toolkit will use on signing process (if enabled).

### Digest Algorithm

[\[store view\]](#)

Algorithm that the toolkit will use on digest process (if signature enabled).

### Lower Case URL Encoding

[\[store view\]](#)

Set True if you are connecting with ADFS and you experience issues validating signatures.

You will need to provide the Service Provider metadata to the Identity Provider administrator.

At the “status” section you will find the link to access a view where the metadata is published (see the source code on the browser to get the XML). It looks like:

```
<?xml version="1.0"?>
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
  validUntil="2016-05-18T18:17:43Z"
  cacheDuration="PT604800S"
  entityID="http://magento2.example.com/sso/saml2/metadata">
  <md:SPSSODescriptor AuthnRequestsSigned="false" WantAssertionsSigned="false"
    protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
      Location="http://magento2.example.com/sso/saml2/sls" />
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</md:NameIDFormat>
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
      Location="http://magento2.example.com/sso/saml2/acs"
      index="1" />
  </md:SPSSODescriptor>
</md:EntityDescriptor>
```