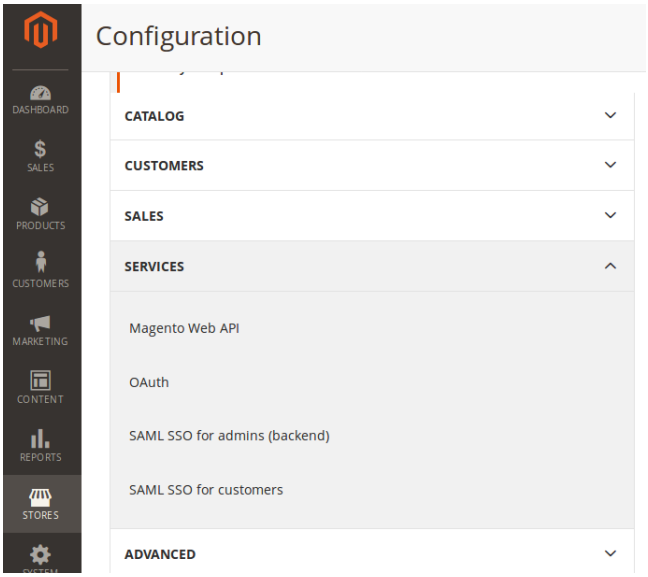
Single Sign On. SAML extension for M 2
(frontend + backend, Implemented by Sixto Martin)

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

Sections and fields contains a description that contains enought information to understand

what info need to be provided.

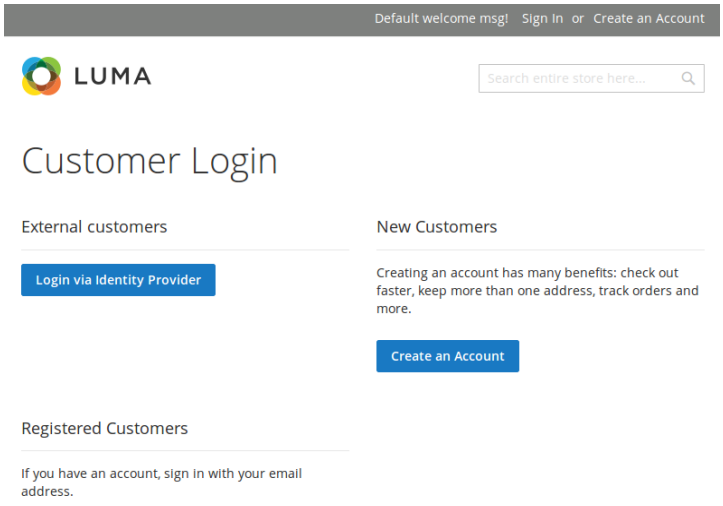


In order to access to the setting panel, log in the admin instance of M2, and got to Stores > Configuration. At the Services tab, the "SAML SSO for customers" link.

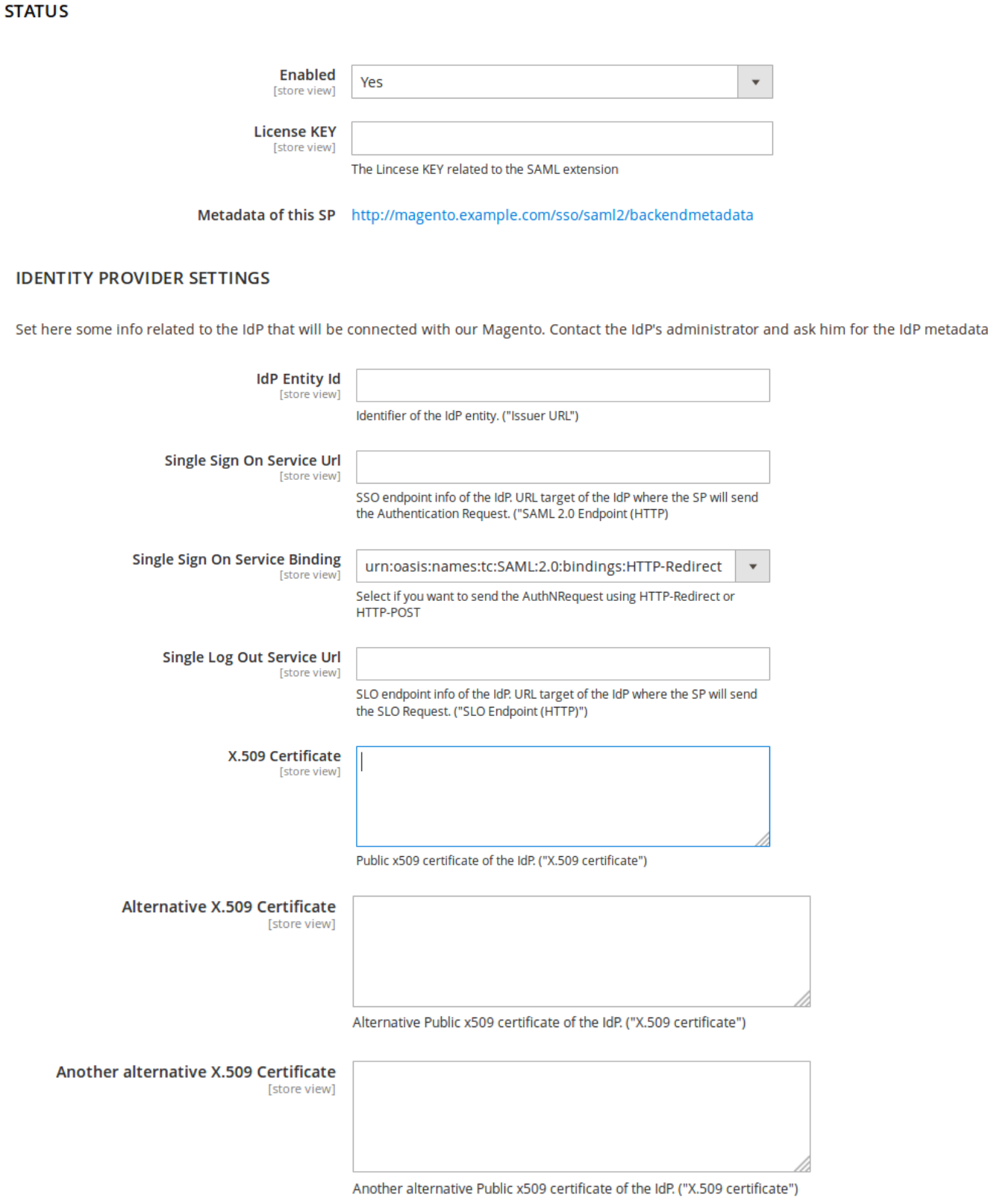
The extension support multi-sites, so if you access to the setting panel of each store you will be able to enable SAML on it. (Each store has its own settings).

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 backend (the frontend has similar and can be found at the User guide of the frontend extension).



OPTIONS

In this section the behavior of the extension is set.

Create user if not exists
[store view]
Yes
Auto-provisioning. If user not exists, Magento will create a new user with the data provided by the IdP.
Review the Mapping section

Update user data
[store view]
Yes
Auto-update. Magento will update the account of the user with the data provided by the IdP (firstname, lastname, roleid).
Review the Mapping section

User ID field
[store view]
Username
Select what field will be used to identify the user: username or email

Require a role when provisioning
[store view]
Yes
If extension is not able to read role from IdP, then will not provision the user.

Default user role id
[store view]
If extension is not able to read role from IdP, and require role is disabled, it will assign this user role when creating new user account.

Single Log Out
[store view]
Yes
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 admin (backend) accounts. In this section we can set the mapping between IdP fields and Magento fields.

Username
[store view]

Email
[store view]

First Name
[store view]

Last Name
[store view]

Role
[store view]

ROLE MAPPING

TThe IdP can use it's own roles. Set in this section the mapping between IdP and Magento backend roles. Accepts multiple valued comma separated. Example: admin,owner,superuser. There are 10 fields, The id means that Role id=1 will match the Magento roles that has id=1 if exists. Review the Role list at System > Roles.

Role id=1 [store view]

Role id=2 [store view]

Role id=3 [store view]

Role id=4 [store view]

Role id=5 [store view]

Role id=6 [store view]

Role id=7 [store view]

Role id=8 [store view]

Role id=9 [store view]

Role id=10 [store view]

CUSTOM MESSAGES

Handle what messages are showed in the login form.

Login Header [store view]

The text that appears as the title, default is: 'External admins'

Login Link [store view]

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)

Debug Mode [store view]

No

Enable it when your are debugging the SAML workflow. Errors and Warnigs will be showed

Strict Mode [store view]

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

SP Entity Id
[store view]

Set the Entity ID for the Service Provider. If not provided, the url where the metadata is published will be used.

NameID Format
[store view]

urn:oasis:names:tc:SAML:1.1:nameid-format:emailAd

Specifies constraints on the name identifier to be used to represent the requested subject.

Encrypt nameID
[store view]

No

The nameID sent by this SP will be encrypted

Sign AuthnRequest
[store view]

No

Sign LogoutRequest
[store view]

No

The samlp:logoutRequest messages sent by this SP will be signed

Sign LogoutResponse
[store view]

No

The samlp:logoutResponse messages sent by this SP will be signed

Reject Unsigned Messages
[store view]

No

Reject unsigned samlp:Response, samlp:LogoutRequest and samlp:LogoutResponse received

Reject Unsigned Assertions
[store view]

No

Reject unsigned saml:Assertion received

Reject Unencrypted Assertions
[store view]

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:PasswordProtecte
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]

http://www.w3.org/2000/09/xmldsig#rsa-sha1

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

Digest Algorithm
[store view]

http://www.w3.org/2000/09/xmldsig#sha1

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

Lower Case URL Encoding
[store view]

No

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

Sign Metadata
[store view]

No

The metadata published by this SP will be signed.

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:

```
− <md:EntityDescriptor validUntil="2019-02-20T18:15:01Z" cacheDuration="PT604800S"
  entityID="http://magento.example.com/sso/saml2/backendmetadata">
  − <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://magento.example.com/admin/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://magento.example.com/admin/admin/index/index" index="1"/>
    </md:SPSSODescriptor>
  </md:EntityDescriptor>
```