# Ping Identity® Self Service Account Manager Installation and Configuration Guide

Version: 1.1.0

# Ping Identity® Self Service Account Manager Product Documentation

# Table of Contents

# Chapter 1:  Welcome to the Self Service Account Manager

The PingData Self Service Account Manager (SSAM) is a Java web application that enables common user account registration, update, and password changes, with optional integration with PingFederate and PingAccess. Entries are managed in the PingData Directory Server.

Topics include:

SSAM overview

Deployment types

SSAM installation prerequisites

Install SSAM

Configure PingFederate

Configure PingAccess

Configure the PingFederate login template

Log into SSAM

Register a new user

Manage passwords

# SSAM overview

SSAM is a simple user interface to the PingData Directory Server that enables:

- New user account registration

- Existing account login

- Account profile editing

- Account password change or recovery, and change verification through mobile phone number or email

- Login from PingFederate

# Deployment types

SSAM supports two deployment scenarios. The authentication mechanism is chosen during the SSAM installation. If necessary it can be changed later by editing the `application.properties`file.

- LDAP authentication directly against the PingData Directory Server (default). The LDAP authentication uses form-based login and LDAP authentication facilities to perform bind operations.

- Authentication delegated to PingFederate and PingAccess, with data accessed from the Directory Server. In this scenario, Ping is responsible for authentication, with PingAccess acting as a proxy that requires authentication for certain URLs.

# SSAM installation prerequisites

SSAM requires a standard Directory Server installation. Before installing SSAM, make sure that the server that will host SSAM is configured with the following:

- An available HTTPS port.

- When using Java 7, the server's Java configuration must specify at least 256M for its `-XX:PermSize` JVM option in its `config/java.properties` file. (This does not apply to Java 8 or later.)

- An SMTP server configured for email notifications.

Note
The default application settings for the servlet container are sufficient to host a basic SSAM application. For a deployment that will support large numbers of simultaneous users, adjust the servlet container's settings to allocate more resources to SSAM.

Have the connection details for interacting with the Directory Server ready before installing SSAM:

- Host name.

- LDAP connection port.

- Bind DN and password.

- The base DN where user entries are located.

- The host name of an SMTP mail server that will be used for notifications.

## Enable reCAPTCHA

The SSAM installation enables configuring reCAPTCHA for additional verification on the application's registration and password recovery pages. To enable this feature, sign up for reCAPTCHA (from the Google site). The site key and secret key properties will be required to enable reCAPTCHA. They can be added either during or after the SSAM installation.

### During installation

To enable reCAPTCHA during installation, use the `--reCaptchaSiteKey` and `--reCaptchaSecretKey` arguments to provide the site key and secret key provided by the reCAPTCHA site registration.

### After installation

To enable reCAPTCHA after installation, navigate to the Directory Server hosting SSAM and edit the `<server-root>/webapps/ssam-config/application.properties` file. Add the necessary information to the `recaptchaSiteKey` and `recaptchaSecretKey` arguments. This requires restarting the server.

## Sample Directory Server installation

SSAM is supported on version 5.1 of the Directory Server. For detailed information about configuring the server post installation, see the *Ping Identity Directory Server Administration Guide*. The following information is needed during the installation:

- Server hostname

- LDAP port

- Root DN and password

- Base DN, location of user entries

Perform the following steps to install the Directory Server:

1. Download the Directory Server zip distribution, `PingDirectory-<version>.zip`.

2. Unzip the file in any location.

   ```
   $ unzip PingDirectory-<version>.zip
   ```

3. Change to the top level PingDirectory folder.

```
$ cd PingDirectory
```

4. Run the `setup` command.

```
$ ./setup
```

5. Enter **yes** to agree to the license terms.

6. Enter the Directory Manager DN for the Directory Server, or accept the default, (`cn=Directory Manager`). This account has full access privileges.

7. Enter a password for the root user DN, and confirm it.

8. To enable the Platform APIs, enter **yes**.

9. Enter the port to accept connections from HTTPS clients or press **Enter** to accept the default. The default may be different depending on the account privileges of the user installing. This port defines the URL port (such as `https://<hostname>:8443/`).

10. Enter the port to accept connections from LDAP clients, or press **Enter** to accept the default.

11. Type **yes** to enable LDAPS, or press **Enter** to accept the default (no).

12. If enabling LDAPS, enter the port to accept connections, or press **Enter** to accept the default LDAPS port.

13. Type **yes** to enable StartTLS for encrypted communication, or press **Enter** to accept the default (no).

14. Select the certificate option for the server and provide the certificate location.

15. Specify the base DN for the Directory Server repository, for example `dc=company,dc=com`. The SSAM installation requires a base DN.

16. To specify particular addresses on which this server will listen for client connections, enter **yes**, or press **Enter** to accept the default (no).

17. Select an option to populate the database. If the **Leave the database empty** option is selected, an LDIF file with a base entry must be manually created at a later time.

18. If this machine is dedicated to the Directory Server, tune the JVM memory allocation to use the maximum amount of memory the **Aggressive** option). This ensures that communication with the Directory Server is given the maximum amount of memory. Choose the best memory option for the system and press **Enter**.

19. Enter **yes** to automatically prime the database, or press **Enter** to accept the default (no).

20. To start the server after the configuration, press **Enter** for (yes).

21. Review the Setup Summary, and enter an option to accept the configuration, redo it, or cancel.

The Directory Server configuration is displayed and the installation is complete.

# Install SSAM

Use the SSAM `setup.sh` script to install SSAM and configure a Directory Server to host the web application.

The `setup.sh` script will generate `dsconfig` batch scripts in the `resource` directory of the extracted SSAM ZIP directory, and automatically apply them as necessary. When configuring a Directory Server for access, this tool will perform the following:

- Create a SSAM user entry with ACIs that SSAM will use when binding to the Directory Server.

- Update the schema if necessary.

- Configure the server for communicating with the SMTP server, and configure the email one time password mechanism. For all configured changes, see `ssam-deploy.dsconfig` and `ssam-ds.dsconfig` after running this tool.

- Rebuild the indexes. (This will not be performed if the Directory Server contains a large number of entries. The installer will provide the command to rebuild indexes manually and this command will be logged to the installation log file.)

## Install Options

The `ssam-installer` provides options to configure SSAM with a Directory Server, or Directory Server and PingFederate and PingAccess. Unzip the SSAM package (`ssam-<version>-<build>.zip`) and run the following to review command help:

```
$ ./setup.sh --help
```

The following options are required during SSAM installation:

| Required Installation Arguments | Description |
| --- | --- |
| `--serverRoot <directory>` | Absolute or relative path to the Directory Server to host SSAM. |
| `--ldapPort <port>` | LDAP or LDAPS port used to communicate with the Directory Server. |
| `--bindDN <dn>` | The DN of the account used to manage the Directory Server. |
| `--bindPassword <password>` | The DN account password. |
| `--peopleBaseDN <dn>` | The base DN under which SSAM user entries exist, or will reside through new account registration. |
| `--smtpServerHostname <hostname>` | The name of the SNMP host that the Directory Server uses for user notifications. |
| `--smtpSenderEmailAddress` | The email address that the Directory Server uses for as the sender for |

| Required Installation Arguments | Description |
| --- | --- |
| `<address>` | notifications. |

## Sample Installation

The following sample configures a Directory Server for SSAM access and to host the web application, and uses the Directory Server keystore for establishing trust of the server by SSAM:

```
$ ./setup.sh --serverRoot </path/to/ds/install/dir> \
  --ldapPort <636> \
  --useSSL \
  --trustStorePath </path/to/ds/install/dir/config/keystore> \
  --bindDN <cn=Directory Manager> \
  --bindPassword <Password> \
  --peopleBaseDN <ou=People,dc=example,dc=com> \
  --smtpServerHostname <smtp.example.com> \
  --smtpSenderEmailAddress <do-not-reply@example.com>
```

## Ping Configuration Options

If installing SSAM with access through PingFederate and PingAccess, the following options are available to configure the log out URLs:

- `--pingAccessLogoutURL <url>` – The URL used for logging out of PingAccess, for example `https://<hostname>/pa/oidc/logout`. This is required if deploying SSAM with Ping. The full URL must be used.

- `--pingFederateLogoutURL <url>` – The URL used for logging out of PingFederate, for example `https://<hostname>:<PingFederatePort>/ext/logout`. This is required if deploying SSAM with Ping. The full URL must be used.

The SSAM application takes care of calling each URL in the right order.

# Configure PingFederate

Please refer to the PingFederate product documentation for installation and configuration information. This section highlights information and configuration that is specific to using SSAM with PingFederate. When configuring PingFederate:

1. Add the Directory Server backend and define the LDAP settings.

- HOSTNAME: `<ssam.company.com:1389>`

- LDAP TYPE: `PingData Directory Server` or `Generic`

- USER DN: `<cn=Directory Manager>`

- PASSWORD: `<password>`

- LDAP DATASTORE: `<ssam.company.com:1389>`

- SEARCH BASE: `ou=people,dc=example,dc=com`

- SEARCH FILTER: `uid=${username}`

2. Define roles and security based on the PingFederate documentation.

3. Configure the logout redirect for the IDP adapter. The path is `/logout` and the logout redirect is `https://<ping-access>/ssam/user`.

# Configure PingAccess

Install and configure PingAccess according to the product documentation. The following information is needed to configure PingAccess with SSAM:

1. Create an IDP Mapping and Attribute Name (`sub`) to Header Name `PING_USER`.

2. Create a Site instance with the Target set to the URL of the SSAM application.

3. Create an Application instance of type `Web` that uses the SSAM site. Resources protected by this application should include:

  - `/user*`

  - `/updatePassword`

  - `/deleteUser`

# Configure the PingFederate login template

Add the following HTML section to the PingFederate login template:

```
$ vi <PingFed>/server/default/conf/template/html.form.login.template.html
. . .
<div class="ping-input-container">
  <input id="password" type="password" size="36"
    name="$pass" onKeyPress="return postOnReturn(event)"
    placeholder="$templateMessages.getMessage
        ($messageKeyPrefix, "passwordTitle")" />
</div>

<a href="https://<hostname:port>/ssam/recoverPassword">Forgot
password?</a><br>
<a href="https://<hostname:port>/ssam/register">Sign up for an account</a>

#if($enableRememberUsername)
```

Note
Clicking the **Cancel** link on the PingFederate login page can cause "access denied" errors if not configured properly. This can be resolved by either changing the **Cancel** link in the template to point to a certain location (such as the home page), or removing or commenting out the **Cancel** link.
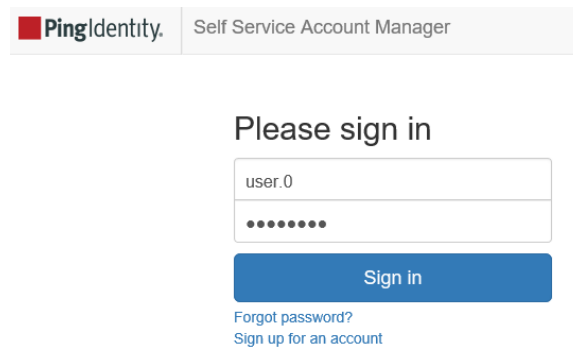
# Log into SSAM

After the Directory Server and SSAM are installed, navigate to `https://<hostname>:<port>/ssam/` and login with a user account from the generated sample data (option in the Directory Server installation). Sample user entries are created using the following format:
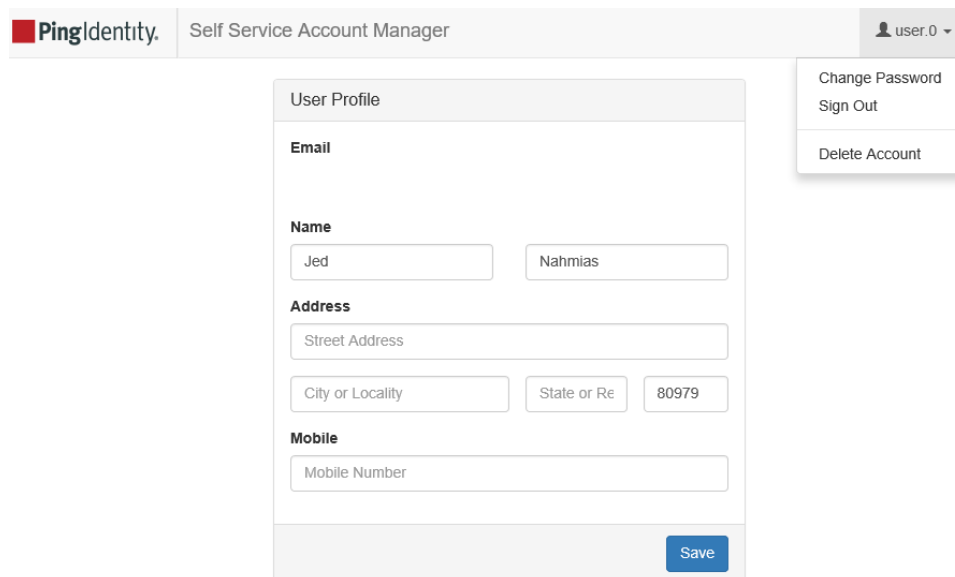
Username: `user.<number>`

Password: `password`

If sample data was not configured during the Directory Server installation, choose Sign up for an account.
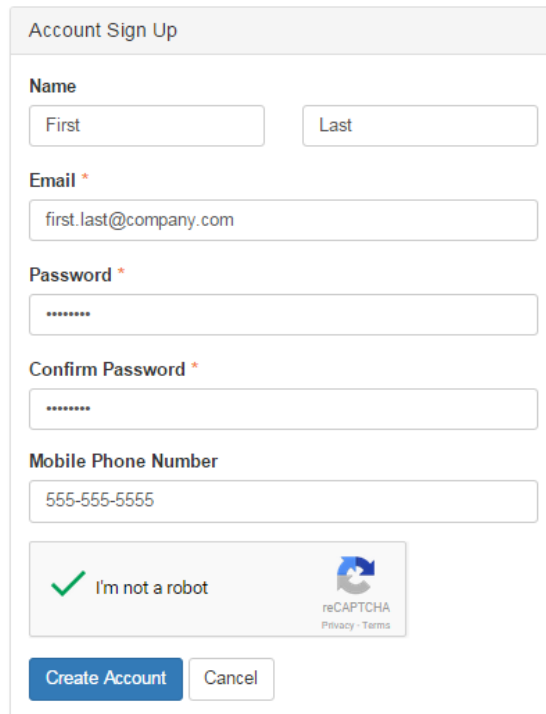


The user's profile data is displayed and can be edited:

# New user registration

SSAM enables new account registration. From the Login page, select **Sign up for an account**. During the account creation process, SSAM will send a registration code to the email address provided by the user.



If the password entered doesn't meet the criteria set in the Directory Server Password Policy, SSAM will display restrictions and text defined in the policy.

# Password management

The **Change Password** drop-down menu option at the top of the User Profile page enables a user to reset the account password. Acceptance criteria for the password is defined in the Directory Server Password Policy.

A user can also reset a password with the **Forgot Password?** option below the Sign in field on the login page. On the Change Password page, the user can enter the email address or phone number associated with the account.



This process uses a temporary, one time password reset token that is sent to the user's email, or another delivery mechanism.

# Chapter 2: Configure SSAM

Password requirements and restrictions that are surfaced in SSAM are defined in the Directory Server and can be configured through password policies. User account information and attributes that are stored are also based on schema definitions in the Directory Server. Once the data requirements are configured for user accounts and passwords, SSAM interface pages can be customized with Velocity templates, including color-scheme, logo, and fields.

Topics Include:

Configure password requirements

Configure the Directory Server schema

Configure SSAM templates

# Configure password requirements

A standard installation of SSAM will use the Directory Server's Default Password Policy for password requirements.

To view the list of password policies configured on the Directory Server, use the `dsconfig` tool. The following example obtains a list of defined password policies in non-interactive mode:

```
$ bin/dsconfig list-password-policies

Password Policy  : Type    : password-    : default-password-
                           : attribute    : storage-scheme
-----------------:---------:-------------:----------------------
Default Password : generic : userpassword : Salted SHA-1
Policy           :         :              :
Root Password    : generic : userpassword : Salted SHA-512
Policy           :         :              :
Secure Password  : generic : userpassword : CRYPT
Policy           :         :              :
```

Use `dsconfig` to view a specific password policy. In this example, view the Default Password Policy that applies to all users for whom no specific policy has been configured.

```
$ bin/dsconfig get-password-policy-prop \
  --policy-name "Default Password Policy"
```

## Modify a password policy

Use the `dsconfig` tool to modify the configuration of any defined password policy. The following example sets properties for the default password policy, and defines two password validators:

```
$ bin/dsconfig set-password-policy-prop \
  --policy-name "Default Password Policy" \
  --set "max-password-age:90 days" \
  --set "password-expiration-warning-interval:14 days" \
  --set "lockout-failure-count:3" \
  --set "password-history-count:6" \
  --add "password-validator:Commonly-Used Passwords" \
  --add "password-validator:Length-Based Password Validator"
```

## Configure one time password delivery

One Time Password (OTP) Delivery mechanisms are used for resetting passwords (in the password recovery flow), and during account registration to validate that the user is a real person. This requires creating and enabling the OTP mechanism on the Directory Server, and defining the delivery mechanism for reset tokens. By default, the SSAM install creates an SMTP delivery mechanism, and configures the operation handlers to use it.

If a different or additional delivery mechanism is needed, such as SMS, configure the password reset token and single use token extended operation handlers with an ordered list of delivery

mechanisms. When sending a token, the server will iterate through the configured delivery mechanisms, to determine if they can be used. If the user's entry has a `mobile` attribute, the delivery mechanism will send an SMS message to it. Otherwise, the SMTP delivery mechanism will send an email using the user's `mail` attribute. The preferred delivery mechanism can be configured on a per-user basis by setting the `ds-auth-preferred-otp-delivery-mechanism` attribute.

The following is a sample for adding a Twilio delivery mechanism to send SMS messages, and configuring the extended operation handlers to prefer this over email:

```
$ bin/dsconfig create-otp-delivery-mechanism \
  --mechanism-name Twilio \
  --type twilio \
  --set enabled:true \
  --set twilio-account-sid:<account-sid> \
  --set "twilio-auth-token:<auth-token>" \
  --set sender-phone-number:<number>
```

```
$ bin/dsconfig set-extended-operation-handler-prop \
  --handler-name Single-Use-Token \
  --set default-otp-delivery-mechanism:Twilio \
  --set default-otp-delivery-mechanism:Email
```

```
$ bin/dsconfig set-extended-operation-handler-prop \
--handler-name "Password Reset Token" \
--set default-token-delivery-mechanism:Twilio \
--set default-token-delivery-mechanism:Email
```

See the *PingData Directory Server Administration Guide* or the *Directory Server Configuration Reference Guide* for complete information about delivering password reset tokens and configuring Extended Operation Handlers.

# Configure the Directory Server schema

By default, entries created by SSAM, contain the `ubidPerson` object class, which enables any schema attributes defined in the `ubidPerson` objectClass to be used. The `ubidPerson` object class is similar to the Directory Server's `inetOrgPerson` object class, but `ubidPerson` objectClass attributes are all optional. Custom schema can be added to the Directory Server for additional attributes. [Velocity template](#) changes are required to use additional attributes.

The default Directory Server `uid` or `mail` attribute must be used to uniquely identify user entries. The Directory Server's identity mapper uses both attributes. SSAM uses a privileged account to perform operations on behalf of the currently authenticated user, which relies on the default Directory Server identity mapper that maps user IDs to entries, as well as a unique attribute plugin configured for the `uid` and `mail` attributes.

If a different attribute must be used to uniquely identify a user, Directory Server server configuration is required. The Directory Server uses the email address in both the `uid` and `mail` attributes to create users. Unless additional configuration is done with the schema, the two attributes should remain synchronized to enable modifying the email address.

By default, SSAM performs SASL PLAIN bind operations to authenticate users when logging in (in non-Ping environments). The authorization ID that is used to bind is based on the username entered into the login form (`u:user@example.com`). SSAM can be optionally configured to use "search and bind" authentication instead of SASL PLAIN by providing a `searchBindFilter` property value. In this case, SSAM will perform a search using the configured base DN and filter to look for the user's entry, and will then perform a simple bind operation using the DN of the user's entry. The `namingAttribute` property is also used by SSAM in this case to search for the user's entry.

If not using the default schema, the `objectClasses` (and possibly `namingAttribute`) properties may need to be updated. SSAM uses the `ubidPerson` object class, but this will need to be modified if using a different structural object class or when adding auxiliary object classes to `ubidPerson`. This is configured in the `application.properties` file.

## Modify the schema

New attributes and object classes can be added to the Directory Server schema using the Directory Server Administrative Console Schema Editor. Make sure to define the attributes first, then define the object classes. To make sure the attributes are surfaced to users through SSAM, edit the application's Velocity templates.

After the new object class is created, edit the SSAM `application.properties` file (in `<server-root>/webapps/ssam-config`). The file must be updated to reflect `objectClasses=ubidPerson,ssamUser`.

The Velocity `user.vm` template is the main template that surfaces user attributes, and should be updated to surface any custom attributes. See Adding an Attribute to the SSAM Templates for steps to add the new `hometown` and `highSchool` attributes to the SSAM pages.

### Note
After all configuration changes are made, including template changes and rebuilding the application `.war` file, the server must be restarted.

# Configure SSAM templates

SSAM relies on Velocity templates for login, registration, password update, and error pages, which are located in:

`src/main/resources/templates`

Files include:

**deletion-success** – Defines the success page for deleting an account.

**error** – Defines the presentation of general error messages displayed to end users.

**login** – Defines the log in page.

**recover-password** – Defines the prompt for information to search for a user account so the password can be changed.

**recover-password-success** – Defines the password change success notification.

**recover-password-verify** – Defines the prompt for the password change code sent by the server and the new password.

**register** – Provides a form for creating a new user account.

**registration-success** – Defines the notification that the user account was successfully created.

**registration-verify** – Defines the information used to verify that the registering user is an actual person, such as requiring a verification code.

**user** – Defines the account attributes that are surfaced. This is the main template used for displaying user attributes, which enables users to perform actions against their accounts.

# Modify application pages

SSAM pages can be customized to reflect specific business requirements and branding. SSAM uses Bootstrap and a standard cascading style sheet located in `sources/src/main/resources/static/css/ssam.css`. If needed, fonts and colors can be changed in this file.

> Note
> Any modifications require rebuilding and redeploying the `ssam.war` file. See Rebuild the SSAM .war File for details.

## Change the company logo

Any images that are used by SSAM are stored in `sources/src/main/resources/static/images`. To change the logo, replace the `brand.png` image with a custom image (with the same name). The templates will pick this up automatically.

## Adding an attribute to the SSAM templates

Any additional attributes must be added to the backend Directory Server first and then added to SSAM's Velocity templates. The SSAM .war file must be rebuilt and the Directory Server server must be restarted for changes to take effect. When editing the SSAM templates, use the following conventions:

- The `name` attribute of each input element should be the same as the corresponding attribute in the schema. This ensures that new attributes can be added by simply updating the Velocity templates, rather than requiring changes to the server-side Java code.

- Set the `value` attribute of each input to `$!attributeName`. The server-side Java code puts all of the attributes into the model, which can be referenced directly as variables in the Velocity template.

- The exclamation (`!`) character is standard Velocity syntax to include the value of the specified variable, or an empty string if there is no variable.

The `user.vm` template is primarily used to display account details, and allow users to update these details. These attributes can also be added to the `register.vm` Velocity template, which would allow users to provide specific data when registering new accounts.

To add the attributes from the [Extending the Schema](#) example, edit the `sources/src/main/resources/templates/user.vm` Velocity template, adding the following statements before the closing `</form>` tag:

```
<div class="form-group">
  <label class="control-label"
   for="hometown">Hometown</label>
  <input type="text" class="form-control"
   name="hometown" value="$!hometown" placeholder="Hometown">
</div>
```

```
<div class="form-group">
  <label class="control-label"
   for="highSchool">High School</label>
  <input type="text" class="form-control"
   name="highSchool" value="$!highSchool" placeholder="High School">
</div>
```

### Make fields required

To make form inputs required, add the `required` attribute before the closing tag to the element:

```
<input ... class="form-control" name="my-attribute" required>
```

To show users that the input is required, add the `required` class to the `class` attribute of the corresponding `label` element:

```
<label for="my-attribute" class="control-label required">...</label>
```

## Rebuild the SSAM .war file

Any configuration changes will require rebuilding the `ssam.war` file. SSAM is a standard Maven project, and requires Maven and the latest Java JDK to build.

Perform the following to rebuild the SSAM package:

1. If needed, install Maven from the `maven.apache.org` site.

2. If needed, install the latest Java JDK from the `www.oracle.com` site.

3. Set environment paths to reflect the locations of these installations.

4. Make any necessary changes to the SSAM templates in `sources/src/main/resources/templates`.

5. From the `sources` directory run the following command:

   ```
   $ mvn clean install
   ```

This will produce an executable war file (`target/ssam-<version>.war`) that can either run by itself (this uses an embedded Apache Tomcat server), or can be deployed in a standard servlet container or the Directory Server (in `<server-root>/webapps/ssam.war`)

# Index