External Authentication Configuration Guide

CUSTOMER

SAP API Management, On-Premise Edition

Typographic Conventions

Type Style	Description
Example	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Textual cross-references to other documents.
Example	Emphasized words or expressions.
EXAMPLE	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
`Example`	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE	Keys on the keyboard, for example, F2 or ENTER.

Document History

Version	Date	Change
1.0	2015-10-27	Document Created
1.1	2016-04-28	Document aligned with SP04 release
1.2	2016-06-22	Document aligned with SP05 release
1.3	2016-10-05	Document aligned with SP06 release
1.4	2017-05-04	Document aligned with SP07 release
1.5	2017-11-21	Document aligned with SP08 release

Table of Contents

Data Protection and Privacy 6

1 Introduction 7
1.1 Audience 7
1.2 Overview 7

2 What you need to know about SAP API Management Edge authentication and authorization 8
2.1 About authentication 8
2.2 About authorization 8

3 Understanding direct and indirect binding authentication 9
3.1 About indirect binding authentication 9
3.2 About direct binding authentication 9

4 Enabling external authentication 10
4.1 Prerequisites 10
4.2 Configuration overview 10
4.3 Configuring the management-server.properties file 11
4.3.1 DIRECT BINDING configuration sample 12
4.3.2 INDIRECT BINDING configuration sample 13
4.4 Testing the installation 14

5 Disabling the reset password link in the SAP API Management Edge UI 16

6 Indirect binding only: Encrypting the external LDAP user's password 17
6.1 Testing the installation 17

7 Configuring TLS/SSL 18
7.1 Testing the configuration 18

8 Configuration required in the event of different sysadmin credentials 19
8.1 Changing the SAP API Management Edge UI password 19
8.1.1 Changing the Edge UI credential for an email address 19
8.1.2 Changing the Edge UI credential for a user ID 20
8.2 Testing the configuration 20
8.3 Modifying the SAP API Management Edge sysadmin username store for utility scripts 20
8.4 Testing the configuration 21

9 Disable external authentication 22

10 External Role Mapping 23
10.1 Prerequisites 23
10.2 Ensure users are registered on SAP API Management Edge and in your directory service 23
10.3 Default configuration 23

10.4 Enabling External Role Mapping 23
10.5 Disabling External Authorization 24
10.6 About the ExternalRoleMapperImpl sample implementation 25

11 Understanding SAP API Management Edge authentication and authorization flows 31
11.1 When logging in through the UI 31
11.2 When logging in through APIs 31

12 Appendix 33
12.1 A. External authentication configuration options for security.properties 33

Data Protection and Privacy

SAP Customer may use SAP API Management to process and monitor personal information in terms of relevant data protection legislation. It is the SAP Customer's responsibility to use the SAP API Management only in compliance with the relevant data protection laws.

1 Introduction

This document explains how to integrate an external directory service into an existing SAP API Management Edge installation. This feature is designed to work with any directory service that supports LDAP, such as Active Directory, OpenLDAP, and others. All the steps are included here to get SAP API Management Edge working with your LDAP service.

An external LDAP solution allows system administrators to manage user credentials from a centralized directory management service, external to systems like SAP API Management Edge that use them. The feature described in this document supports both direct and indirect binding authentication.

1.1 Audience

This document assumes that you are an SAP API Management Edge global system administrator and that you have an account the external directory service.

1.2 Overview

By default, SAP API Management Edge uses an internal OpenLDAP instance to store credentials that are used for user authentication. However, you can configure SAP API Management Edge to use an external authentication LDAP service instead of the internal one. The procedure for this external configuration is explained in this document.

SAP API Management Edge also stores role-based access authorization credentials in a separate, internal LDAP instance. Whether or not you configure an external authentication service, authorization credentials are always stored in this internal LDAP instance. The procedure for adding users that exist in the external LDAP system to the SAP API Management Edge authorization LDAP are explained in this document.

Note that *authentication* refers to validating a user's identity, while authorization refers to verifying the level of permission an authenticated user is granted to use SAP API Management Edge features.

2 What you need to know about SAP API Management Edge authentication and authorization

It's useful to understand the difference between authentication and authorization and how SAP API Management SAP API Management Edge manages these two activities.

2.1 About authentication

Users who access SAP API Management Edge either through the UI or APIs must be authenticated. By default, SAP API Management Edge user credentials for authentication are stored in an internal OpenLDAP instance. Typically, users must register or be asked to register for an SAP API Management account, and at that time users provide their username, email address, password credentials and metadata. This information is stored in and managed by the authentication LDAP.

However, if you wish to use an external LDAP to manage user credentials on behalf of SAP API Management Edge, you can do so by configuring SAP API Management Edge to use the external LDAP system instead of the internal one. When an external LDAP is configured, user credentials are validated against that external store, as explained in this document.

2.2 About authorization

SAP API Management organization administrators can grant specific permissions to users to interact with SAP API Management Edge entities like API proxies, products, caches, deployments, and so on. Permissions are granted through the assignment of roles to users. SAP API Management Edge includes several built-in roles, and, if needed, org administrators can define custom roles. For example, a user can be granted authorization (through a role) to create and update API proxies, but not to deploy them to a production environment.

The key credential used by the SAP API Management Edge authorization system is the user's email address. This credential (along with some other metadata) is always stored in SAP API Management Edge's internal authorization LDAP. This LDAP is entirely separate from the authentication LDAP (whether internal or external).

Users who are authenticated through an external LDAP must also be manually provisioned into the authorization LDAP system. Details are explained in this document.

Note: User passwords from the external LDAP system are never stored / cached in the internal authorization system.

Refer to "Organization and Environment Maintenance" in the SAP API Management Technical Operations Guide.

For a deeper view, see Understanding SAP API Management Edge authentication and authorization flows.

3 Understanding direct and indirect binding authentication

The external authorization feature supports both direct and indirect binding authentication through the external LDAP system.

Indirect binding authentication requires a search on the external LDAP for credentials that match the email address, username, or other ID supplied by the user at login. With direct binding authentication, no search is performed--credentials are sent to and validated by the LDAP service directly. Direct binding authentication is more efficient because there is no searching involved.

3.1 About indirect binding authentication

With indirect binding authentication, the user enters a credential, such as an email address, username, or some other attribute, and SAP API Management Edge searches authentication system for this credential/value. If the search result is successful, the system extracts the LDAP DN from the search results and uses it with a provided password to authenticate the user.

The key point to know is that indirect binding authentication requires the caller (e.g., SAP API Management Edge) to provide external LDAP admin credentials so that SAP API Management Edge can "log in" to the external LDAP and perform the search. You must provide these credentials in an SAP API Management Edge configuration file, which is described later in this document. Steps are also described for encrypting the password credential.

3.2 About direct binding authentication

With direct binding authentication, SAP API Management Edge sends credentials entered by a user directly to the external authentication system. In this case, no search is performed on the external system. Either the provided credentials succeed or they fail (e.g., if the user is not present in the external LDAP or if the password is incorrect, the login will fail).

Direct binding authentication does not require you to configure admin credentials for the external auth system in SAP API Management Edge (as with indirect binding authentication); however, there is a simple configuration step that you must perform, which is described later in this document.

4 Enabling external authentication

This section explains how to obtain, install, and configure the components required to integrate an external LDAP service into SAP API Management Edge for user authentication.

- Prerequisites
- Configuring the `management-server.properties` file
- Testing the installation

4.1 Prerequisites

- You must have an SAP API Management Edge SP07 installation.
- You must have global system administrator credentials on SAP API Management Edge to perform this installation.
- You need to know the root directory of your SAP API Management Edge installation. The default root directory is `/opt`.
- You must add your SAP API Management Edge global system administrator credentials to the external LDAP. Remember that by default, the sysadmin credentials are stored in the SAP API Management Edge internal LDAP. Once you switch to the external LDAP, your sysadmin credentials will be authenticated there instead. Therefore, you must provision the credentials to the external system before enabling external authentication in SAP API Management Edge.

  For example, if you have configured and installed SAP API Management Edge with global system administrator credentials as...

  username: `sapapiedgeuser@mydomain.com`

  password: `Secret123`

then the user sapapiedgeuser@mydomain.com with password `Secret123` must also be present in the external LDAP.

- If you are running a management server cluster, note that you must perform all of the steps in this document for each management server.

4.2 Configuration overview

The main activity you'll perform is configuring the management-server.properties file. This activity includes stopping and starting the SAP API Management Edge management server, deciding whether you want to use direct or indirect binding, encrypting sensitive credentials, and other related tasks.

In the following sections, we walk you through each step.

4.3 Configuring the management-server.properties file

1. Important: Decide now whether you intend to use the indirect or direct binding authentication method. This decision will affect some aspects of the configuration. See Understanding direct and indirect binding authentication.
2. Important: You must do an additional configuration (described later in this document) under either (or both) of the following circumstances: (a) if you intend to have users log in using usernames that are not email addresses. In this case, your sysadmin user must also authenticate with a username and/or (b) if the password for your sysadmin user account in your external LDAP is different from the password you configured when you first installed SAP API Management Edge. See Additional configuration required in the event of different sysadmin credentials
3. Important: You must do these config steps on each SAP API Management Edge's management server (if you are running more than one).
4. Open `/opt/apigee/customer/application/management-server.properties` in a text editor. If the file does not exist, create it.
5. Add the following line. Note: Be sure that there are no trailing spaces at the end of the line.

   ```
   conf_security_authentication.user.store=externalized.authentication
   ```

   This line is required. It adds the external authentication feature to your SAP API Managementinstallation.
6. To make this step easy, we have created two well-commented sample configurations -- one for direct and one for indirect binding authentication. See the samples below for the binding you wish to use, and complete the configuration:

- DIRECT BINDING configuration sample
- INDIRECT BINDING configuration sample

7. Restart the Management Server:
   ```
   > /opt/apigee/apigee-service/bin/apigee-service edge-management-server restart
   ```

8. Verify that the server is running:
   ```
   > /opt/apigee/apigee-service/bin/apigee-all status
   ```

4.3.1 DIRECT BINDING configuration sample

```
## The first property is always required to enable the external authorization feature. Do not change it.
conf_security_externalized.authentication.implementation.class=com.apigee.rbac.impl.LdapAuthenticatorImpl

## Identify the type of binding:

   # Set to "true" for direct binding
   # Set to "false" for indirect binding.
## Set it to true for DIRECT binding.
conf_security_externalized.authentication.bind.direct.type=true

## The next seven properties are needed regardless of direct or indirect binding. You need to configure these per your external authentication installation.
## The IP or domain for your external LDAP instance.
conf_security_externalized.authentication.server.url=ldap://localhost:389

## Your external LDAP server version.
conf_security_externalized.authentication.server.version=3


## The server timeout in milliseconds.
conf_security_externalized.authentication.server.conn.timeout=50000


## Change these baseDN values to match your external LDAP service. This attribute value will be provided by your external LDAP administrator, and may have more or fewer dc elements depending on your setup.
conf_security_externalized.authentication.user.store.baseDN=dc=apigee,dc=com

## Do not change this search string. It is used internally.
conf_security_externalized.authentication.user.store.search.query=(&(${userAttribute}=${userId}))

## Identifies the external LDAP property you want to bind against for Authentication. For example if you are binding against an email address, this would typically be in the userPrincipalName property in your external LDAP instance. Alternatively if you are binding against the user’s ID, this would typically be in the sAMAccountName property:
conf_security_externalized.authentication.user.store.user.attribute=userPrincipalName

## The LDAP attribute where the user email value is stored. For direct binding, set it to userPrincipalName.
```

```
conf_security_externalized.authentication.user.store.user.email.attribute=userPrincipa
lName

## ONLY needed for DIRECT binding.
## The direct.bind.user.directDN property defines the string that is used for the bind
against the external authentication service. Ensure it is set as follows:
conf_security_externalized.authentication.direct.bind.user.directDN=${userDN}
```

4.3.2 INDIRECT BINDING configuration sample

```
## Required to enable the external authorization feature. Do not change it.
conf_security_externalized.authentication.implementation.class=com.apigee.rbac.impl.Ld
apAuthenticatorImpl

## Identifies the type of binding:
   # Set to "true" for direct binding
   # Set to "false" for indirect binding.
## Set it to false for INDIRECT binding.
conf_security_externalized.authentication.bind.direct.type=false

## The next seven properties are needed regardless of direct or indirect binding. You
need to configure these per your external LDAP installation.
## The IP or domain for your external LDAP instance.
conf_security_externalized.authentication.server.url=ldap://localhost:389

## Replace with your external LDAP server version.
conf_security_externalized.authentication.server.version=3


## Set the server timeout in milliseconds.
conf_security_externalized.authentication.server.conn.timeout=50000


## Change these baseDN values to match your external LDAP service. This attribute
value will be provided by your external LDAP administrator, and may have more or fewer
dc elements depending on your setup.
conf_security_externalized.authentication.user.store.baseDN=dc=apigee,dc=com



## Do not change this search string. It is used internally.
conf_security_externalized.authentication.user.store.search.query=(&(${userAttribute}=
```

```
${userId}))

## Identifies the external LDAP property you want to bind against for Authentication.
For example if you are binding against an email address, this would typically be in
the userPrincipalName property in your external LDAP instance. Alternatively if you
are binding against the user’s ID, this would typically be in the sAMAccountName
property. See also Additional configuration required in the event of different
sysadmin credentials.
conf_security_externalized.authentication.user.store.user.attribute=userPrincipalName

## Used by SAP API Management to perform the Authorization step and currently, SAP API
Management only supports email address for Authorization. Make sure to set it to the
attribute in your external LDAP that stores the user’s email address. Typically this
will be in the userPrincipalName property. See also What you need to know about Edge
authentication and authorization.
conf_security_externalized.authentication.user.store.user.email.attribute=userPrincipa
lName

## The external LDAP username (for a user with search privileges on the external LDAP)
and password and whether the password is encrypted. You must also set the attribute
externalized.authentication.bind.direct.type to false.
## The password attribute can be encrypted or in plain text. See Indirect binding
only: Encrypting the external LDAP user’s password for encryption instructions. Set
the password.encrypted attribute to "true" if the password is encrypted. Set it to
"false" if the password is in plain text.
conf_security_externalized.authentication.indirect.bind.server.admin.dn=myExtLdapUsern
ame
conf_security_externalized.authentication.indirect.bind.server.admin.password=myExtLda
pPassword
conf_security_externalized.authentication.indirect.bind.server.admin.password.encrypte
d=true
```

4.4 Testing the installation

1. Verify that the server is running:
   `/opt/apigee/apigee-service/bin/apigee-all status`

2. Execute this command, providing a set of SAP API Management Edge global system admin credentials. The API call we're going to test can only be executed by an SAP API Management Edge sysadmin.

   Important: The identical credentials must exist in your external LDAP account. If not, you need to add them now. Note that the username is usually an email address; however, it depends on how you have configured external authentication, as explained previously in this document.

```
curl -v http://<your-management-server-ip>:8080/v1/o -u <SAP API Management Edge
Sysadmin Username>
```

For example:

```
curl -v http://192.168.52.100:8080/v1/o -u jdoe@mydomain.com
```

3. Enter your password when prompted.

   If the command returns a 200 status and a list of organizations, the configuration is correct. This command verifies that the API call to the SAP API Management Edge's management server was successfully authenticated through the external LDAP system.

5 Disabling the reset password link in the SAP API Management Edge UI

By default, the log in screen of the SAP API Management Edge UI includes a link that lets a user reset their password:

However, this link is not integrated with an external authentication server, so you can hide it by using the following procedure:

1. Open the ui.properties file in an editor. If the file does not exist, create it:
   ```
   > vi /<inst_root>/apigee/customer/application/ui.properties
   ```
2. Set the conf_apigee_apigee.feature.disablepasswordreset token to true in ui.properties:
   ```
   conf_apigee_apigee.feature.disablepasswordreset="true"
   ```
3. Save your changes.
4. Restart the SAP API Management Edge UI:
   ```
   > /<inst_root>/apigee/apigee-service/bin/apigee-service edge-ui restart
   ```

To later re-enable this link, set the `conf_apigee_apigee.feature.disablepasswordreset` token to false and restart the SAP API Management Edge UI.

6 Indirect binding only: Encrypting the external LDAP user's password

If you are using indirect binding, you need to provide an external LDAP username and password in `management-server.properties` that SAP API Management uses to log into the external LDAP and perform the indirect credential search.

Note: Using plain text passwords in config files may be adequate for testing purposes; however, for production environments, encryption is highly recommended.

The following steps explain how to encrypt your password:

1. Execute the following Java utility, replacing the <YOUR EXTERNAL LDAP PASSWORD> with your actual external LDAP password:
   ```
   java -cp /opt/apigee/edge-gateway/lib/thirdparty/*:/opt/apigee/edge-gateway/lib/kernel/*:/opt/apigee/edge-gateway/lib/infra/libraries/* com.apigee.util.CredentialUtil --password="<YOUR EXTERNAL LDAP PASSWORD>"
   ```
2. In the output of the command, you will see a newline followed by what looks like a random character string. Copy that string.
3. Edit `/opt/apigee/customer/application/management-server.properties`.
4. Update the below property, replacing `<myAdPassword>` with the string you copied from step 1, above.
   ```
   conf_security_externalized.authentication.indirect.bind.server.admin.password=<myAdPassword>
   ```
5. Be sure the following property is set to true:
   ```
   conf_security_externalized.authentication.indirect.bind.server.admin.password.encrypted=true
   ```
6. Save the file.
7. Restart the Management Server:
   ```
   >/opt/apigee/apigee-service/bin/apigee-service edge-management-server restart
   ```
8. Verify that the server is running:
   ```
   > /opt/apigee/apigee-service/bin/apigee-all status
   ```

6.1 Testing the installation

See the section Configuring the security.properties file, and perform the same test described there.

7 Configuring TLS / SSL

This section explains how to configure SSL for the external authorization server.

1. Install the external LDAP Certificate Services.
2. Obtain the Server Certificate.
   For example: `certutil -ca.cert client.crt`

3. Change to your latest Java version home directory:
   `cd /usr/java/latest`

4. Import the Server Certificate. For example:
   `sudo ./bin/keytool -import -keystore ./jre/lib/security/cacerts -file <FULLY-QUALIFIED-PATH-TO-THE-CERT-FILE> -alias <CERT-ALIAS>`

   Where `<CERT-ALIAS>` is optional, but recommended. Replace `<CERT-ALIAS>` with a text name that you can use later to refer to the certificate, for example if you want to delete it.

   Note: The Default Keystore password used by Java is 'changeit'. If this has been changed already you will need to get your sysadmin to provide the keystore password so you add your certificate.

5. Open `/opt/apigee/customer/application/management-server.properties` in a text editor.
6. Change the `conf_security_externalized.authentication.server.url` property value as follows:
   Old Value : `ldap://localhost:389`
   New Value : `ldaps://localhost:636`

7. Restart the Management Server:
   `/opt/apigee/apigee-service/bin/apigee-service edge-management-server restart`

8. Verify that the server is running:
   `/opt/apigee/apigee-service/bin/apigee-all status`

7.1 Testing the configuration

See the section Configuring the security.properties file, and perform the same test.

8 Configuration required in the event of different sysadmin credentials

When you first installed SAP API Management Edge, a special kind of user was created called a sysadmin user, and at the same time some additional config files were updated with this user's details. If you configure your external LDAP to authenticate using a non-email address username and / or you have a different password in your external LDAP for this sysadmin user, then you will need to make the changes described in this section.

There are two locations that need to be updated:

- SAP API Management Edge UI logs into the SAP API Management server using credentials that are stored encrypted in a configuration file on the Edge UI. This update is required when either / both username or password for your sysadmin user is different.
- SAP API Management Edge stores the sysadmin username in another file which is used when running various SAP API Management utility scripts. This update is only required when the username of your sysadmin user is different.

8.1 Changing the SAP API Management Edge UI password

The way you change the SAP API Management Edge UI password depends on how your external LDAP server represents usernames:

- If usernames are email addresses, use the setup.sh utility to update the SAP API Management Edge UI
- If the usernames are IDs, instead of an email address, use API calls and property files to update the SAP API Management Edge UI

Both procedures are described below.

8.1.1 Changing the Edge UI credential for an email address

1. Edit the silent config file that you used to install the SAP API Management Edge UI to set the following properties:

```
ADMIN_EMAIL=newUser
APIGEE_ADMINPW=newPW
SMTPHOST=smtp.gmail.com
SMTPPORT=465
SMTPUSER=foo@gmail.com
SMTPPASSWORD=bar
SMTPSSL=y
```

   Note that you must include the SMTP properties when passing the new password because all properties on the UI are reset.

2. Use the `apigee-setup` utility to reset the password on the SAP API Management Edge UI from the config file:
   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p ui -f configFile
   ```

8.1.2 Changing the Edge UI credential for a user ID

1. Encrypt the user ID and password:
   ```
   > java -cp "/opt/apigee/edge-ui/conf:/opt/apigee/edge-ui/lib/*" utils.EncryptUtil userName:PWord
   ```
2. Open the ui.properties file in an editor. If the file does not exist, create it:
   ```
   vi /opt/apigee/customer/application/ui.properties
   ```
3. In `ui.properties`, set the `conf_apigee_apigee.mgmt.credential` token to the value returned by the call in Step 1:
   ```
   conf_apigee_apigee.mgmt.credential="STRING_RETURNED_IN_STEP_1"
   ```
4. Set the owner of `ui.properties` to 'apigee':
   ```
   > chown apigee:apigee /opt/apigee/customer/application/ui.properties
   ```
5. Restart the Edge UI:
   ```
   > /opt/apigee/apigee-service/bin/apigee-service edge-ui restart
   ```

8.2 Testing the configuration

1. Open the management UI in a browser at:
   `http://<management-server-IP>:9000/`

   For example:
   `http://192.168.52.100:9000/`

2. Log in using the new credentials. If the login succeeds, the configuration is correct.

8.3 Modifying the SAP API Management Edge sysadmin username store for utility scripts

1. Edit the silent config file that you used to install the SAP API Management Edge UI to set the following property to change the value of `ADMIN_EMAIL` to the username you will be using for your sysadmin user in your external LDAP:
   `APIGEE_EMAIL=newUser`

2. Use the `apigee-setup utility` to reset the username on all Edge component from the config file:
   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p edge -f configFile
   ```

You must run this command on all Edge component on all Edge nodes, including: Management Server, Router, Message Processor, Qpid, Postgres.

8.4 Testing the configuration

Verify that you can access the central POD. On the Management Server, run the following CURL command:

```
> curl -u sysAdminEmail:password http://localhost:8080/v1/servers?pod=central\
```

You should see output in the form:

```
[ {
 "internalIP" : "192.168.1.11",
 "isUp" : true,
 "pod" : "central",
 "reachable" : true,
 "region" : "dc-1",
 "tags" : {
   "property" : [ ]
 },
 "type" : [ "application-datastore", "scheduler-datastore", "management-server",
"auth-datastore", "apimodel-datastore", "user-settings-datastore", "audit-datastore"
],
 "uUID" : "d4bc87c6-2baf-4575-98aa-88c37b260469"
}, {
 "externalHostName" : "localhost",
 "externalIP" : "192.168.1.11",
 "internalHostName" : "localhost",
 "internalIP" : "192.168.1.11",
 "isUp" : true,
 "pod" : "central",
 "reachable" : true,
 "region" : "dc-1",
 "tags" : {
   "property" : [ {
     "name" : "started.at",
     "value" : "1454691312854"
   }, ... ]
 },
 "type" : [ "qpid-server" ],
 "uUID" : "9681202c-8c6e-4da1-b59b-23e3ef092f34"
} ]
```

9 Disable external authentication

Perform these steps if you want to turn off external authentication and revert to using the internal authentication LDAP in SAP API Management Edge.

Important: You must do the following steps on each SAP API Management Edge's management server.

1. Open `/opt/apigee/customer/application/management-server.properties` in a text editor.
2. Set the `conf_security_authentication.user.store` property to ldap.

   Note: Be sure that there are no trailing spaces at the end of the line.
   `conf_security_authentication.user.store=ldap`

3. OPTIONALLY, only applicable if you were using a non-email address username or a different password in your external LDAP for your sysadmin user:
   1. Follow the steps you previously followed in Additional configuration required in the event of different sysadmin credentials, above, but substituting the external LDAP username with your SAP API Management Edge sysadmin user's email address.

4. Restart the management server:
   `/opt/apigee/apigee-service/bin/apigee-service edge-management-server restart`

5. Verify that the server is running:

`/opt/apigee/apigee-service/bin/apigee-all status`

6. Important: An SAP API Management Edge organization administrator must take the following actions after external authentication is turned off:
   - Make sure there are no users in SAP API Management Edge that should not be there. You need to manually remove those users.
   - Communicate to users that because the external authentication has been turned off, they need to either start using whatever their original password was (if they remember) or complete a "forgot password" process in order to log in.

10 External Role Mapping

External Role Mapping lets you map your own groups or roles to role-based access control (RBAC) roles and groups created on SAP API Management Edge.

10.1 Prerequisites

- You must be an SAP API Management administrator with global system admin credentials to perform this configuration.
- You need to know the root directory of your SAP API Management Edge installation. The default root directory is `/opt`. If you chose a different root directory during the SAP API Management Edge installation, use that instead of `/opt` as you follow these instructions.
- Obtain the required JAR files from SAP API Management.

10.2 Ensure users are registered on SAP API Management Edge and in your directory service

When using role mapping, all users who access SAP API Management Edge must exist in both your external directory service and in the SAP API Management Edge user repository. That means when you add a user to your external directory service, you must also add that same user to the SAP API Management user repository.

For example, user *a01@company.com* exists in your external directory group 'apiadmin'. You then want to map user *a01@company.com* to the orgadmin role in SAP API Management Edge. Therefore, user *a01@company.com* must first be added to the orgadmin group on SAP API Management Edge.

10.3 Default configuration

External role mapping is disabled by default.

10.4 Enabling External Role Mapping

1. Before you can complete the following configuration, you must create a Java class that implements the `ExternalRoleMapperService` interface. For details about this implementation, see About the ExternalRoleMapperImpl sample implementation.

2. Log into your SAP API Management Edge Management Server and then stop the Management Server:
   `/opt/apigee/apigee-service/bin/apigee-service edge-management-server stop`

3. Check the status of the servers. Be sure the Management Server is stopped / not running:
   `/opt/apigee/apigee-service/bin/apigee-all status`

4. Open `/opt/apigee/customer/application/management-server.properties` in a text editor.
5. Edit the management-server.properties file with the following settings:

   ```
   conf_security_authentication.user.store=externalized.authentication
   conf_security_externalized.authentication.role.mapper.enabled=true
   conf_security_externalized.authentication.role.mapper.implementation.class=com.cust
   omer.authorization.impl.ExternalRoleMapperImpl
   ```

   Important: The implementation class and package name referenced above (`ExternalRoleMapperImpl`) is only an example -- it is a class that you must implement and that you can name the class and package whatever you wish. For details about implementing this class, see About the ExternalRoleMapperImpl sample implementation.

6. Save the `management-server.properties file`.
7. Start the Management Server:

   ```
   /opt/apigee/apigee-service/bin/apigee-service edge-management-server start
   ```

8. Verify that the server is running:

   ```
   /opt/apigee/apigee-service/bin/apigee-all status
   ```

10.5 Disabling External Authorization

To disable external authorization:

1. Open `/opt/apigee/customer/application/management-server.properties` in a text editor.
2. Change the authentication user store to ldap:
   `conf_security_authentication.user.store=ldap`
3. Set this property to false:
   `conf_security_externalized.authentication.role.mapper.enabled=false`
4. Restart the Management Server:
   `/opt/apigee/apigee-service/bin/apigee-service edge-management-server restart`

10.6 About the ExternalRoleMapperImpl sample implementation

In the `management-server.properties config` file described previously in Enabling External Role Mapping, note the this line:

```
conf_security_externalized.authentication.role.mapper.implementation.class=com.customer.authorization.impl.ExternalRoleMapperImpl
```

This class implements the `ExternalRoleMapperService` interface, and is required. You need to create your own implementation of this class that reflects your respective groups. When finished, place the compiled class in a JAR and put that JAR in `/<install_dir>/apigee/edge-gateway/lib/infra/libraries`.

Caution:

If you apply a patch to Edge, or if you upgrade Edge to a later version, you must re-copy the JAR file to `/<instal_dir>/apigee/edge-gateway/lib/infra/libraries`.

You can name the class and package whatever you wish as long as it implements `ExternalRoleMapperService`, is accessible in your classpath, and is referenced correctly in the `management-server.properties` config file.

Below is a well-commented sample implementation of an `ExternalRoleMapperImpl` class. To compile this class, you must reference the following JAR file included with Edge:

```
/<install_dir>/apigee/edge-gateway/lib/infra/libraries/authentication-1.0.0.jar
```

Note: For better readability, we recommend that you copy the code into a text editor or IDE with wider margins.

```
package com.apigee.authenticate;

import com.apigee.authenticate.factory.AuthenticationUtil;
import com.apigee.authenticate.factory.LdapContextFactory;
import com.apigee.authentication.ConfigBean;
import com.apigee.authentication.ConnectionException;
import com.apigee.authentication.ExternalRoleMapperService;
import com.apigee.authentication.NameSpace;
import com.apigee.authentication.NameSpacedRole;
import com.apigee.authorization.namespace.OrganizationNamespace;
import com.apigee.authorization.namespace.SystemNamespace;
import java.util.Collection;
import java.util.HashSet;
import javax.naming.NamingEnumeration;
import javax.naming.NamingException;
import javax.naming.directory.Attributes;
```

```
import javax.naming.directory.DirContext;
import javax.naming.directory.InitialDirContext;
import javax.naming.directory.SearchControls;
import javax.naming.directory.SearchResult;

/** *
 * Sample Implementation constructed with dummy roles with required namespaces.
 *
 * Created by GopiAlagar on 6/12/15.
 */

public class ExternalRoleMapperImpl implements ExternalRoleMapperService {
       InitialDirContext dirContext = null;

       @Override
       public void stop() throws Exception {
       }

       /**
        *
        * This method would be implemented by the customer, Below is the basic
        * example.
        *
        * If User has sysadmin role then it's expected to set SystemNameSpace
        * along with the
        * res\quested NameSpace. Otherwise role's requestedNameSpace to be set
       * for the NameSpacedRole.
        *
        * Collection<NameSpacedRole> results = new HashSet<NameSpacedRole>();
        *
        * NameSpacedRole sysNameSpace = new NameSpacedRole("sysadmin",
        * SystemNamespace.get());
        *
        * String orgName =
        * ((OrganizationNamespace)requestedNameSpace).getOrganization();
        *
        * NameSpacedRole orgNameSpace = new NameSpacedRole ("orgadmin",
        * requestedNameSpace);
        *
        * results.add(sysNameSpace);
        *
```

```
 * results.add(orgNameSpace);
 */

public Collection<NameSpacedRole> getUserRoles(String userName,
            String password, NameSpace requestedNameSpace) {
      /*
       * There are 3 actions performed in the below implementation
       *
       * 1. Authenticate Given User against ADS
       *
       * 2. Fetch the internal groups from the ADS
       *
       * 3. Map the internal group into the apigee-edge roles
       */

      /*************************************************************/
      /****************** Authenticate Given User ******************/
      /*************************************************************/

      // Customer Specific Implementation will override this method
      // implementation

      String dnName = AuthenticationUtil
                  .getDistinguishedNameForEmailId(userName);

      String userEmail = null;

      if (dnName == null) {
            System.out.println("Error ");
      }

      DirContext dirContext = null;

      Collection<NameSpacedRole> results = new HashSet<NameSpacedRole>();

      try {

            dirContext = LdapContextFactory.createLdapContextUsingCredentials(
                        dnName, password);

            userEmail = AuthenticationUtil.getEmailForUserName(userName);
```

```
                    /*************************************************/
                    /********** Fetch internal groups ******************/
                    /*************************************************/

                    String groupDN = "OU=Groups,DC=corp,DC=wacapps,DC=net";
                    SearchControls controls = new SearchControls();
                    controls.setSearchScope(SearchControls.ONELEVEL_SCOPE);
                    NamingEnumeration<SearchResult> groups =
dirContext.search(groupDN,
                                  "(objectClass=*)", new Object[] { "", "" },
controls);

                    if (groups.hasMoreElements()) {
                           while (groups.hasMoreElements()) {
                                  SearchResult searchResult = groups.nextElement();
                                  Attributes attributes = searchResult.getAttributes();
                                  String groupName =
attributes.get("name").get().toString();


                                /************************************/
                                /** Map the internal group into the ***/
                                /** apigee-edge roles               ***/
                                /************************************/

                                  if (groupName.equals("BusDev")) {
                                         results.add(new
NameSpacedRole("businessAdmin",
                                                       SystemNamespace.get()));

                                  } else if (groupName.equals("DevSupport")) {
                                         results.add(new NameSpacedRole("devOpsAdmin",
                                                       SystemNamespace.get()));

                                  } else if (groupName.equals("Engineering")) {
                                         if (requestedNameSpace instanceof
OrganizationNamespace) {
                                                String orgName =
((OrganizationNamespace) requestedNameSpace)
                                                              .getOrganization();
```

```
                                        results.add(new
NameSpacedRole("orgadmin",
                                                    new
OrganizationNamespace(orgName)));
                                }

                            } else if (groupName.equals("Operations")
                                        || groupName.equals("IT")) {

                                results.add(new NameSpacedRole("sysadmin",
                                            SystemNamespace.get()));

                            } else if (groupName.equals("Marketing")) {

                                results.add(new NameSpacedRole("marketAdmin",
                                            SystemNamespace.get()));

                            } else {

                                results.add(new NameSpacedRole("readOnly",
                                            SystemNamespace.get()));
                            }
                        }

                    } else {

                        /*                          *
                         * In case of no group found or exception found we throw
empty
                         * roles.
                         */

                        System.out.println(" !!!!! NO  GROUPS FOUND !!!!!");
                    }

                } catch (Exception ex) {
                    ex.printStackTrace();
                    System.out.println("Error in authenticating User: {}"
                            + new Object[] { userName });
                } finally {
                    AuthenticationUtil.closeDirContext(dirContext);
```

```
		}

		return results;

	}

	@Override
	public void start(ConfigBean arg0) throws ConnectionException {

		try {
			// Create InitialDirContext
			dirContext = LdapContextFactory.createLdapContext();
		} catch (NamingException e) {
			// TODO Auto-generated catch block
			throw new ConnectionException(e);
		}
	}
}
```

11 Understanding SAP API Management Edge authentication and authorization flows

This document explains how authentication and authorization work on SAP API Management Edge. This information may provide useful context when you configure an external LDAP with SAP API Management Edge.

The authentication and authorization flows depend whether a user authenticates through the management UI or through the APIs.

11.1 When logging in through the UI

When you log in to SAP API Management Edge through the UI, SAP API Management Edge performs a separate login step to the SAP API Management Edge's management server using the SAP API Management Edge global system administrator credentials.

The credentials are stored in /opt/apigee4/conf/ui/apigee.conf and were automatically set up when you first installed SAP API Management Edge.

The following UI login steps:

1. The user enters login credentials in the login UI.
2. SAP API Management Edge logs in to the Management Server using the global system admin credentials.
3. The global system admin credentials are authenticated and authorized. The UI uses these credentials to make certain platform API requests.
    1. If external authentication is enabled, the credentials are authenticated against the external LDAP, otherwise, the internal SAP API Management Edge LDAP is used.
    2. Authorization is always performed against the internal LDAP.
4. The credentials entered by the user are authenticated and authorized.
    1. If external authentication is enabled, the credentials are authenticated against the external LDAP, otherwise, the internal SAP API Management Edge LDAP is used.
    2. Authorization is always performed against the internal LDAP.

11.2 When logging in through APIs

When you log in to SAP API Management Edge through an API, only the credentials entered with the API are used. Unlike with UI login, a separate login with system admin credentials is not required.

The following API login steps:

1. The user enters login credentials in the login UI.
2. The credentials entered by the user are authenticated and authorized.

3. If external authentication is enabled, the credentials are authenticated against the external LDAP, otherwise, the internal SAP API Management Edge LDAP is used.
4. Authorization is always performed against the internal LDAP.

12 Appendix

12.1 A. External authentication configuration options for security.properties

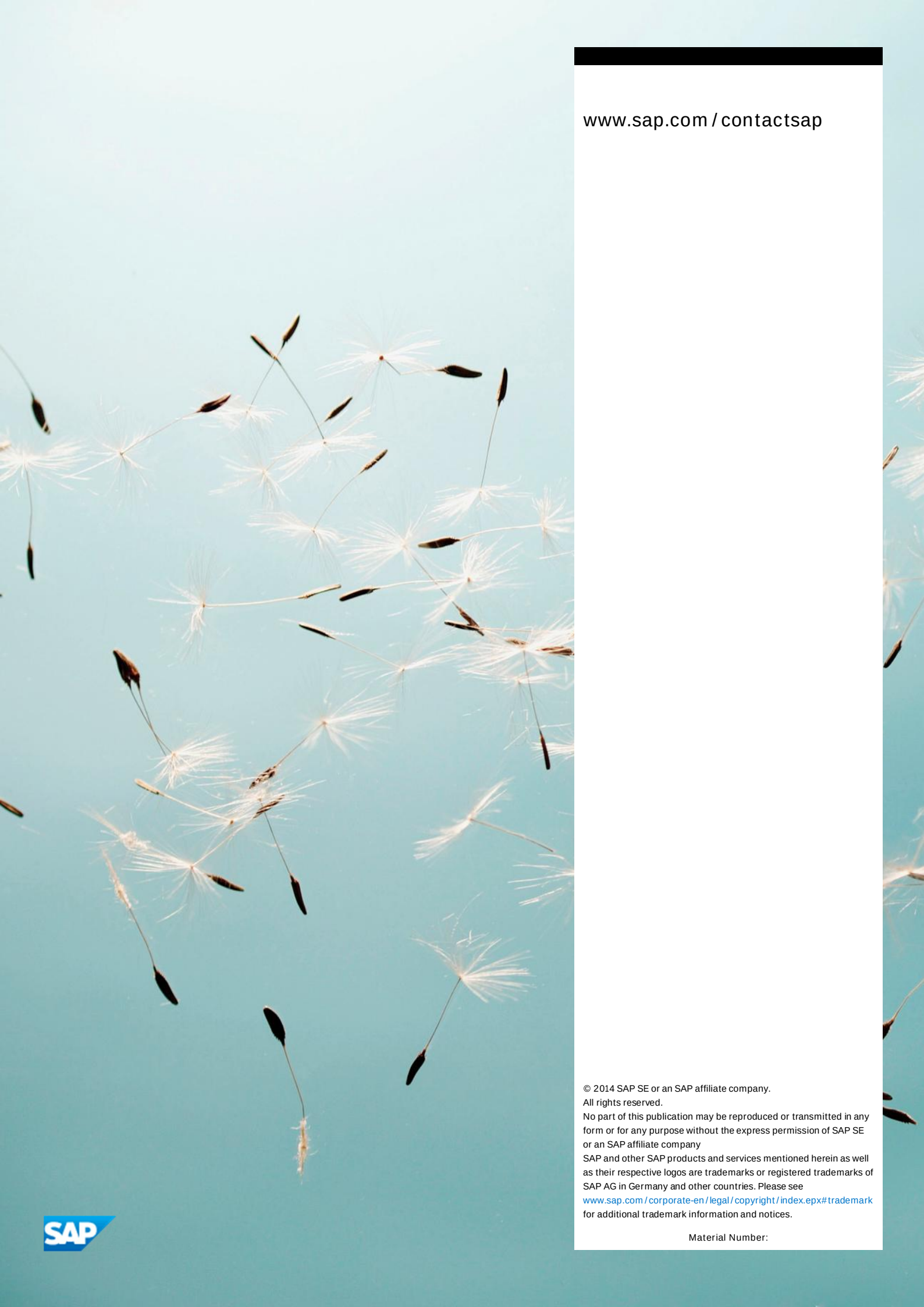
The following table provides a comparison view of `management-server.properties` attributes required for direct and indirect binding for external authentication. Direct and indirect binding are described in Understanding direct and indirect binding authentication

Note, in the following table, values are provided in between " ". When editing the `management-server.properties` file, include the value between the quotes (" ") but do not include the actual quotes.

Property	DIRECT bind	INDIRECT bind
`conf_security_externalized.authentication.implementation.class=com.apigee.rbac.impl.LdapAuthenticatorImpl`		
	This property is always required to enable the external authorization feature. Do not change it.	
conf_security_externalized.authentication.bind.direct.type=		
	Set to "true".	Set to "false".
`conf_security_externalized.authentication.direct.bind.user.directDN=`		
	If the username is an email address, set to `"${userDN}"`. If the username is an ID, set to `"CN=${userDN},CN=Users,DC=apigee,DC=com"`, replacing the CN=Users, `DC=apigee,DC=com` with appropriate values for your external LDAP.	Not required, comment out.
`conf_security_externalized.authentication.indirect.bind.server.admin.dn=`		
	Not required, comment out.	Set to the username/email address of a user with search privileges on the external LDAP.

`conf_security_externalized.authentication.indirect.bind.server.admin.password=`		
	Not required, comment out.	Set to the password for the above user.
`conf_security_externalized.authentication.indirect.bind.server.admin.password.encrypted=`		
	Not required, comment out.	Set to "`false`" if using a plain-text password (NOT RECOMMENDED) Set to "`true`" if using an encrypted password (RECOMMENDED)
conf_security_externalized.authentication.server.url=		
	Set to "`ldap://localhost:389`", replacing "`localhost`" with the IP or domain for your external LDAP instance.	
`conf_security_externalized.authentication.server.version=`		
	Set to your external LDAP server version, e.g. "3".	
`conf_security_externalized.authentication.server.conn.timeout=`		
	Set to a timeout (number in milliseconds) that is appropriate for your external LDAP.	
`conf_security_externalized.authentication.user.store.baseDN=`		
	Set to the baseDN value to match your external LDAP service. This value will be provided by your external LDAP administrator. E.g. in SAP API Management we might use "DC=apigee,DC=com"	
`conf_security_externalized.authentication.user.store.search.query=(&(${userAttribute}=${userId}))`		
	Do not change this search string. It is used internally.	
`conf_security_externalized.authentication.user.store.user.attribute=`		
	This identifies the external LDAP property you want to bind against. Set to whichever property contains the username in the format that your users use to log into SAP API Management Edge. For example: If users will log in with an email address and that credential is stored in "`userPrincipalName`", set above to "`userPrincipalName`".	

	If users will log in with an ID and that is stored in "sAMAccountName", set above to "sAMAccountName".
conf_security_externalized.authentication.user.store.user.email.attribute=	
	This is the LDAP attribute where the user email value is stored. This is typically "userPrincipalName" but set this to whichever property in your external LDAP contains the user's email address that is provisioned into SAP API Management's internal authorization LDAP.



www.sap.com/contactsap

Material Number: