



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Web Services and Applications Deployment Guide

SAML authentication

SAML authentication

Contents

- [1 SAML authentication](#)
 - [1.1 Configuring SAML](#)
 - [1.2 Generating security keys](#)
 - [1.3 Next step](#)

Web Services supports Security Assertion Markup Language (SAML) for single sign-on (SSO) authentication.

Configuring SAML

To enable SAML, make the following configuration changes in the `serverSettings` section of the **application.yaml** file on each of your Web Services nodes (**server-settings.yaml** if you're installing Web Services and Applications version 8.5.201.09 or earlier):

Start

1. Set the following options in the SSL and CA section:
 - **caCertificate** — should point to a JKS key storage that includes the SAML encryption key. See [Generating security keys](#) for details.
 - **jksPassword** — should be the password for the **caCertificate** key storage.
2. Set the following option in the SAML section:
 - **samlSettings** — the following properties are mandatory:
 - `encryptionKeyName`
 - `signingKeyName`
 - `identityProviderMetadata`
3. Save the changes to the file. Your configuration should look something like this:

```
# SSL and CA
caCertificate: /Users/samluser/Documents/Keys/keystore.jks
jksPassword: password

# SAML
samlSettings:
  serviceProviderEntityId: genesys.staging.GWS
  encryptionKeyName: client
  signingKeyName: client
  identityProviderMetadata: /Users/samluser/Documents/Metadata/idp-metadata.xml
```

4. To activate SAML authentication, append the browser URL for Workspace Web Edition with `?authType=saml`.
5. To enable extended SAML logging, add the following string to **logback.xml** file: `<logger name="org.springframework.security.saml" level="%LEVEL%"/>`, where valid values for LEVEL are INFO (preferred) or DEBUG.

End

Generating security keys

You can use the `keytool` utility that comes with the Java SDK to generate a JKS key store. Use the

following command:

```
keytool -genkey -keystore <path_to_jks_file> -alias <key_name> -keypass <key_password>  
-storepass <store_password> -dname <distinguished_name>
```

If you already have a JKS key store, you can add a key to it by executing the command above with the same file name and the new key name and key password. For example:

```
keytool -genkey -keystore /opt/keystore.jks -alias encryption_key -keypass genesys -storepass  
genesys -dname "CN=GWS, OU=R&D, O=Genesys, L=Daly City, S=California, C=US"
```

Next step

- [Back to Configuring security](#)