niota – SAML configuration

Important! SAML authentication requires **HTTPS** connection.

Prerequisit: meta database table `et_user` must contain column `userid_saml`. Run the following commands to create column if it does not exist:

```
alter table et_user add userid_saml varchar(128) CHARACTER SET UTF8MB4;
create or replace view v_et_user as select * from et_user;
```

Apache configuration

Add two new lines to Apache's config file httpd.conf – default path is `c:/Apache24x64/conf/httpd.conf` which may differs on your environment.

First define a variable for simplesamlphp – framework for SAML authentication used by niota - configuration directory path. Add the following line to the end of the file:

```
SetEnv SIMPLESAMLPHP_CONFIG_DIR
"c:/Apache24x64/htdocs/niota/app/vendor/simplesamlphp/simplesamlphp/config"
```

Take care about Apache's correct path on your environment.

Then you have to define an alias for simplesamlphp's public folder. Simplesamlphp admin page can be accessed here even authentication process will use this folder. Path definition must be changed if Apache's path differs. Add the following line tot he end of file httpd.conf:

```
Alias /simplesaml
"c:/Apache24x64/htdocs/niota/app/vendor/simplesamlphp/simplesamlphp/public"
```

After adding these definitions restart Apache server.

Sample result URL for simplesamlphp opening page. You can nothing to do here. It is only a welcome page:

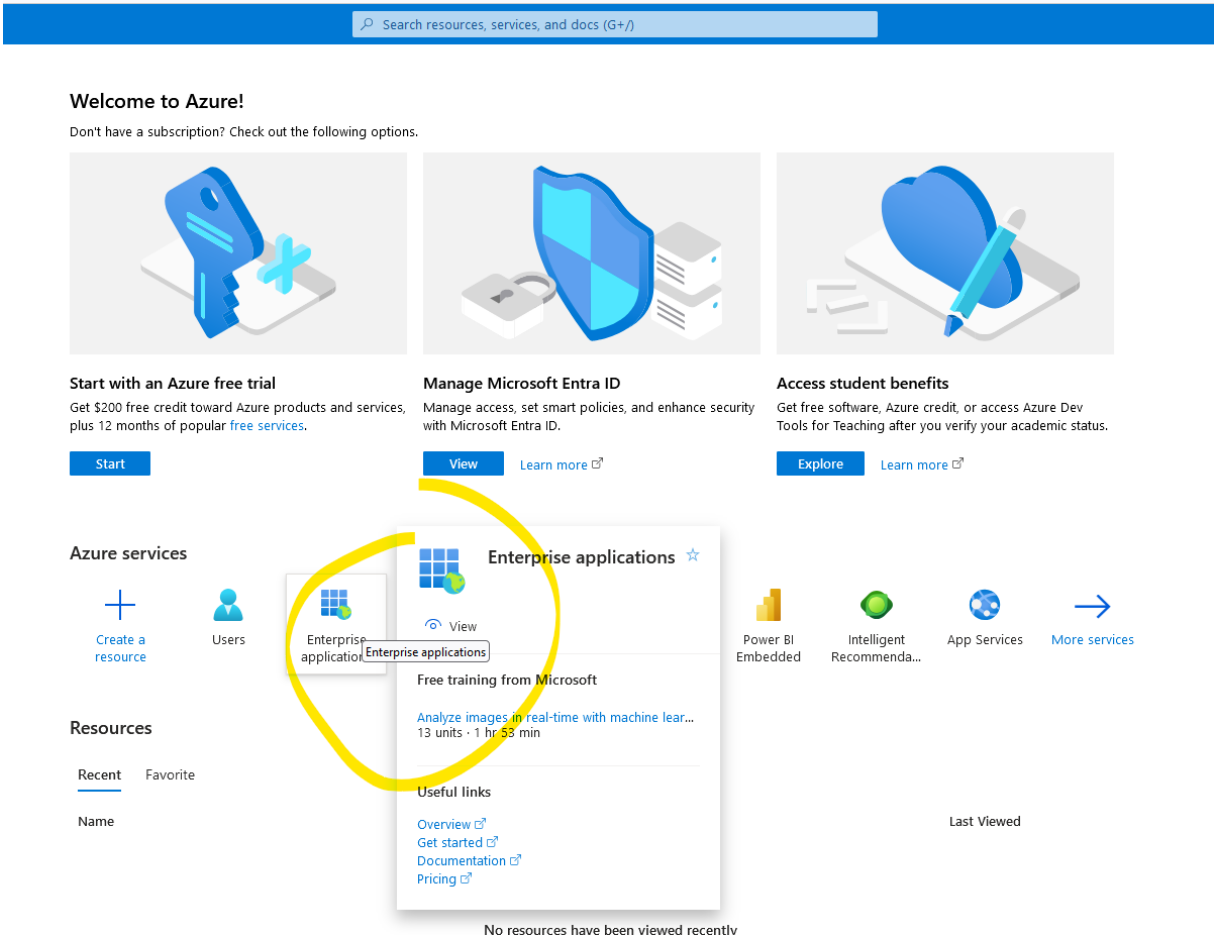
https://office.etixpert.com/simplesaml/index.php

Sample result URL for simplesamlphp admin page which will be used after further configuration:

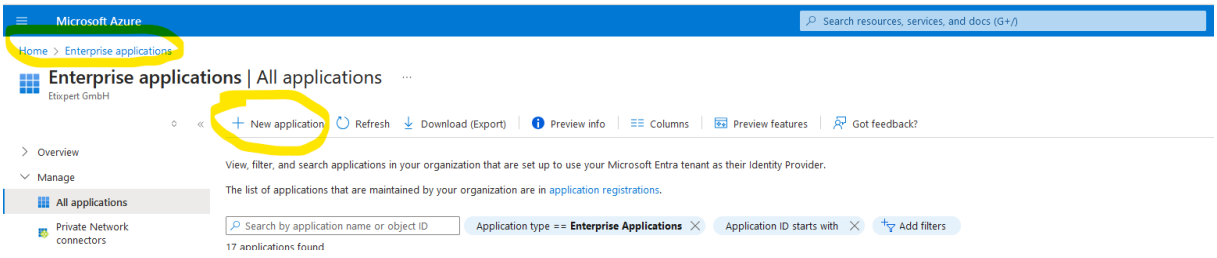
https://office.etixpert.com/simplesaml/admin/index.php

Azure configuration

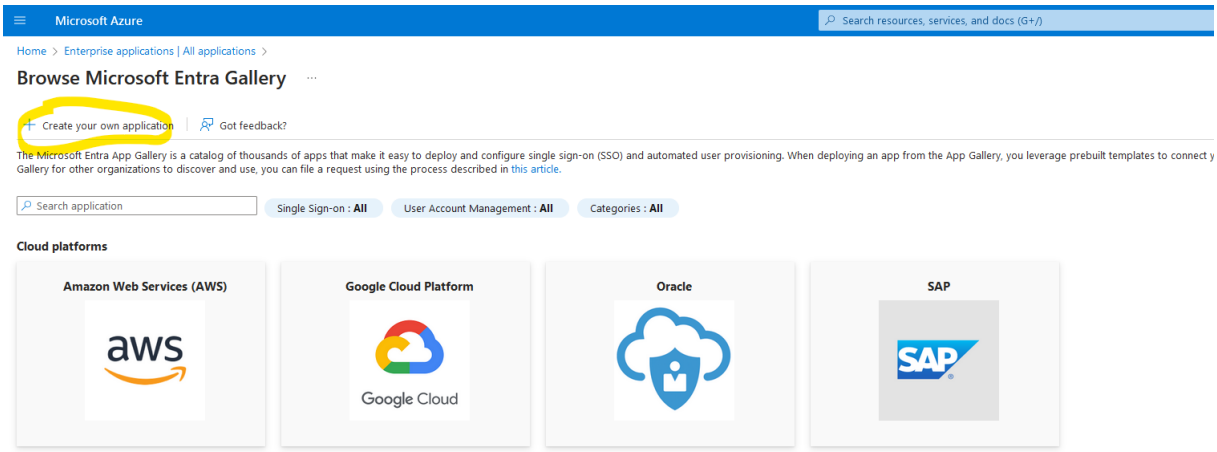
If niota application does not exist in Azure then you have to create it. Go to menu item *Entreprise applications*.



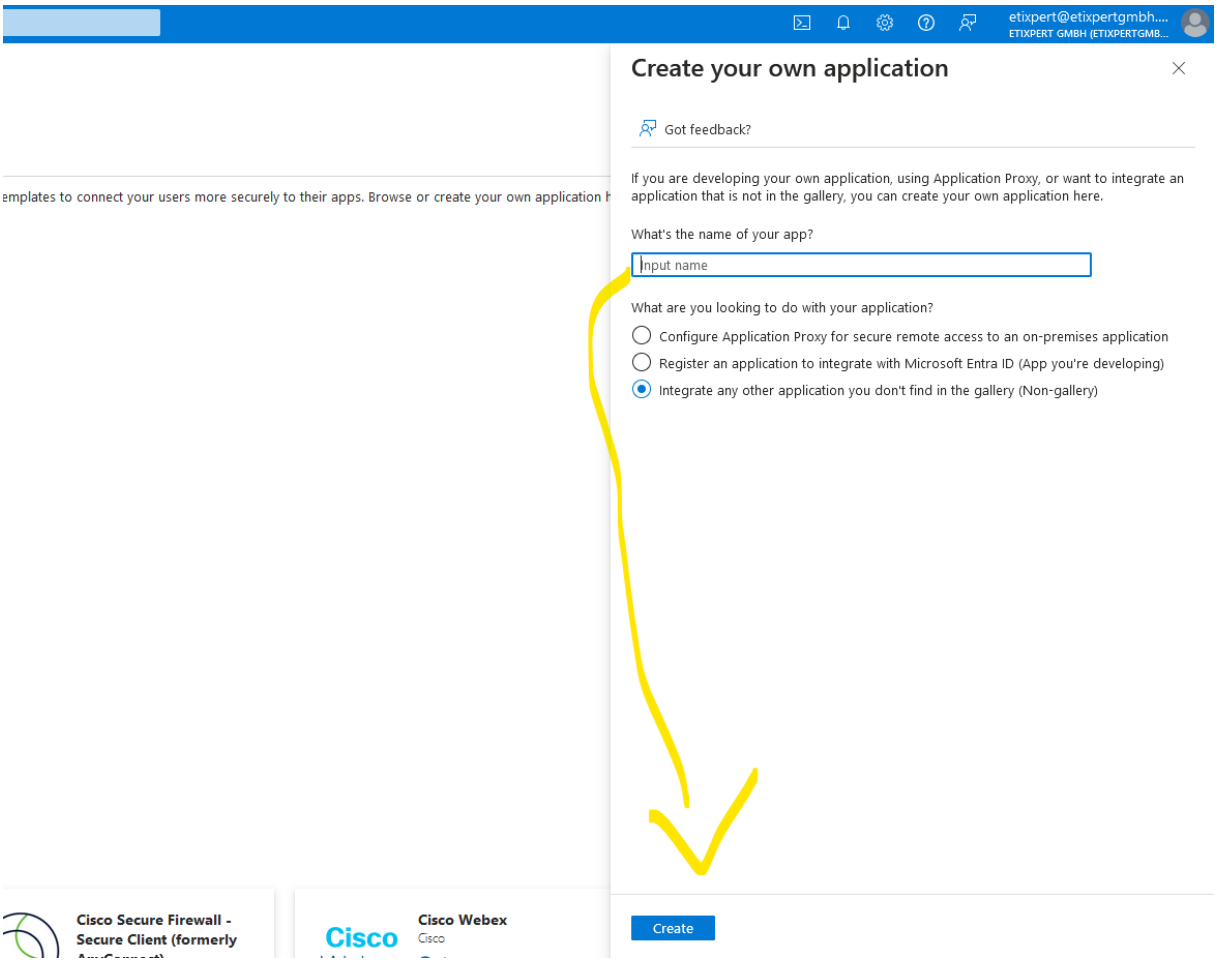
Select menu item *New application*:



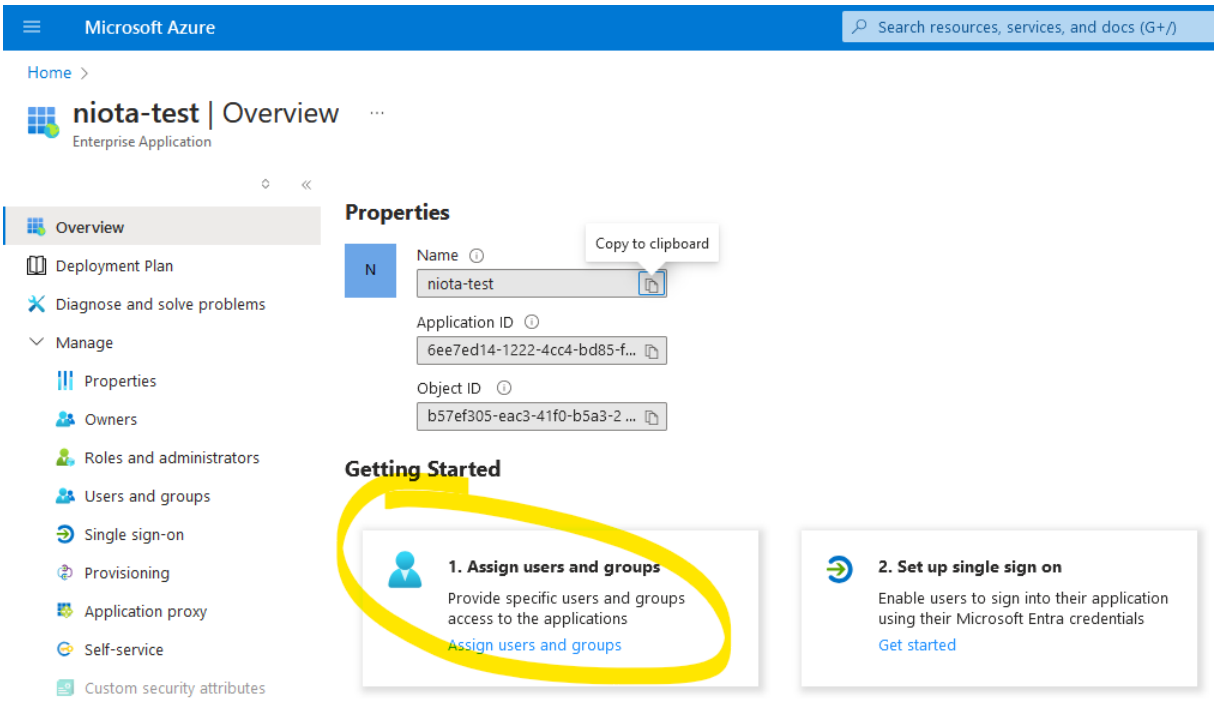
Chose menu item *Create your own application:*



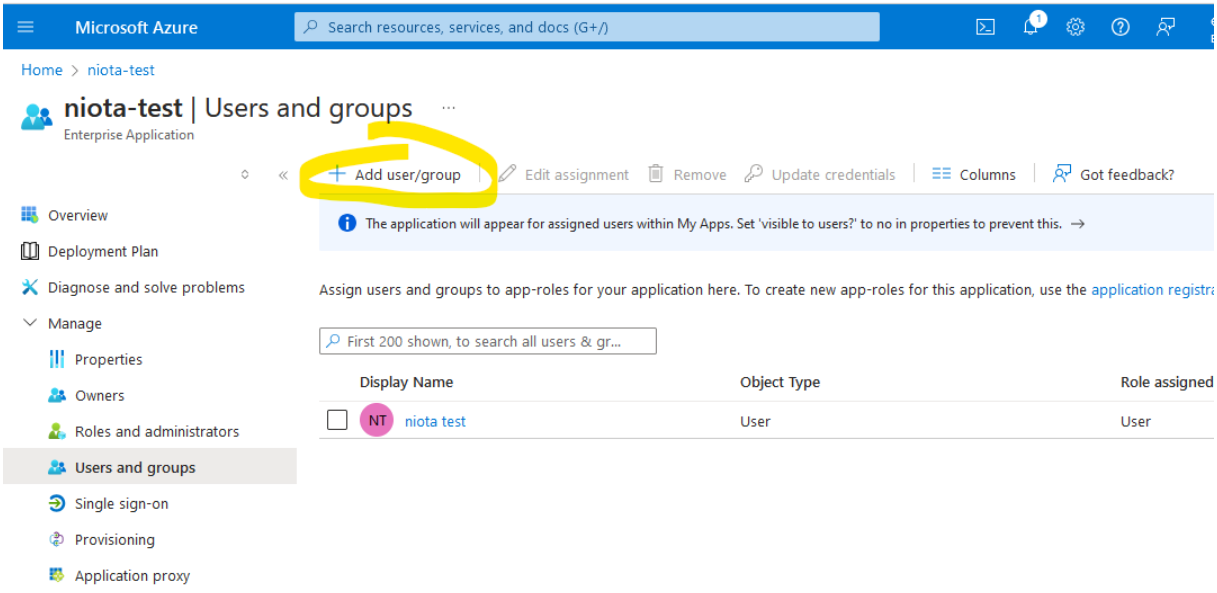
Enter the name of the application then click button *Create* at the bottom of the page:



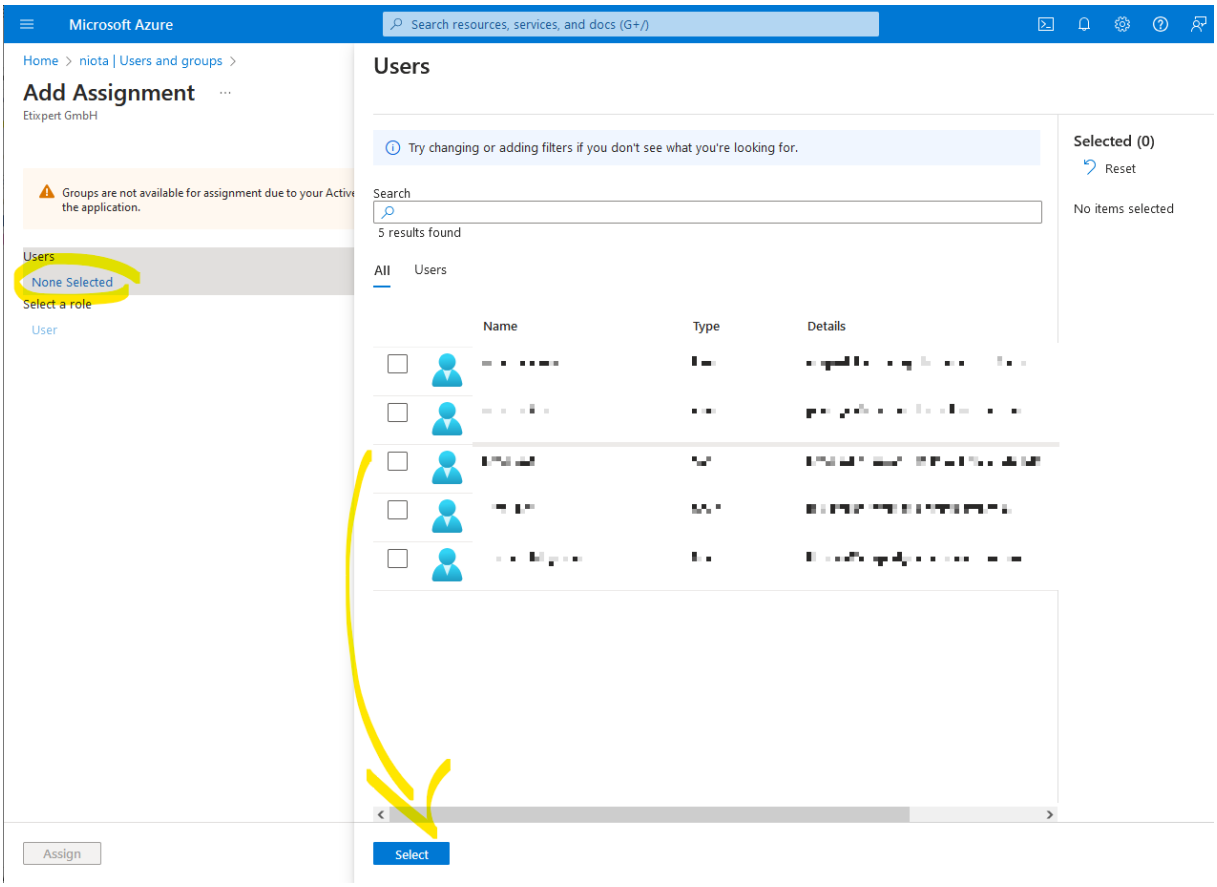
Then select your application and define users who can use SAML authentication. Select menu item *Assign users and groups.*



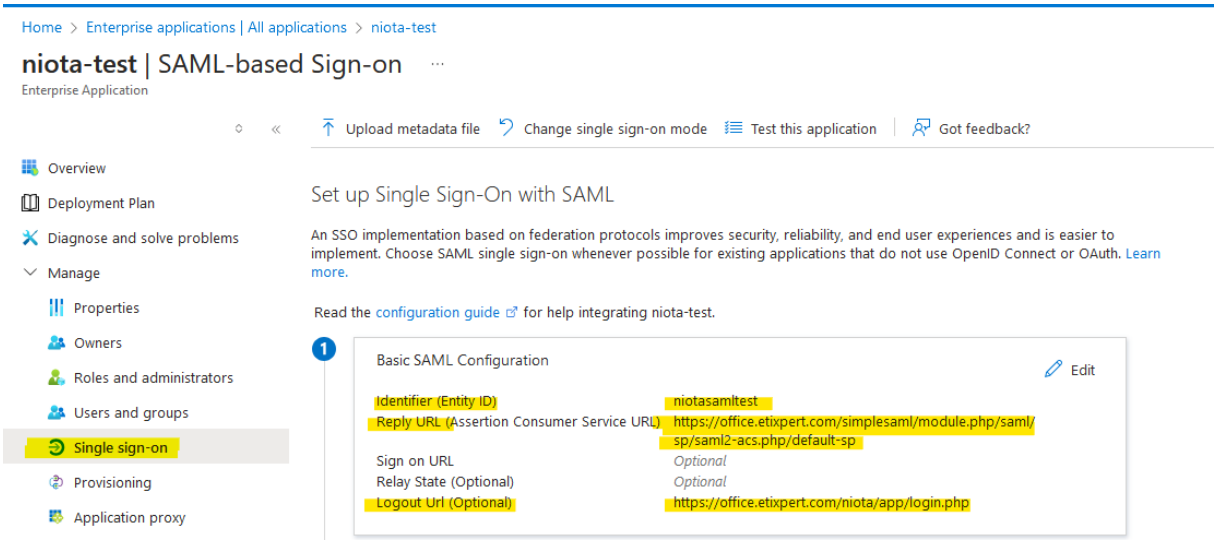
Click item *Add user/group* to assign a user to the application



Click button *None selected* then select users from the list and submit the selection – see screenshot below:



Configure SAML in Azure
Select menu item *Single sign-on* from the left menu



Make the following configuration steps and fill the parameters:

- **Identifier (Entry ID)** – a unique identifier of the application. Entered value will be used in samlphp configuration.
- **Reply URL (Assertion Consumer Service URL)** - URL of the saml service.
  Sample URL:
  `https://office.etixpert.com/simplesaml/module.php/saml/sp/saml2-acs.php/default-sp`
  The following part of the URL is fixed: „`/simplesaml/module.php/saml/sp/saml2-`

acs.php/default-sp".
note: niota.etixpert.com/simplesaml part of the URL refers to the alias set in Apache's configuration file.

- **Logout Url (Optional)** - the page has to be loaded after logging out of Azure by using SAML in niota software. This must be the login page of niota application. Example URL: `https://office.etixpert.com/niota/app/login.php`

simplesamlphp configuration

config.php

Open simplesamlphp configuration file config.php, which can be found in niota installation folder: `c:\Apache24x64\htdocs\niota\app\vendor\simplesamlphp\simplesamlphp\config\config.php`

This is the general configuration file of simplesamlphp library. These settings will be used during SAML authentication.

Define setting baseurlpath according to your environment:

```
  * baseurlpath is a *URL path* (not a filesystem path).
     * A valid format for 'baseurlpath' is:
     * [(http|https)://(hostname|fqdn)[:port]]/[path/to/simplesaml/]
     *
     * The full url format is useful if your SimpleSAMLphp setup is hosted behind
     * a reverse proxy. In that case you can specify the external url here.
     * Specifying the full URL including https: will let SimpleSAMLphp know
     * that it runs on HTTPS even if the backend server is plain HTTP.
     *
     * Please note that SimpleSAMLphp will then redirect all queries to the
     * external url, no matter where you come from (direct access or via the
     * reverse proxy).
     */
    'baseurlpath' => 'https://office.etixpert.com/simplesaml/',
```

Note: URL part `/simplesaml/` refers to the alias set in Apache's configuration file (see at Apache configuration).

If path of your niota installation differs then the ones below, then edit the following path definitions according to your installation details:

```
/*
     * The following settings are *filesystem paths* which define where
     * SimpleSAMLphp can find or write the following things:
     * - 'cachedir': Where SimpleSAMLphp can write its cache.
     * - 'loggingdir': Where to write logs. MUST be set to NULL when using a
logging
     *                 handler other than `file`.
     * - 'datadir': Storage of general data.
     * - 'tempdir': Saving temporary files. SimpleSAMLphp will attempt to create
     *   this directory if it doesn't exist. DEPRECATED - replaced by cachedir.
     * When specified as a relative path, this is relative to the SimpleSAMLphp
     * root directory.
     */
    'cachedir' =>
'c:/Apache24x64/htdocs/niota/app/vendor/simplesamlphp/simplesamlphp/cache/',
    'loggingdir' => 'c:/Apache24x64/htdocs/niota/app/log/',
    'datadir' =>
'c:/Apache24x64/htdocs/niota/app/vendor/simplesamlphp/simplesamlphp/data/',
    'tempdir' =>
'c:/Apache24x64/htdocs/niota/app/vendor/simplesamlphp/simplesamlphp/tmp/',
```

Certification directory also must be set:

```
/*
     * Certificate and key material can be loaded from different possible
     * locations. Currently two locations are supported, the local filesystem
     * and the database via pdo using the global database configuration. Locations
```

```
     * are specified by a URL-link prefix before the file name/path or database
     * identifier.
     */

    /* To load a certificate or key from the filesystem, it should be specified
     * as 'file://<name>' where <name> is either a relative filename or a fully
     * qualified path to a file containing the certificate or key in PEM
     * format, such as 'cert.pem' or '/path/to/cert.pem'. If the path is
     * relative, it will be searched for in the directory defined by the
     * 'certdir' parameter below. When 'certdir' is specified as a relative
     * path, it will be interpreted as relative to the SimpleSAMLphp root
     * directory. Note that locations with no prefix included will be treated
     * as file locations.
     */
    'certdir' =>
'c:/Apache24x64/htdocs/niota/app/vendor/simplesamlphp/simplesamlphp/cert/',
```

Technical support information:

```
/*
     * Some information about the technical persons running this installation.
     * The email address will be used as the recipient address for error reports,
and
     * also as the technical contact in generated metadata.
     */
    'technicalcontact_name' => 'Administrator',
    'technicalcontact_email' => 'support@etixpert.com',
```

Define password for the administrative surface of simplesamlphp. You must enter the surface later to do some further configuration steps.

```
/*
     * This password must be kept secret, and modified from the default value 123.
     * This password will give access to the installation page of SimpleSAMLphp
with
     * metadata listing and diagnostics pages.
     * You can also put a hash here; run "bin/pwgen.php" to generate one.
     */
    'auth.adminpassword' => 'my-secret-password',
```

Set domain as trusted where niota is running:

```
/*
     * Array of domains that are allowed when generating links or redirects
     * to URLs. SimpleSAMLphp will use this option to determine whether to
     * to consider a given URL valid or not, but you should always validate
     * URLs obtained from the input on your own (i.e. ReturnTo or RelayState
     * parameters obtained from the $_REQUEST array).
     *
     * SimpleSAMLphp will automatically add your own domain (either by checking
     * it dynamically, or by using the domain defined in the 'baseurlpath'
     * directive, the latter having precedence) to the list of trusted domains,
     * in case this option is NOT set to NULL. In that case, you are explicitly
     * telling SimpleSAMLphp to verify URLs.
     *
     * Set to an empty array to disallow ALL redirects or links pointing to
     * an external URL other than your own domain. This is the default behaviour.
     *
     * Set to NULL to disable checking of URLs. DO NOT DO THIS UNLESS YOU KNOW
     * WHAT YOU ARE DOING!
     *
     * Example:
     *   'trusted.url.domains' => ['sp.example.com', 'app.example.com'],
```

```
     */
    'trusted.url.domains' => ['office.etixpert.com'],
```

Debug flags optionally:

```
'debug' => [
        'saml' => false,
        'backtraces' => FALSE,
        'validatexml' => false,
    ],
```

Set log level:

```
/*
     * Define the minimum log level to log. Available levels:
     * - SimpleSAML\Logger::ERR     No statistics, only errors
     * - SimpleSAML\Logger::WARNING No statistics, only warnings/errors
     * - SimpleSAML\Logger::NOTICE  Statistics and errors
     * - SimpleSAML\Logger::INFO    Verbose logs
     * - SimpleSAML\Logger::DEBUG   Full debug logs - not recommended for
production
     *
     * Choose logging handler.
     *
     * Options: [syslog,file,errorlog,stderr]
     *
     * If you set the handler to 'file', the directory specified in loggingdir
above
     * must exist and be writable for SimpleSAMLphp. If set to something else, set
     * loggingdir above to 'null'.
     */
    'logging.level' => SimpleSAML\Logger::INFO,
```

If you use proxy url this must be defined:

```
/**********************
     | PROXY CONFIGURATION |
     **********************/

    /*
     * Proxy to use for retrieving URLs.
     *
     * Example:
     *   'proxy' => 'tcp://proxy.example.com:5100'
     */
    'proxy' => 'tcp://office.etixpert.com',
```

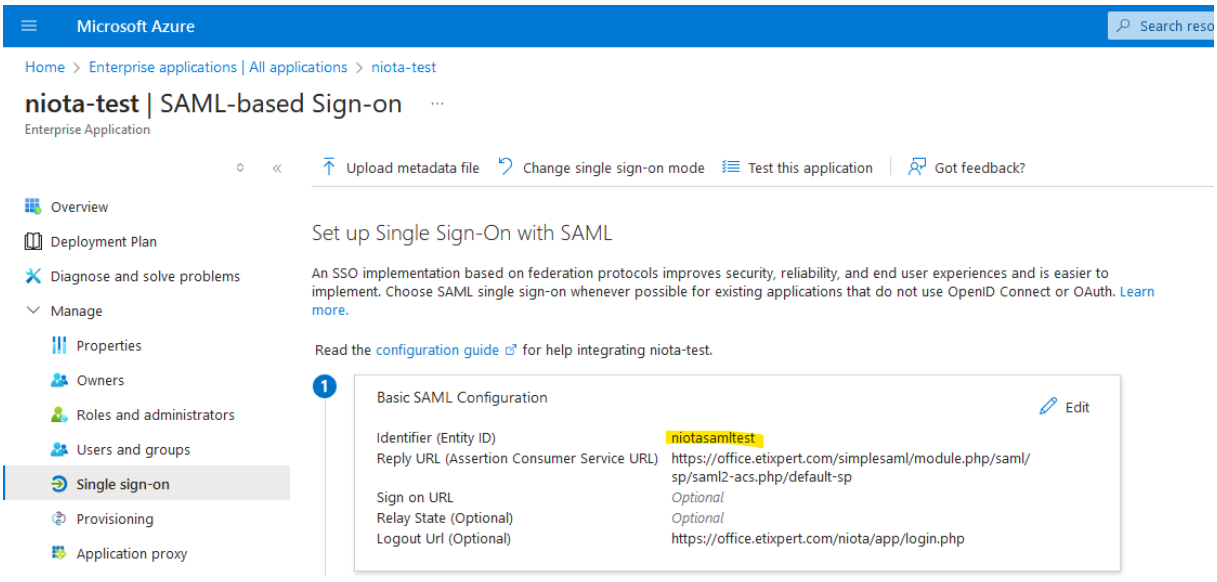
authsources.php

This is the configuration file of the service provider which is used during the SAML authentication process.

Required settings are placed inside section: `default-sp.`

Identifier (Entity ID) defined in Azure surface must be set here in setting: entityID.

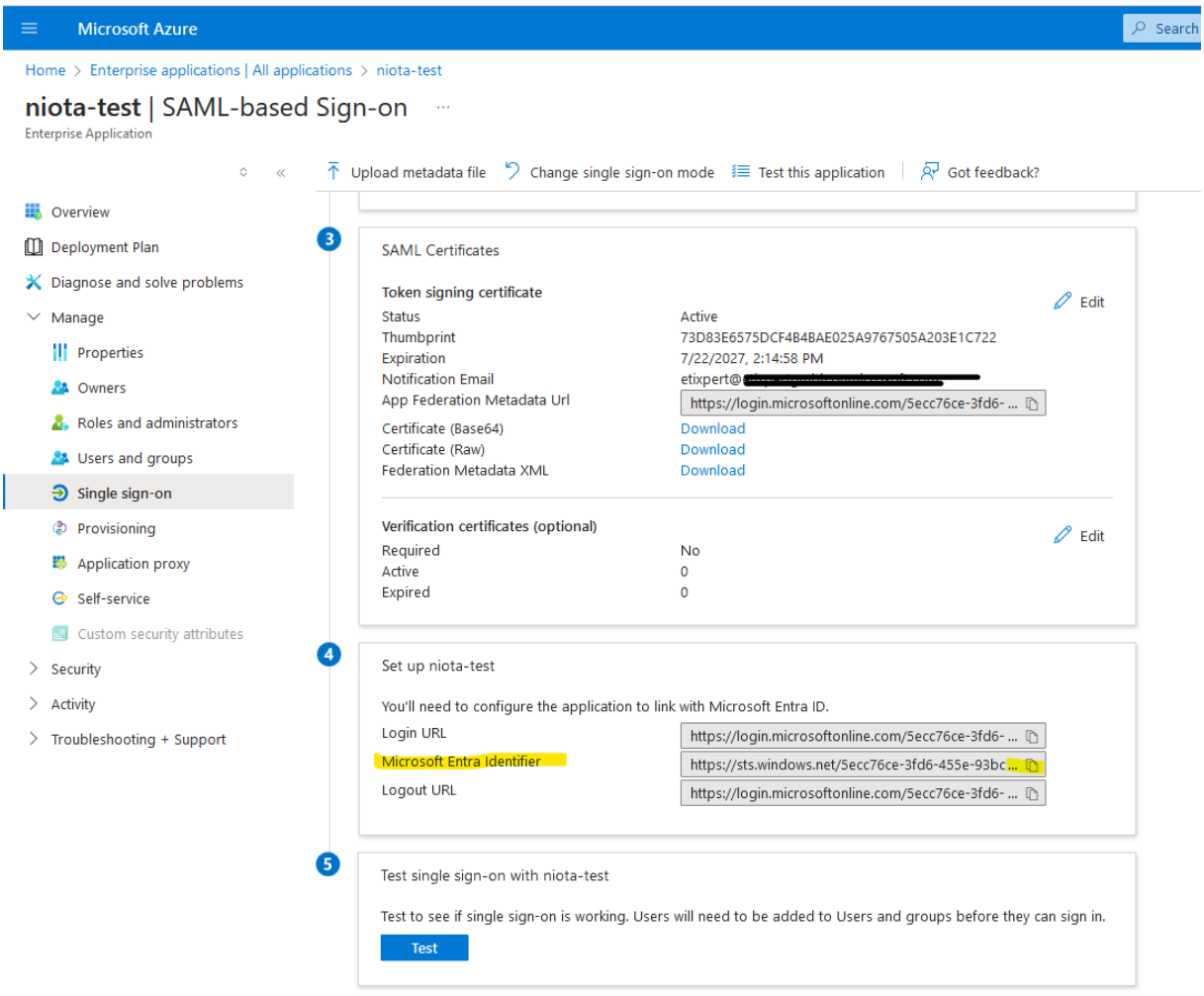
```
  // The entity ID of this SP.
  'entityID' => 'niotasamltest',
```



Define URL of the SAML identity provider.

```
// The entity ID of the IdP this SP should contact.
// Can be NULL/unset, in which case the user will be shown a list of available IdPs.
'idp' => 'https://sts.windows.net/5ecc76ce-3fd6-455e-93bc-82d4dc5bc78f/',
```

You can find the value to enter on Azure surface at menu item *Microsoft Entra Identifier*:



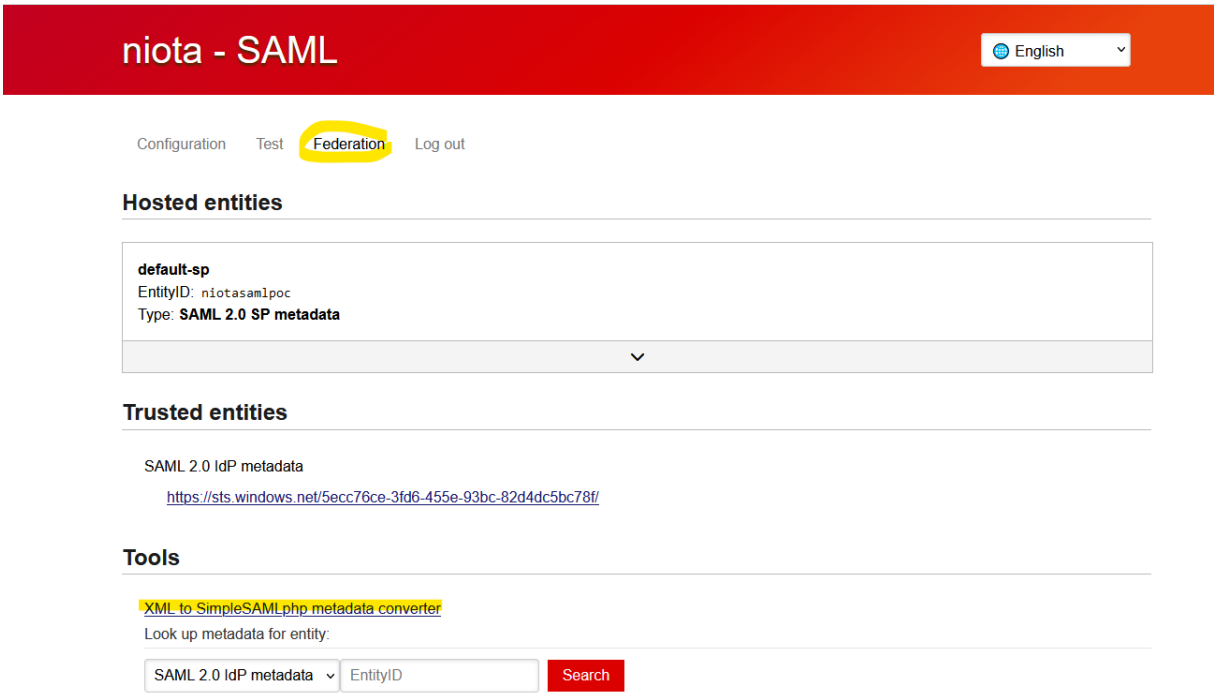
If you use porxy you must set this flag to true. Otherwise set it as false:

```
/*
         * If SP behind the SimpleSAMLphp in IdP/SP proxy mode requests
         * AuthnContextClassRef, decide whether the AuthnContextClassRef will be
         * processed by the IdP/SP proxy or if it will be passed to the original
         * IdP in front of the IdP/SP proxy.
         */
        'proxymode.passAuthnContextClassRef' => true,
```

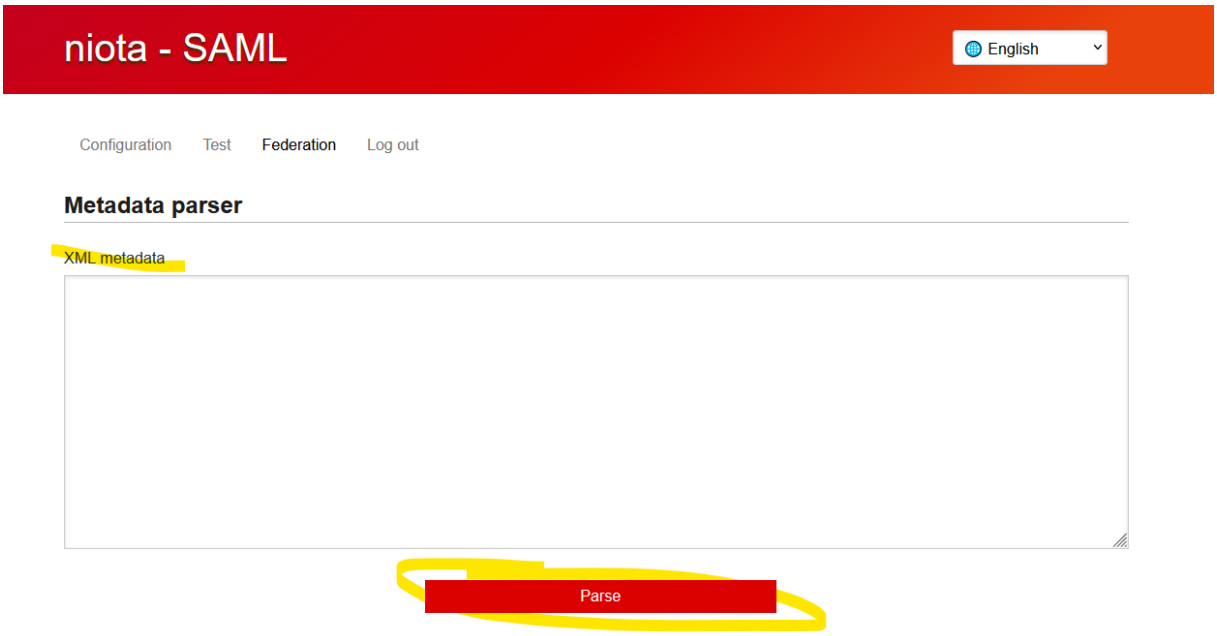
saml20-idp-remote.php

Open admin surface of simplesamlphp. Sample link:
https://office.etixpert.com/simplesaml/admin/index.php

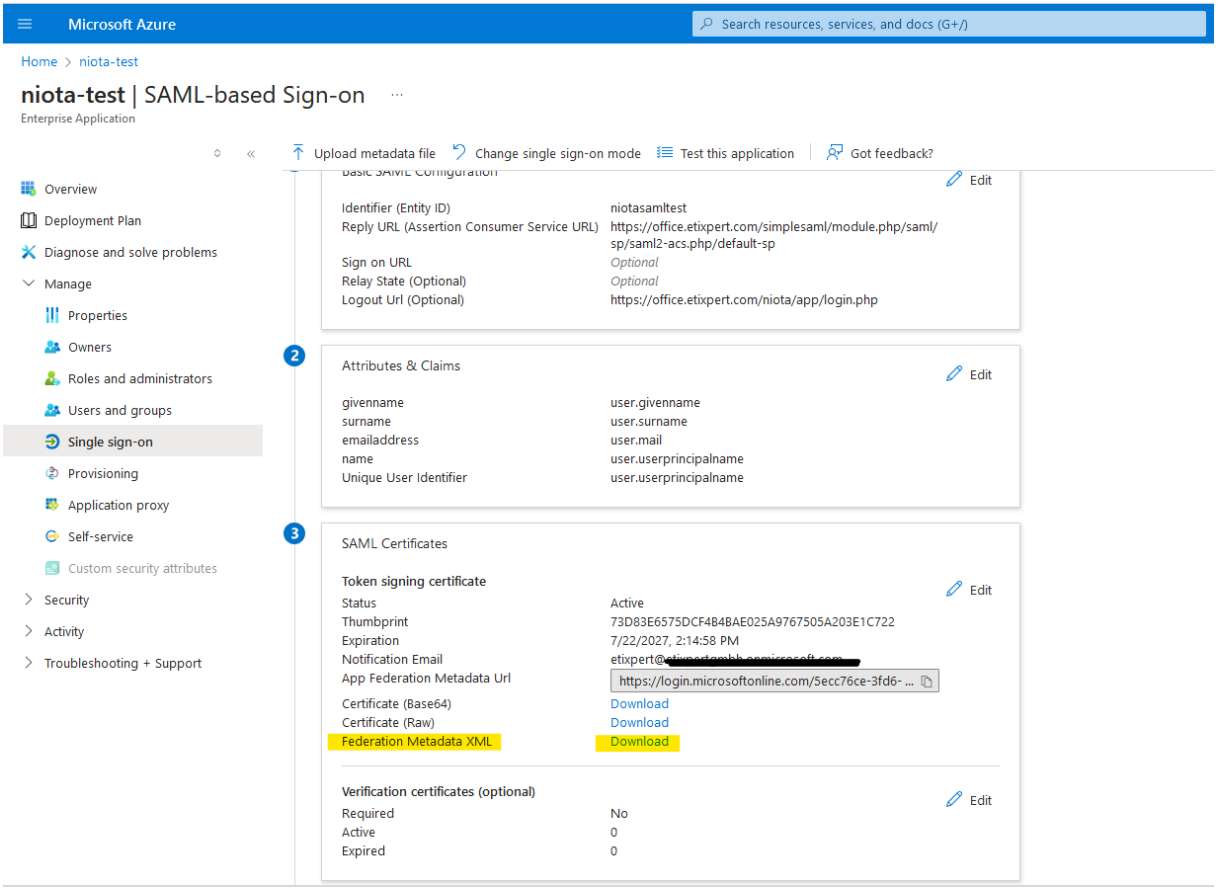
Enter to admin surface by using password (`'auth.adminpassword'`) set in file `config.php`. Select menu item *Federation* then click on link *XML to SimpleSAMLphp metadata converter.*



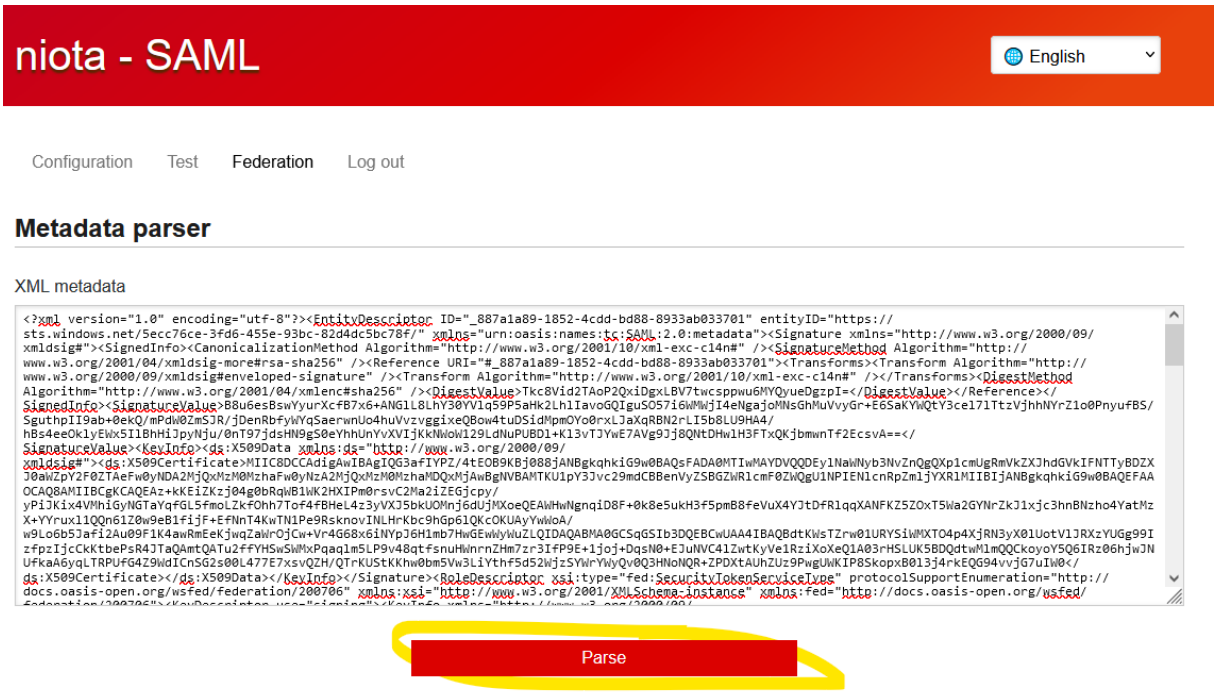
Here you must enter and parse the metadata definition in a later step. As next step you have to download metadata definition in XML format from Azure surface to be able to fill the input field of metadata parser.



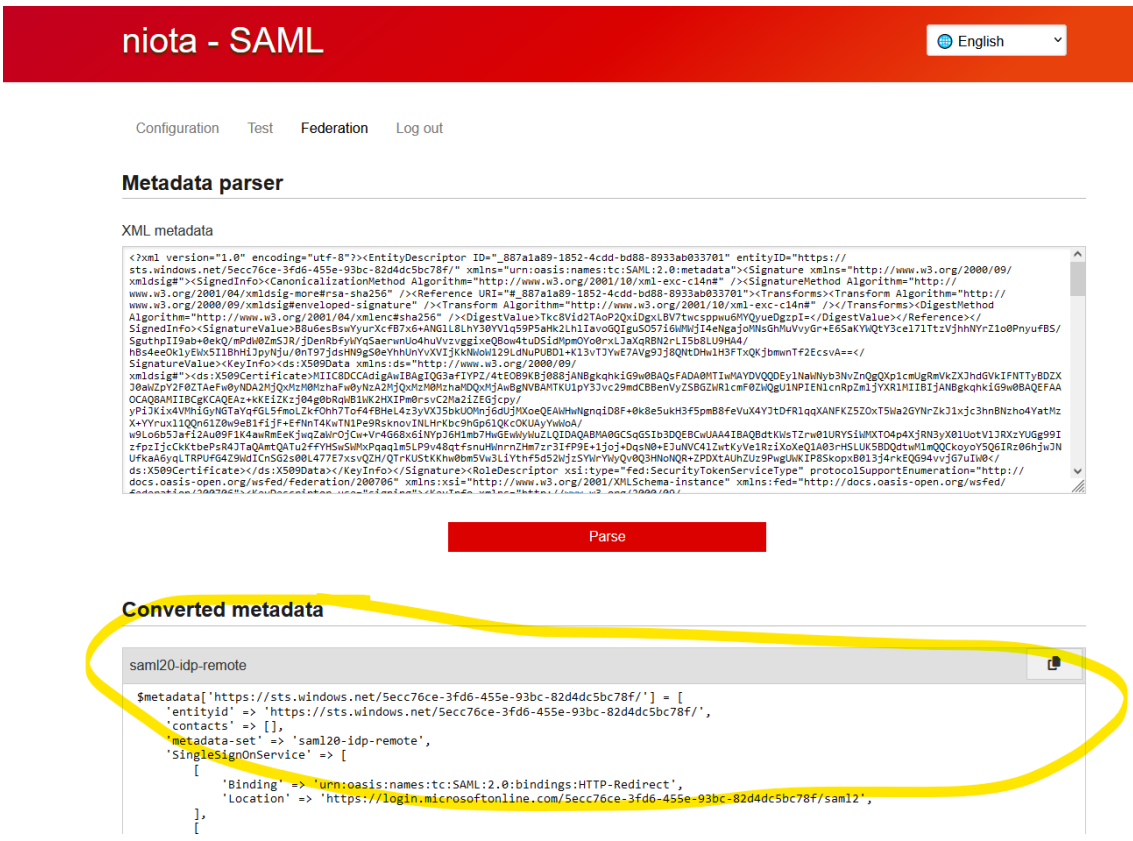
First you must download the metadata XML file from Azure surface:



Copy and paste the content of the downloaded xml file to the Metadata parser textbox and click button Parse.



Converted metadata must be copied and pasted into file `saml20-idp-remote.php`. File path if niota is installed in the default directory:
`c:\Apache24x64\htdocs\niota\app\vendor\simplesamlphp\simplesamlphp\metadata\saml20-idp-remote.php`



saml20-idp-remote.php file content will look like this (not the full content is displayed in the screenshot):

```
<?php

/**
 * SAML 2.0 remote IdP metadata for SimpleSAMLphp.
 *
 * Remember to remove the IdPs you don't use from this file.
 *
 * See: https://simplesamlphp.org/docs/stable/simplesamlphp-reference-idp-remote
 */
$metadata['https://sts.windows.net/5ecc76ce-3fd6-455e-93bc-82d4dc5bc78f/'] = [
    'entityid' => 'https://sts.windows.net/5ecc76ce-3fd6-455e-93bc-82d4dc5bc78f/',
    'contacts' => [],
    'metadata-set' => 'saml20-idp-remote',
    'SingleSignOnService' => [
        [
            'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect',
...
```

After these configuration steps test the authentication process:

If connection test is successful, then go to niota configuration.

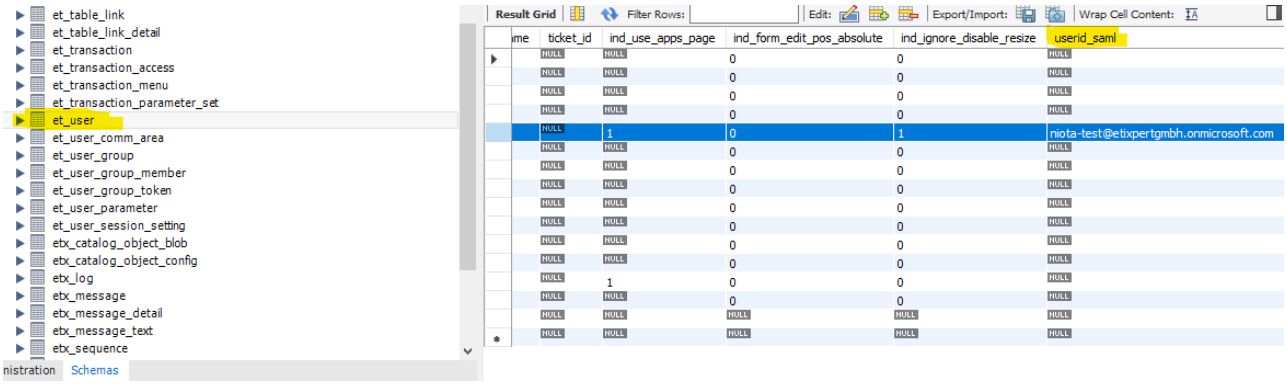
niota configuration

Open configuration file `system_defaults.php`.

SAML authentication related constants must be copied to the client's `niota/conf/system_defaults.php` file from file `niota/app/conf/system_defaults.php` in the case of an installation update. Following section must exist in file `system_defaults.php`:

```
// --- SAML authentication
// settings affects SAML authentication behavior. SAML authentication method can
be used only on system level.
// SAML_AUTHENTICATION_ENABLE - enables SAML authentication for users and displays
related GUI items. Accepted values are: true/false
// SAML_AUTHENTICATION_IDPNAME - identity provider of SAML process.
//                               Accepted values: SAML_IDP_NAME_AZURE - stands for
'Azure', currently the only supported IDP.
define('SAML_AUTHENTICATION_ENABLE',                    true);
define('SAML_AUTHENTICATION_IDPNAME',                   SAML_IDP_NAME_AZURE);
define('SAML_AUTHENTICATION_FORCESAMLLOGOUT',               true);
```

Define belonging SAML id for users in table `et_user`. SAML id is stored in column `userid_saml`. Field references to the user id used for the SAML authentication process. Without this step niota-user id cannot match with SAML-id and user cannot enter niota system.



After these steps you can use SAML authentication on niota surface: