

Authentic - Administrator Guide

Pierre Cros
pcros@entrouvert.com

Copyright: Copyright © 2005 Entr'ouvert

Contents	2
1 Overview	5
1.1 What is identity federation ?	5
1.2 What are the benefits of identity federation ?	5
1.3 Different federation identity protocols and standards	6
1.3.1 SAML	6
1.3.2 Liberty ID-FF	6
1.3.3 Shibboleth	7
1.3.4 WS-Federation	7
1.3.5 Liberty ID-WSF	7
2 How to get and install Authentic	9
2.1 Installation under Debian Sarge	9
2.1.1 Package Installation	9
2.1.2 Apache Configuration	10
2.2 Installation with another Linux distribution	11
2.3 Installation under Windows	11
3 Basic Authentic configuration	12
3.1 Administrator creation	12
3.2 Basic configuration of the Identity Provider	14
3.2.1 Public and private keys creation	14
3.2.2 Identity Provider configuration	15
3.2.3 Saving the metadata file	16

4	Service Provider installation	17
4.1	Service Provider Example: Candle	17
4.1.1	Authentic-like installation	17
4.1.2	Public and private keys creation	18
4.1.3	Service Provider creation	18
4.1.4	Saving the metadata file	19
4.2	Declaring Authentic as Identity Provider on Candle	19
4.3	Declaring Candle as Service Provider on Authentic	20
4.4	Service Provider example: Spip	21
4.4.1	Authentic-like installation	21
4.4.2	Public and private keys creation	21
4.4.3	Service Provider creation	22
4.4.4	Saving the metadata file	22
4.5	Declaring Authentic as Spip Identity Provider	22
4.6	Declaring Spip as Service Provider on Authentic	22
5	Authentic use and settings	23
5.1	Creating and modifying users	23
5.1.1	Adding a user manually	24
5.1.2	Import identities from a CSV file	25
5.1.3	Using a LDAP directory	26
5.1.4	Allow the users to create their identities	27
5.1.5	Modifying a user datas	27
5.2	Identity parameters	27
5.2.1	Identity Options	27
5.2.2	Identity Storage	28
5.2.3	Passwords	28
5.3	Customisation parameters	28
5.3.1	Language	28
5.3.2	Themes	28
5.3.3	Templates	29
5.3.4	Pages publiques	29
5.3.5	Email	29
5.3.6	Cancel button	29
5.4	Logs	30

5.5	Debug Settings	31
5.5.1	Options de Debug	31
5.5.2	Debug Panel	32
5.5.3	Declaring a Authentic bug	32
6	Advanced Settings	33
6.1	Theme customisation	33
6.2	Template Customisation	33
6.3	Public pages customisation	34
6.3.1	Account Management	34
6.3.2	Registration	34
6.3.3	Registration Completed	34
6.3.4	Changing Password	35
6.3.5	Login	35
6.3.6	Lost Password	35
6.3.7	Lost Password Question	35
6.3.8	Lost Password (mailed)	35
6.3.9	Updating Personal Information	35
7	Licenses	36

Authentic is an identity management solution (an Identity Provider) designed for identity federation and Single Sign-On in conformity with [Liberty Alliance](#) norms and standards (ID-FF 1.2 and ID-WSF). It uses the [Lasso](#) library which is certified by the [Liberty Alliance](#) consortium. [Lasso](#) and Authentic are released under the terms of the [GNU/GPL license](#).

1.1 What is identity federation ?

Identity federation is the combination of technological and business needs to enable exchanges between different networks and domains in a secure and reliable manner. The main purpose of federation is to share identity information across heterogeneous systems and identity platforms.

An identity federation based system enables the users connexion with a single username and password (or any other authentication mean) instead of having one for each service. This username and this password are typed only once at the time of connection to the first service. The user is then automatically authenticated on all the services sharing the federated identity.

We can compare a federated identity with a passport used to prove your identity and to allow you to travel from one country to the other.

1.2 What are the benefits of identity federation ?

There can be a lot of advantages :

- to secure the accesses to your applications on all the networks, public and private (and particularly concerning a possible extension of extranet transactions);
- to simplify the access to your applications using Single Sign-On (simplification for the users and the administrators);

- to reduce the costs (costs of helpdesk in particular, costs related to the management of a partner's or a customer's users);
- some federation protocols guarantee, in addition, the respect of the user's private life (the user controls the exchanged data, no unique username is exchanged).

1.3 Different federation identity protocols and standards

There are a lot of them, often linked with each other. [SAML](#) (Security Assertion Markup Language) is for example the common base of ID-FF ('Liberty Alliance'_ Identity Federation Framework) and of [Shibboleth](#). In addition a new layer came on top of the pile of existing standards : the "identity services". They are network distributed services which work with a user controlled identity. This controlled identity services defines which specific information (or attributes) can be used by those services to bring a customized and adapted answer. This allows an increased customisation of the services, intelligent transactions based on identity information.

1.3.1 SAML

the Security Assertion Markup Language (SAML [_](#)) designed by [OASIS](#) was the first standard allowing the exchange of protected assertion. Most of the big companies offering access management solutions took part in its development and it was explicitly conceived for the business-to-business relations.

[SAML](#) 1.1 undoubtedly proposes less superfluous functionalities than other standards. It is a clean, simple and concise protocol. However this concision which makes it relatively easy to implement is a handicap when it comes to using [SAML](#) within another environment, business-to-consumer or business-to-employee for example. [SAML](#) 1.1 lack some functionalities concerning the confidentiality, the safety and mobile clients support. The release of [SAML](#) 2.0 should largely cure it.

[SAML](#) 2.0 is a form of convergence between [SAML](#) 1.1, Liberty ID-FF 1.2 and [Shibboleth](#). The [OASIS](#) will was to take the best in each of these protocols and to include it in a single and coherent framework.

1.3.2 Liberty ID-FF

Liberty Identity Federation Framework (ID-FF 1.2 and its predecessor ID-FF 1.1) was worked out by the [Liberty Alliance](#) consortium, founded mid 2001 by Sun, and joined by several hundreds of companies (France Telecom, Vodafone, VeriSign, Mastercard, etc).

ID-FF is based on [SAML](#) and allows more complex deployment scenarios. It introduces mainly:

- the user control on the operated federations;
- true Single Sign-On and Single Logout;

- a true anonymisation (no unique username circulating between the Service Providers and the Identity Providers);
- an authentication context (allowing to provide information on the authentication used itself and on what surrounds it, like the inscription procedure);
- the metadata exchange.

Designed for companies, ID-FF allows to couple the requirements for a strong authentication with the respect of the users private life, this is why the ADAÉ very strongly encourages its use within the French administrations.

1.3.3 Shibboleth

[Shibboleth](#) is compliant with [SAML 2.0](#). [Shibboleth](#) is an [Internet2](#) project. [Internet2](#) is a consortium leaded by american universities working in partnership with industry and the government. It is an open source implementation (Apache licence) authorising the inter-institutional sharing of web resources subjected to an access control.

[Shibboleth](#) provides a standardized gateway between the existing authentication on campuses and resources suppliers of all kind. It includes metadata exchange and privacy rules allowing agreements between small groups of partners. It is mainly used in higher education field.

1.3.4 WS-Federation

WS-federation Microsoft, IBM, and VeriSign work on a set of specifications (called “WS-Security roadmap” or “WS -*) for their next generation platform of Web services.

[WS-federation](#) is one of these specifications, it defines a model for the federation and the functions related to the identity.

[WS-federation](#) is designed for companies, the business-to-business and business-to-employee relations. The use of privacy is optional there and it misses the multi-client support, which makes it not very suitable with the business-to-consumer environment at the moment.

Because it is relatively recent [WS-Federation](#) was not tested and deployed as much as other protocols, it is thus advisable to use it with caution.

1.3.5 Liberty ID-WSF

the Identity-based Web Services Framework (ID-WSF) of [Liberty Alliance](#) is on top of the pile of the federation protocols. The specificity of ID-WSF is the identity service discovery which allows attribute sharing under the user control.

ID-WSF gathers the following elements:

- permission based attribute sharing (the user determines which attributes can be published and who can use them);

- identity service discovery (determines how the Service Providers learn where to find identity information);
- interaction service (allows the Service Providers and Identity Providers to interact in real time with the user to obtain its assent and the necessary authorisations);
- Extended client support (gives the option for client devices to host their own identity service or act as an Identity Provider);
- Identity service templates (a reusable mechanism for building new identity services that can leverage the web services framework);
- usage directives (which provide a means for including privacy directives in the attribute exchange);

ID-WSF is well-suited for business-to-business and business-to-consumer deployments where it is crucial to share attribute information in a privacy-oriented manner. Relying parties in the transaction will be able to search and discover identity information from distributed identity services that the end-user has registered. Policies related to attribute release can be defined ahead of time or on the fly via an interaction service that can communicate with the end user to obtain permissions.

CHAPTER 2

How to get and install Authentic

2.1 Installation under **Debian Sarge**

To work correctly Authentic relies on :

- [Apache](#) (1.3 or 2, Apache2 recommended) ;
- [Lasso](#) (0.6.3) ;
- [Quixote](#) (2.0) ;
- [mod_python](#) or [SCGI](#) (SCGI recommended).

2.1.1 Package Installation

You need to add the following line to your `/etc/apt/sources.list`; this will give you access to the repository where Authentic is stored:

```
deb http://deb.entrouvert.org/ sarge main
```

As root type :

```
apt-get update
apt-get install authentic
```

All the required packages are installed.

If you don't want to modify your `sources.list` file, you can manually download and install the required packages with the `dpkg -i` command :

- Authentic and Quixote 2.0 on <http://authentic.labs.libre-entreprise.org/> ;
- Lasso on <http://lasso.entrouvert.org>.

2.1.2 Apache Configuration

You need then to configure Apache to have a Authentic virtual host. The following example file is called authentic and is installed. It works (replacing `www.example.com` by the domain Name you've chosen for Authentic, we'll use `authentic.example.com`) for Apache2 and [SCGI](#). You can find it in `/etc/apache2/sites-enabled` :

```
<VirtualHost *>
  ServerAdmin webmaster@localhost
  ServerName authentic.example.com
  DocumentRoot /usr/share/authentic/web/
  SCGIMount / 127.0.0.1:3002
  <LocationMatch "^/(css|images|js)/.*">
    SCGIHandler off
  </LocationMatch>
  SSLEngine On
  CustomLog /var/log/apache2/authentic-access.log combined
  ErrorLog /var/log/apache2/authentic-error.log
</VirtualHost>
```

To activate Authentic site you need to make a link toward this file from the `/etc/apache2/sites-available/` directory :

```
a2ensite authentic
```

You have to make sure Apache is configured to support SSL as well : check you have the following line in `/etc/apache2/ports.conf` :

```
Listen 443
```

Add it if you don't find it. Add then SSL support in Apache :

```
a2enmod ssl
```

Next it is necessary for [SCGI](#) to be enabled :

```
a2enmod scgi
```

You can then reload [Apache](#) (still as root) :

```
/etc/init.d/apache2 reload
```

Don't forget to modify your `/etc/hosts` file if necessary. Authentic works, the administration interface is reachable : <http://authentic.example.com/admin>.

2.2 Installation with another Linux distribution

We suppose [Apache](#), [SCGI](#) or [mod_python](#) are already installed. You need then to download and install the following sources :

- Lasso <http://lasso.entrouvert.org>;
- Quixote <http://www.mems-exchange.org/software/Quixote/>;
- Authentic <http://authentic.labs.libre-entreprise.org/>.

To install Authentic, uncompress the sources you have downloaded and launch the `setup.py` script :

```
tar xzf authentic*.tar.gz
cd authentic*
python setup.py install
```

You need then to configure [Apache](#) correctly.

To launch Authentic you can type as root in a terminal :

```
authenticctl.py start
```

Note

Note that for security reasons, it is better to have Authentic launched by a dedicated user, this user must have writing right on `/var/lib/authentic`.

Once Authentic is working, the administration interface is : <http://authentic.example.com/admin>.

2.3 Installation under Windows

We did not proceed any installation of Authentic under Windows so far. But as all the required components works with this OS, the installation is possible and we may describe it soon. Don't hesitate to tell us about your attempts.

CHAPTER 3

Basic Authentic configuration

3.1 Administrator creation

We consider you are the Authentic administrator and we are going to help you create your account.

At first, you have to go on the administration interface : <http://authentic.example.com/admin/>.

Figure 3.1: the administration interface when no user has been created yet.

Click on the “Identity Management” tab, then on the “add identity” link.

Figure 3.2: Creation of the first identity, the administrator one

Fill the following fields :

- Name (type your christian name and first name);
- Email (type your Email);
- Roles (choose the only available role “administrator”);
- Username (the user name chosen for the administrator);
- Password (the password chosen for the administrator).

the account is created you can now connect as administrator on the identification page.

Figure 3.3: First connexion

3.2 Basic configuration of the Identity Provider

3.2.1 Public and private keys creation

If you don't hold pem format keys, you need to create them. To create a couple public key/private key with [OpenSSL](#), use the following commands :

```
openssl genrsa -out name-of-the-private-key.pem 2048
```

This command creates the private key in a file named name-of-the-private-key.pem. :

```
openssl rsa -in name-of-the-private-key.pem -pubout \  
-out name-of-the-public-key.pem
```

This command extracts the public key from the private key in a file names name-of-the-public-key.pem.

3.2.2 Identity Provider configuration

Figure 3.4: Identity Provider Configuration

The two first fields are automatically filled, don't play with them unless you know what you are doing.

Fields :

- Provider ID (an username which necessarily is a URL);
- Base URL (All the [Liberty Alliance](#) required URL are located under this root);
- Organisation name (Name of the organisation who manage the Identity Provider);
- Private Key (pem format private key);
- Public Key (pem format public key);
- Identity Provider Introduction, Common Domain (the Identity Provider introduction is a [Liberty Alliance](#) mechanism allowing an Identity Provider, for a particular domain, to create a cookie on the client machine. This is useful when several Identity Providers are associated to a Service Provider : this cookie can associate the Service Providers within a domain with the Identity Provider which delivered the cookie.);

- ID-FF Proxy Support (the proxy ID-FF option allows an identity provider to act as an active proxy between a Service Provider and the final Identity Provider. It is usefull only when several Identity Providers are used).

3.2.3 Saving the metadata file

In this Authentic administration interface you can save the metadata file. this is usefull when it comes to configure a Service Provider. Act as follows :

- click on the "Settings" tab ;
- you see an "Identity Provider Metadata" link. Do a right click and "save the link target as" ;
- choose the file name (for example metadata-authentic.xml) and the place you want to save it.

CHAPTER 4

Service Provider installation

4.1 Service Provider Example: **Candle**

Candle is a [Liberty Alliance](#) Service Provider specifically designed to work with Authentic. You may prefer to install your own Service Provider, it may work without any problem if it is compliant with [Liberty Alliance](#). **Candle** has the advantage of being developed by the Authentic team (the user interface is very similar, beware of confusion) and to be fully operational.

4.1.1 Authentic-like installation

To install **Candle** under [Debian Sarge](#), just type as root :

```
echo 'deb http://deb.entrouvert.org/ sarge-experimental' \  
>> /etc/apt/sources.list
```

This command add the repository which contains all the required packages in your sources.list file.

Still as root type :

```
apt-get update  
apt-get install candle
```

All the required packages are installed.

Concerning other distributions, download the sources on this site <http://lasso.entrouvert.org/links> and follow exactly the same steps as for the Authentic installation ([Installation with another Linux distribution](#)).

Once the software is installed, the **Candle** administration interface is available <http://candle.example.com/ac>

4.1.2 Public and private keys creation

If you don't have pem format keys, you need to create them. To create a couple public key/private key with [OpenSSL](#), use the following commands :

```
openssl genrsa -out name-of-the-private-key.pem 2048
```

This command creates the private key in a file named name-of-the-private-key.pem :

```
openssl rsa -in name-of-the-private-key.pem -pubout \  
-out name-of-the-public-key.pem
```

This command extracts the public key of the private key in a file named name-of-the-public-key.pem

4.1.3 Service Provider creation

Go on the [Candle](#) administration interface <http://candle.example.com/admin>. Click on the "Settings" tab then on the "Service Provider" link.

Figure 4.1: Configuration of Candle

the two first fields are automatically filled don't play with them unless you know what you are doing.

Fields :

- Provider ID (an username which necessarily a URL);
- Base URL (All the [Liberty Alliance](#) required URLs are under this base URL);
- Organisation Name (name of the organisation which manages the identity provider);
- Private Key (pem format private key);
- Public Key (pem format public key);
- Identity Provider Introduction, Common Domain (the Identity Provider Introduction is a [Liberty Alliance](#) mechanism allowing an Identity Provider, for a particular domain, to create a cookie on the client machine. This is useful when several Identity Providers are associated to a Service Provider : this cookie can associate the Service Providers within a domain with the Identity Provider which delivered the cookie.);

4.1.4 Saving the metadata file

In the [Candle](#) administration interface you can save the metadata file. This is usefull when it comes to declare [Candle](#) as Service Provider on Authentic. Proceed as follow:

- click on the “Settings” tab;
- you see a link “ Service Provider Metadata”. Do a right click and “save the link target as”;
- choose the file name (for example metadata-candle.xml) and the place you want to save it.

4.2 Declaring Authentic as Identity Provider on [Candle](#)

On [Candle](#) administration interface, click on the “Settings” tab, then on the “Identity Providers” link . Click again on “New”.

Figure 4.2: Declare Authentic as Candle Identity Provider

Fill the following fields:

- Metadata (Authentic metadata file);
- Public Key (Authentic public key);
- CA Certificate Chain (certificate gathering all the authentication chain to the root CA).

4.3 Declaring **Candle** as Service Provider on Authentic

You need to declare **Candle** as Service Provider linked to the Identity Provider Authentic. In order to do so, go on the Authentic administration interface:

- click on the tab "Settings";
- click on the "Liberty Providers" link;
- click on the "New" link.

Figure 4.3: Declaring a new Service Provider on Authentic

You need to fill the following fields:

- Metadata ([Candle](#) metadata file);
- Public Key ([Candle](#) Public Key);
- CA Certification Chain (Certificate gathering all the authentication chain to the root CA).
- Allow IdP initiated Single Sign On : (Allow Single Sign-On from the Identity Provider and not only from the Service Provider).

4.4 Service Provider example: Spip

Spip is a CMS.

4.4.1 Authentic-like installation

4.4.2 Public and private keys creation

If you don't have pem format keys, you need to create them. To create a couple public key/private key with [OpenSSL](#), use the following commands :

```
openssl genrsa -out name-of-the-private-key.pem 2048
```

This command creates the private key in a file named name-of-the-private-key.pem :

```
openssl rsa -in name-of-the-private-key.pem -pubout \  
-out name-of-the-public-key.pem
```

This command extracts the public key of the private key in a file named name-of-the-public-key.pem

4.4.3 Service Provider creation

4.4.4 Saving the metadata file

4.5 Declaring Authentic as Spip Identity Provider

4.6 Declaring Spip as Service Provider on Authentic

CHAPTER 5

Authentic use and settings

5.1 Creating and modifying users

You have four different ways of adding new users :

- Create them one by one;
- Create a lot of them automatically using a CSV file;
- Collect the informations of an LDAP directory;
- Allowing the users to create their identity themselves.

Clicking on the “Identity Management” tab you see the users list.

Figure 5.1: Users list

In front of each user name, take place four icons allowing the following actions on the user account : see, modify, remove, see the logs.

5.1.1 Adding a user manually

To create users one by one, click on the “Identity Management” tab then on the “Add identity” link.

Figure 5.2: Add an identity

Fill the following fields:

- Name (type les Name and first name de the user);
- Email (type the Email de the user);
- Roles (choose the role “administrator” if you intend to create another administrator or leave it blank to create a normal user);
- Username (the username chosen for the user);
- Password (the Password chosen for the user).

5.1.2 Import identities from a CSV file

Instead of creating many users one by one, you can generate them automatically thanks to a CSV file formatted as follows :

```
Username;Password;Name;Email
```

Click on the “Identity Management” tab then on the “Import identities from CSV file” link. Click on the “Choose File” button and select the CSV file you have prepared.

5.1.3 Using a LDAP directory

You can use the user base of your LDAP (or LDAPs) directory as a data source: All your LDAP users will have their identity on Authentic. The user must have a direct access to the LDAP directory. Once the LDAP directory declaration is done, one of the users has to be set as the administrator (using Authentic interface), or all the users will be able to access the administration interface.

Click on the "Settings" tab, then on the "Identity Storage» link, select LDAP directory in the list, click on the submit button.

Figure 5.3: LDAP directory configuration

Fill the different LDAP parameters:

- LDAP URL (LDAP or LDAPs server URL);
- LDAP Base (root of the LDAP tree);
- LDAP Object Class (Class to which belong the objects "user", Active Directory default is "user");
- LDAP Object Username Attribute (Field which contains the username in the LDAP directory, Active Directory default is "sAMAccountName");
- LDAP Object User Name Attribute (Field which contains the name of the user in the LDAP directory, Active Directory default is "UserName");

- LDAP Object Email Attribute (Field which contains the email in the LDAP directory, Active Directory default is "mail");
- Massive LDAP Directory (Check this to improve performances of a big LDAP directory, Active Directory default is "").

Please be aware the user LDAPREADER must be allowed to "bind" on the directory with his/her username and password. The user LDAPREADER selected to become administrator must be allowed to list LDAP objects.

5.1.4 Allow the users to create their identities

Instead of having the administrator being the only one in charge of identity creation, you can select an option allowing anybody to create its own account from the login page. When this option is selected, a new link stands on the login page. This link allows anybody to reach an identity creation form similar to the one used by the administrator.

To activate this option, click on the "Settings" tab, then on the "Identity Options" link. In the list "Identity Creation", choose "Self-registration" then submit.

5.1.5 Modifying a user datas

To modify a user datas, click on the "Settings" tab, then on the second icon in front of his name. You can then change what has to be.

5.2 Identity parameters

Some parameters exist about the way identities are created. You can set them on the "Settings" tab, under the "Identities" section.

5.2.1 Identity Options

Clicking the "Identity Options" link you can set four elements:

- Identity Creation : defines if the user can create identities themselves or if only the administrator is allowed to. When "Self-registration" is selected a new link stand on the login page allowing anybody to create an account and to receive his password by email.
- Notify Administrators on Registration : defines if the administrator must receive an email when an account is created by a user.
- Use email as username : defines if the email address must be used as the username. When this checkbox is selected the user email address will automatically be used as username, the user can't choose it anymore.

- Welcome email : allow to write a welcome email which will be automatically sent to the new users. The text is up to you but is definitely more useful if it includes the user password. At the place you want to display it in the text simply type "[password]" (without the quotes). You can insert the username as well using "[username]" and the server name using "[hostname]".

5.2.2 Identity Storage

Following the "Identity Storage" link you can choose two different storage types:

- Default storage (files) (stores the identities in separate files in /var/lib/authentic/);
- LDAP directory (allows the use of a LDAP directory, cf. [Using a LDAP directory](#)).

5.2.3 Passwords

Following the "Passwords" link, you can set the following elements:

- Allow the user to change his password;
- Choose to generate automatically the initial password;
- Determine the way a lost password can be retrieved (sending an email to the administrator, receiving it by email automatically, answer a question before receiving it by email);
- Define the biggest and the smallest size allowed for passwords (0 means no limitation).

5.3 Customisation parameters

Some options are available to customise Authentic. They are on the "Settings" tab, under the section "customisation".

5.3.1 Language

Following the "Language" link you can set the interface language.

5.3.2 Themes

Following the "Theme" link, you can select different graphic themes which will change the user interface design. The administration interface always remains the same, it uses the "Default" theme. You will find more details about themes in the advanced settings chapter, [Theme customisation](#) section.

5.3.3 Templates

Following the "Template" link, you access the generic model used to display public pages, and you can modify it. The "Restore default template" button restores the original model. The templates syntax is explained in the advanced settings chapter, [Template customisation](#) section.

5.3.4 Pages publiques

To go a step further in customisation, you can modify each of these pages:

- Account Management
- Registration
- Registration Completed
- Changing Password
- Login
- Lost Password
- Lost Password Question
- Lost Password (mailed)
- Updating Personal Information

How to modify these pages is explained in the advanced settings chapter, [Public pages customisation](#) section.

5.3.5 Email

Following the "Email" link, you can define three different things:

- SMTP server (server used by Authentic to send emails);
- Email Sender (which will appear in the "from" field of emails sent to users);
- Reply-To Address (if you wish the adress used for the users response to be different from the Email Sender address).

5.3.6 Cancel button

Following the "Login Screen" link, you can add to this screen a cancel button. This button allows a redirection towards the Service Provider from which the user came. It happens to be usefull when a user reach the Login screen by mistake.

5.4 Logs

The "Logs" tab give access to some informations about the users actions on the server:

Figure 5.4: Informations about the users actions

the following information are collected

- authentication : an authentication succeeded;
- authentication failure : an authentication failed;
- changed password : a password has been changed;
- changing password page : the page allowing to change password has been hit;
- changing password page (had_errors) : on the page allowing to change password the user provoked errors (not typing twice the same new password, or typing a new password similar to the old one);
- created new identity (username) : an identity has been created by the administrator;
- deleted identity (username) : an identity has been deleted;
- fedterm to (provider ID) : the Service Provider ended a federation;

- internal server error : Authentic had an internal server error;
- login page : the Login Screen has been hit;
- login page, cancel : the cancel button on the Login Screen has been hit;
- login page, proxying to (provider ID) : on the Login Screen, the user asked for being redirected toward another Identity Provider. Authentic becomes a proxy for this provider.
- lost password page : the page allowing to retrieve a lost password has been hit;
- lost password -> email password (username) : a password has been sent to the user by email;
- logout : a simple deconnexion succeeded;
- SLO from (Provider ID) : a global deconnexion initiated by the Service Provider using http (redirect) protocol succeeded;
- SLO/SOAP from (provider ID) : a global deconnexion initiated by the Service Provider using SOAP protocol succeeded;
- SSO from (username) : a Single Sign-On connexion succeeded;
- SSO to (provider ID) : a Single Sign-On connexion initiated by the Service Provider succeeded;
- updated identity : the user modified his/her identity;
- updating personal information : the user modified his/her personal information;
- user created new identity (username) : a user created an identity;

5.5 Debug Settings

To simplify debugging, options are available in the "Settings" tab, in the "Debug" section.

5.5.1 Options de Debug

Following the "Debug Options" link you can set:

- Enable debug panel (the activation of a "Debug" tab which works as explained in the following section);
- Email for Tracebacks (the email address to which error logs are sent);
- Display Exceptions (the display or not of errors and the format, text or html, in which they are displayed).

5.5.2 Debug Panel

If the option "Enable debug panel" is selected, a new tab appears in the administration interface, the "Debug" tab. Click on this tab, then on the "Sessions" link.

Figure 5.5: Sessions list

You get a list of the last sessions with for each of them :

- the connexion address;
- the username;
- the connexion time;
- the last access time.

5.5.3 Declaring a Authentic bug

You can declare a bug or a feature you would like to see implemented on <http://bugs.entrouvert.org>.

Advanced Settings allows you to fully customise Authentic public pages, playing with these parameters, in this order (it matters) : themes, templates, public pages.

6.1 Theme customisation

The theme rules the general style of public pages. Some themes are available and defines (using Cascading Style Sheets) the basic elements of public pages display. You can define your own theme (with your own banner and logo...) if you are familiar enough with CSS.

A theme gathers only two files : desc.xml and name-of-the-style-sheet.css. desc.xml is an XML file containing some basic informations about the theme: its name and version, its label, its description and its author. The style sheet defines the different properties applied to each pages elements. Those two files must be put in the same directory under /usr/share/authentic/themes/. Once the directory and the two files are created the theme becomes available in the administration interface, you can use it.

6.2 Template Customisation

The template defines the structure of all the public pages within a particular theme. It means, apart from the themes, it is possible yet to modify the display of all public pages, modifying the template. Templates are simple text files which contain (among others) some variables, written between square brackets. These variables are substituted in the public page by their values. Here are the variables you should know to modify the template:

FIXME

- [page_title]: the page title displayed in the title bar if it is correctly set in a <title> tag;
- [css]: the style sheet file name;

- [script]: javascripts used by Authentic for some features like sorting lists;
- [onload]: javascript instructions associated to an event. This variable should be set as the value of the <body> tag onload attribute;
- [body_class]: this variable should be used as the value of the <body> tag class attribute. This value is used by the style sheet;
- [title]: the page title, displayed on top of it;
- [org_name]: the organisation name you entered when creating the identity provider;
- [prelude]: empty variable at the moment;
- [breadcrumb]: empty variable at the moment;
- [body]: main content of the page, usually set between the title and the footer;

you can test a variable is not empty with this syntax: [if-any variable-name]...[end]

6.3 Public pages customisation

To go a step further in customisation, you can define for each public page some variations within the selected theme and template. As for the template, you can use some variables written between square brackets.

6.3.1 Account Management

This page is displayed to the user immediately after his/her identification. The variables available for this page are:

- identity_label (the user name) ;
- idp_sso_list (List of the Service Providers the users can connect to) ;
- federations_list (active federations list).

6.3.2 Registration

This page is displayed to the user when he/she creates his/her identity. Only one variable is available for this page:

- register_form (the form filled to create the identity).

6.3.3 Registration Completed

This page is displayed to the user when he/she has validated his/her registration. No variables are available.

6.3.4 Changing Password

This page is displayed to the user when he/she wants to modify his/her password. Only one variable available for this page:

- `change_password_form` (the form filled to change password) ;

6.3.5 Login

This page is the page on which the users must enter his/her username and password. The variables available for this page:

- `login_form` (the form allowing to type the username and the password);
- `authentication_failure` (the message indicating authentication failure).

6.3.6 Lost Password

This page is displayed to the user when he/she tries to retrieve his/her password. The variables available for this page:

- `lost_password_form` (the form used to type the username);
- `behaviour` : this variable is not displayed but contains the kind of behaviour adopted in case of password loss (nothing, emailed password, question asked). this variable must be used with this kind of conditions:
 - `[is behaviour "email_reminder"]` You will receive an email [end];
 - `[is behaviour "dumb_question"]` You will be asked a question [end].

6.3.7 Lost Password Question

This page is displayed to the user when he/she tries to retrieve his/her password and he/she has to answer a question to do so. Only one variable available for this page:

- `lost_password_question_form` (the form asking the question and recording the answer).

6.3.8 Lost Password (mailed)

This page is displayed to the user when he/she tries to retrieve his/her password and it has been sent by email. No variables available.

6.3.9 Updating Personal Information

This page is displayed to the user when he/she updates his/her identity datas. Only one variable available for this page:

- `info_form` (the form displaying user datas and recording modifications).

CHAPTER 7

Licenses

Authentic, [Candle](#) and [Lasso](#) are released under the terms of the [GNU/GPL license](#).