REPORT ON

PINGIDENTITY

|  |  |
| --- | --- |
| Document Name | PingIdentity Implementation |
| Classification | For Deloitte use only |
| Version | 3.1 |
| Effective Date | 22-04-2022 |
| Document Owner | Deloitte |
| Author | //Write your own name here// |
| Status | Draft |

**Document Information**

**History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Revision Authors | Notes |
|  | 1.0 |  | Draft |
|  | 1.1 |  | Updated the Document |

TABLE OF CONTENTS

[CHAPTER-1: INTRODUCTION TO PING IDENTITY 5](#_Toc101531402)

[1.1 Overview 5](#_Toc101531403)

[1.2 Open Standard Protocols 6](#_Toc101531404)

[1.2.1 SAML 2.0 6](#_Toc101531405)

[1.2.2 OAuth 2.0 7](#_Toc101531406)

[1.2.3 OpenID Connect 8](#_Toc101531407)

[1.2.4 SAML v/s OAuth v/s OpenID Connect 8](#_Toc101531408)

[CHAPTER-2: PINGDIRECTORY 10](#_Toc101531409)

[2.1 What is PingDirectory? 10](#_Toc101531410)

[2.2 Installation of PingDirectory 10](#_Toc101531411)

[2.3 Configuration of PingDirectory 11](#_Toc101531412)

[2.4 Folder Configuration 13](#_Toc101531413)

[2.5 Protocol used in PingDirectory 14](#_Toc101531414)

[2.5.1 What is LDAP? 14](#_Toc101531415)

[2.5.2 LDAP v/s Relational Database 14](#_Toc101531416)

[2.5.3 Terminologies used in LDAP 14](#_Toc101531417)

[2.6 Commands in PingDirectory 15](#_Toc101531418)

[CHAPTER-3: PINGFEDERATE 16](#_Toc101531419)

[3.1 What is PingFederate? 16](#_Toc101531420)

[3.2 Installation and Configuration 17](#_Toc101531421)

[3.3 Folder Configuration 19](#_Toc101531422)

[3.4 Protocols used in Ping Federate 20](#_Toc101531423)

[3.4.1 Working of SAML Protocol 20](#_Toc101531424)

[3.4.2 IdP initiated SSO 21](#_Toc101531425)

[3.4.3 SP initiated SSO 22](#_Toc101531426)

[CHAPTER-4: PING ACCESS 24](#_Toc101531427)

[4.1 What is PingAccess? 24](#_Toc101531428)

[4.2 Installation and Configuration 24](#_Toc101531429)

[4.3 Protocols used in PingAccess 25](#_Toc101531430)

[4.3.1 Terminologies used in OAuth 2.0 25](#_Toc101531431)

[4.3.2 How OAuth 2.0 Works? 26](#_Toc101531432)

[4.3.3 How OAuth 2.0 and OpenID connect works together? 27](#_Toc101531433)

[CHAPTER-5: USE-CASES 28](#_Toc101531434)

[5.1 Use-Case I: Group Allocation of users in Ping Directory 28](#_Toc101531435)

[5.2 Use-Case II: 29](#_Toc101531436)

[5.2.1 Creating Adapter and PCV 29](#_Toc101531437)

[5.2.2 Creating SP Connection 30](#_Toc101531438)

[5.3 Use-Case III: 33](#_Toc101531439)

LIST OF FIGURES

[Figure 1: Sequence Diagram of SAML 2.0 6](https://amedeloitte.sharepoint.com/sites/NYL984/Shared%20Documents/General/REPORT.docx#_Toc101530211)

[Figure 2: Sequence Diagram of OAuth 2.0 7](https://amedeloitte.sharepoint.com/sites/NYL984/Shared%20Documents/General/REPORT.docx#_Toc101530212)

[Figure 3: Sequence Diagram of OAuth 2.0 7](https://amedeloitte.sharepoint.com/sites/NYL984/Shared%20Documents/General/REPORT.docx#_Toc101530213)

[Figure 5: PingDirectory can be used with data stores such as Active Directory, LDAP and RDBS 9](https://amedeloitte.sharepoint.com/sites/NYL984/Shared%20Documents/General/REPORT.docx#_Toc101530214)

[Figure 6: Installation Path of PingDirectory 12](https://amedeloitte.sharepoint.com/sites/NYL984/Shared%20Documents/General/REPORT.docx#_Toc101530215)

[Figure 7: Example of Directory Information Tree 13](https://amedeloitte.sharepoint.com/sites/NYL984/Shared%20Documents/General/REPORT.docx#_Toc101530216)

[Figure 8: Ping Federate can be used for SaaS Apps, 15](#_Toc101530217)

[Figure 9: Sequence Diagram for IdP-Initiated SSO 20](#_Toc101530218)

[Figure 10:Sequence Diagram for SP-Initiated SSO 21](#_Toc101530219)

[Figure 11: Working of OAuth 2.0 24](https://amedeloitte.sharepoint.com/sites/NYL984/Shared%20Documents/General/REPORT.docx#_Toc101530220)

[Figure 12: OAuth2.0 and OIDC Works together 25](https://amedeloitte.sharepoint.com/sites/NYL984/Shared%20Documents/General/REPORT.docx#_Toc101530221)

TABLE OF USE-CASES

|  |  |  |
| --- | --- | --- |
| S.No | Use-cases | Description |
| 1. | Use-case 1: Group Allocation of users in Ping Directory |  |
| 2. | Use-case 2: |  |
| 3. | Use-case 3: |  |

# CHAPTER-1: INTRODUCTION TO PING IDENTITY

## 1.1 Overview

Ping Identity provides Federated identity Management, Identity Access Management and Single Sign On (SSO) solutions to web Identities.

* Identity Federation- It is a system of trust established between two or more parties in such a way that the Service provider (SP) trusts the identity provider (IDP) for the authentication of the user and if the user is successfully authenticated then the Identity provider sends SP a message called assertion. Assertion contains user sign in name and other attributes of the user which is required by Service provider to establish the session with the user and to figure out what kind of resource access the SP should provide.

For example - When we use any application like Facebook there is an option to sign in using google account this is because a trust is set up between these two parties due to which Facebook trusts the authentication process of google and user is not required to fill in the same details again. In this example Facebook act as Service Provider and Google act as Identity Provider.

Use of such solution in an organization is known as Federated Identity Management.

* Identity Access Management – It is the discipline that enables the right individuals to access the right resources at the right times for the right reasons. Identity refers to the virtual entity of user that has unique and other attributes associated with it which is used for mapping the right user as different user have different attributes. So IAM refers to process of creation of digital identity, authenticating them when they log in, authorizing their access to specified resources, and monitoring and managing these identities throughout their lifecycle.
* Single Sign On (SSO) – Single Sign On is the process by which user gets access to all the applications by authenticating only once into an application. For example, if you log in to Gmail, you get access to YouTube, Google Analytics, and other Google apps without filling in your credentials again.

PingIdentity products include: PingID, PingDirectory, PingFederate, PingAccess, PingDataGovernance and PingOne.

1. PingID- It is a cloud based multifactor authentication solution that balances secured access to applications with ease of use to end users.
2. PingDataGovernance- It provides policy based, fined-grained access control for data protection and filtering for regulatory compliance.
3. PingFederate- It is a federation server used for authentication of users and provides single sign on to partners, employees, and customer identities.
4. PingAccess – It is a policy engine that provides secure access to applications by allowing to define various rules and policies that should be followed by the users for accessing the resources.
5. PingDirectory- It is a PingOne Cloud Directory Service used to store identity and sensitive profile data of customers and uses LDAP protocol to access identities.
6. PingOne- It is an Identity-as-a-service (IDaaS) single sign on (SSO) offering that enables enterprises to give their users federated access to applications with a single click from a secure device.

## 1.2 Open Standard Protocols

Service providers (SPs) and identity providers (IdPs) communicate identity information using open standard protocols for identity federation.

The protocols used in PingIdentity are: -

1. SAML 2.0
2. OAuth 2.0
3. OpenID Connect

### 1.2.1 SAML 2.0

* SAML stands for Security Assertion Markup Language. It is a secure XML-based communication mechanism for communicating identities between organization.
* It eliminates the need to maintain multiple authentication credentials such as passwords in multiple locations. It is used to provide SSO to web-based applications and can be used for both authentication and authorization.

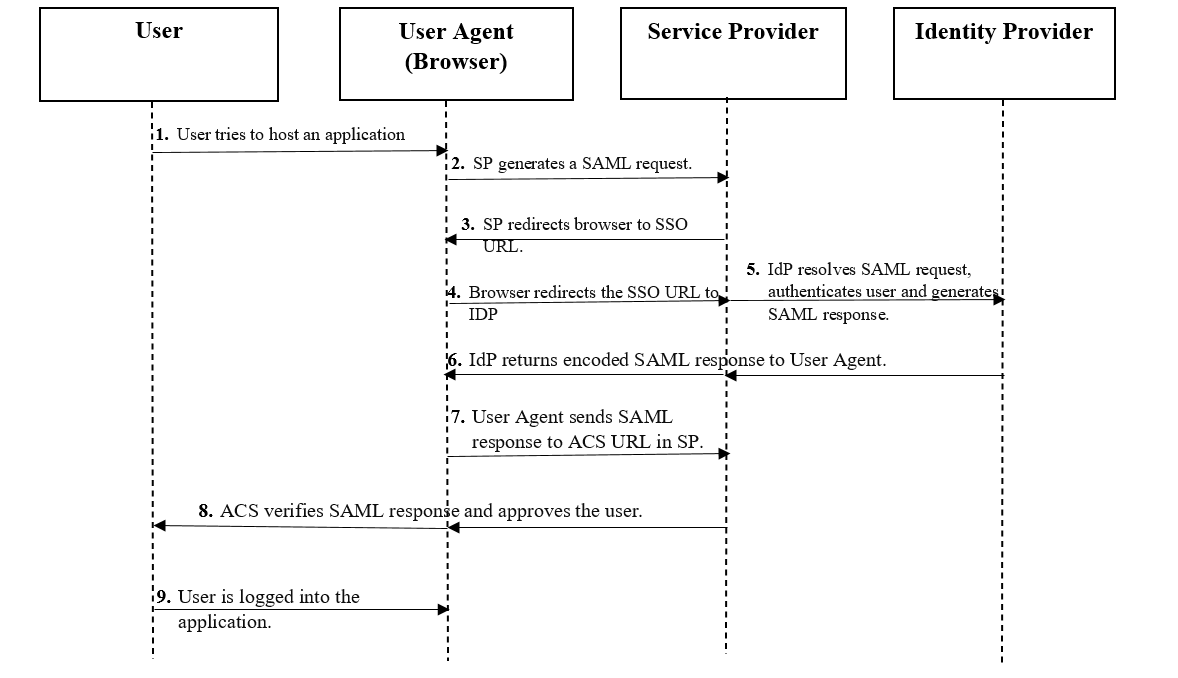


Figure : Sequence Diagram of SAML 2.0

### 1.2.2 OAuth 2.0

OAuth stands for Open Authorization. It is a standard protocol which allows an application or website to have access to resources which are hosted by some other application. It allows a client application to perform some actions on the resources on behalf on user without having user’s credentials. OAuth is an authorization standard and not an authentication one.



Figure : Sequence Diagram of OAuth 2.0

### 

### 1.2.3 OpenID Connect

OpenID connect is a simple identity layer on the top of OAuth 2.0 protocol. It enables client to verify the identities of End-user used based on authentication performed by authentication server as well to obtain basic profile information about the End-user



Figure 3: Working of OpenID Connect

### 1.2.4 SAML v/s OAuth v/s OpenID Connect

|  |  |  |  |
| --- | --- | --- | --- |
|  | **SAML** | **OAuth** | **OpenID Connect** |
| What is it? | Open standard for authentication and authorization | Open standard for authorization | Open standard for authentication |
| History | Developed by OASIS in 2001 | Developed by Twitter and Google in 2006 | Developed by OpenID foundation in 2014 |
| Application | SSO For enterprise | API Authorization | SSO for consumer application |
| Format | XML | JSON | JSON |

# CHAPTER-2: PINGDIRECTORY

## 2.1 What is PingDirectory?

PingDirectory is the PingOne Cloud Platform's Directory service, which securely stores and maintains identity of users at a large scale. PingDirectory is used by many businesses to store and manage sensitive customer, partner, and employee data, such as credentials, profiles, preferences, and privacy settings. PingDirectory allows to manage numerous identities and the corresponding attributes associated with these identities. PingDirectory allows enterprises to save costs by avoiding redundant and inconsistent user data. It prevents cyberattacks as identities and profile data is stored in encrypted format. It can be deployed on Ping’s, public and premise cloud. It uses LDAP protocol to access the identities.

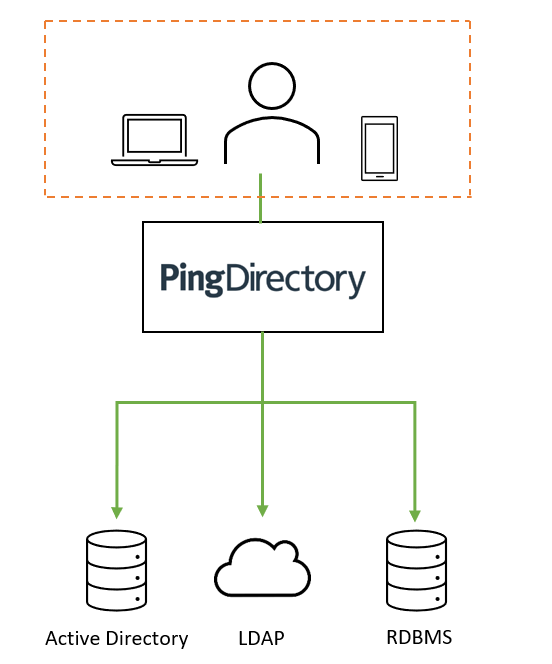


Figure : PingDirectory can be used with data stores such as Active Directory, LDAP and RDBS

## 2.2 Installation of PingDirectory

|  |  |  |
| --- | --- | --- |
| **S.No** | **Steps** | **Screenshots** |
| 1. | Request the license key for PingDirectory on below URL.  <https://www.pingidentity.com/en/account/request-license-key.html> |  |
| 2. | Go to the link: <https://support.pingidentity.com/s/> and then download the license key under the tab ***Manage License Keys → View → Download*** |  |
| 3. | Inside the VM, go to the URL shown below and download the Linux based Product Distribution (ZIP) file.  [https://www.pingidentity.com/en/resources/ downloads/pingdirectory-downloads.html](https://www.pingidentity.com/en/resources/%20downloads/pingdirectory-downloads.html) |  |
| 4. | Extract the downloaded PingDirectory zip file. |  |

## 2.3 Configuration of PingDirectory

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Steps** | **Screenshots** |
| 1. | Inside the PingDirectory folder, open a terminal and run the commands as shown in the figure to begin installation |  |
| 2. | Copy the acquired license file for ping directory to the location mentioned. |  |
| 3. | Enter the host name as **localhost** |  |
| 4. | Set the password and reconfirm it. |  |
| 5. | Select the default options and Press Enter |  |
| 6. | Enter the Same password in encryption passphrase that was set earlier during step 3 |  |
| 7. | Enter any instance and location name for the Directory Server |  |
| 8. | Select the default options and Final Configuration should look as shown in the image aside. |  |

## 2.4 Folder Configuration



Figure : Installation Path of PingDirectory

The PingDirectory installation path would have folders such as bat, bin, collector, config etc.

1. Bat folder- Command line utilities for Windows Operating System.
2. Bin folder- Command line utilities for Linux Operating System.
3. Config Folder- It consist of important folders/files:
4. Schema Folder- It comprised of files which contains attributes and object class for defining entry. Custom attributes can also be created
5. Config file- It stores all the information about directory. This is one of the most important files. The extension of this file is ‘.ldif’.
6. Db Folder- In the backend, there Is a database that actually interacts using LDAP.

* All the database files are stored in directory.
* Cache is stored here.

1. Log Folder - It consist of following log files:
2. Access Log- It consists of information regarding:

All operations performed with entries in Directory like adding, deleting, create etc. But attributes will not be present here for user only its DN.

* Error regarding Wrong password entered will be present here.
* Its purpose is to check whether Authentication was successful or not as it contains request and error code for unsuccessful Authentication.

1. Error Logs- These are important for administration server as it will contain information regarding:

* High Resource Utilization
* Memory full
* Restart of any instance

1. Audit Log – Modifications done to the directory server like addition, deletion etc. Attributes for the user will be also present here.

## 2.5 Protocol used in PingDirectory

PingDirectory uses LDAP protocol to authenticate the identities.

### 2.5.1 What is LDAP?

* LDAP stands for Lightweight Directory Access Protocol. It is a client server based standard protocol used to authenticate the identities stored in the directory.
* As the name suggests the operations are much faster as compared to Relational databases.

### 2.5.2 LDAP v/s Relational Database

|  |  |
| --- | --- |
| LDAP | Relational Database |
| * LDAP has hierarchical tree like structure. | * It has Tabular structure. |
| * In LDAP, it is difficult to represent complex relations. | * In database, complex relationship can be represented efficiently. |
| * LDAP is more optimized for read operation. | * Relational Databases are more optimized for write operations. |

### 2.5.3 Terminologies used in LDAP

1. Entry**:** LDAP entry is a collection of information about an entity.Each entry consists of three components which includes distinguished name, collection of attributes and collection of object classes.
2. Text

   Description automatically generatedDirectory Information tree (DIT):It is a hierarchal structure which represents the entries of organization. It is also called as namespace or LDAP tree.

Figure : Example of Directory Information Tree

1. Distinguished Name (DN):Distinguished name of an entry identifies the unique path of an entry in Directory information tree. It is comprised of attribute-value pairs.

* In figure 3, distinguished name for uid=bob can be expressed as: uid=bob, ou=people, dc=example, dc=com

1. Object Class**:** Object class is used to group related information. It is used to define the structure of entries including the attributes. Each object class allows some attributes which can be seen in schema file.

## 2.6 Commands in PingDirectory

|  |  |  |
| --- | --- | --- |
| **S.No** | **Commands** | **Screenshots** |
| 1. | **start-server.bat** command is used to start the PingDirectory server |  |
| 2. | **stop-server.bat** command is used to stop the ping directory server. |  |
| 3. | Ldapsearch operation is used to fetch the particular account from the Directory.  Command used is:  **ldapsearch.bat -h localhost -p 389 -D cn="Directory Manager" -b dc=example,dc=com uid=\***  where  **-**h is host name  -p is port number  -D is for Binding DN  -b is base |  |
| 4. | Ldapmodify command is used to modify the attributes values of one or more than one user accounts.  Command used is: **ldapmodify -h localhost -p 389 -D "cn=Directory Manager" -f "filepath"** |  |

# CHAPTER-3: PINGFEDERATE

## 3.1 What is PingFederate?

Ping Federate is one of the products of Ping Identity that serves as a global authentication authority, allowing any kind of application to gain authentication and single-sign-on features to provide users with seamless resource access and eliminate insecure password proliferation. PingFederate is a full-featured federation server that provides customers, partners, and employees with identity management, web single sign-on, and API security. Users can securely access the applications they need from any device using a single identity. Ping Federate supports the many open-standard protocols such as [OAuth](https://www.pingidentity.com/en/resources/client-library/articles/oauth.html), [OpenID Connect](https://www.pingidentity.com/en/resources/client-library/articles/openid-connect.html), [SAML](https://www.pingidentity.com/content/ping/en/resources/client-library/articles/saml.html) etc. Also, it can be deployed on both on-cloud and on-prem.

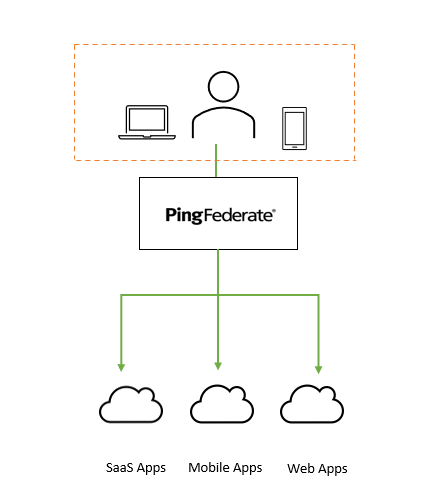


Figure : Ping Federate can be used for SaaS Apps,

Mobile Apps and Web Apps

## 3.2 Installation and Configuration

|  |  |  |
| --- | --- | --- |
| **S.No.** | **STEPS** | **SCREENSHOTS** |
| 1. | Go to the link:  <https://www.pingidentity.com/en/account/request-license-key.html> and request for the Ping Federate license key. |  |
| 2. | Go to the link: <https://support.pingidentity.com/s/> and then click on ***Manage License Keys → View → Download*** to download the license key. |  |
| 3. | Inside the VM, go to the link: [https://www.pingidentity.com/en/resources/ downloads/pingfederate.html](https://www.pingidentity.com/en/resources/%20downloads/pingfederate.html) and download the Linux-based Product Distribution (ZIP) file. |  |
| 4. | Extract the downloaded Ping Federate zip file. |  |
| 5. | Go to the bin folder inside the extracted Ping Federate folder and add the following line at the beginning of run.sh file:  JAVA\_HOME="/usr/lib/jvm/java-11-openjdk-amd64" |  |
| 6. | Now, open the terminal inside the bin folder and type the following command as shown in the image to run the Ping Federate. |  |
| 7. | Open the link:  <https://localhost:9999/> to access the PingFederate console application. Accept the agreement and then click no to Set Up Without PingOne for Enterprise and then select the downloaded license file. |  |

## 3.3 Folder Configuration

Some of the important Ping Federate log files are:

|  |  |  |
| --- | --- | --- |
| S.No. | File | Description |
| 1. | admin.log | Actions performed by administrative-console users are recorded here. |
| 2. | transaction.log | Individual identity-federation runtime transactions at specified levels of detail are recorded here. |
| 3. | audit.log | Transaction log information with additional details at runtime, intended to facilitate security auditing and regulatory compliance are recorded here. |
| 4. | server.log | Runtime and administrative server activities are recorded here. |

**Audit log explanation:**

Audit log contains the details of the transaction log information along with additional details at runtime, intended to facilitate security auditing and regulatory compliance. Activities from SSO, SLO, OAuth, WS-Trust STS, and SCIM inbound provisioning transactions are recorded in the audit log. Audit log contains the following details:

* Transaction Time**:** Gives the details of the transaction time of the session.
* Tracking id**:** It is the unique id for the session.
* Event**:** It tells about the type of session that took place. Example: AUTHN\_ATTEMPT, SSO etc.
* Subject**:** Subject of the transaction.
* ID: Incoming IP address.
* App**:** Target SP application.
* Protocol: Protocol used to complete the transaction for that session.
* Host: Host IP address.
* Status: Status of the transaction.
* AdapterID: ID of Adapter instance.
* Responsetime: The time elapsed (in milliseconds) from when a final request for a transaction is received to when the audit message is written.

Example of Audit log:

2022-04-20 01:25:52,534| tid:r-mNh32hLEQ-7kz-9aj6ZP5gPJU| SSO| mailofsample5@deloitte.com| 127.0.0.1 | /| https://deSloitte-9a.my.salesforce.com| SAML20| ubuntu| IdP| success| LDAPadapter2| | 140

In this log,

* *2022-04-20 01:25:52,534*: Transaction time
* *tid:r-mNh32hLEQ-7kz-9aj6ZP5gPJU*: Tracking id
* *SSO*: Event
* *mailofsample5@deloitte.com*: Subject of transaction
* *127.0.0.1*: Incoming IP address
* *https://deSloitte-9a.my.salesforce.com*: Target SP application
* *SAML20*: Protocol used to complete the transaction
* *Ubuntu*: Host name
* *Success*: status of the transaction
* *IdP*: LDAPadapter2
* *140*: Response time

## 3.4 Protocols used in Ping Federate

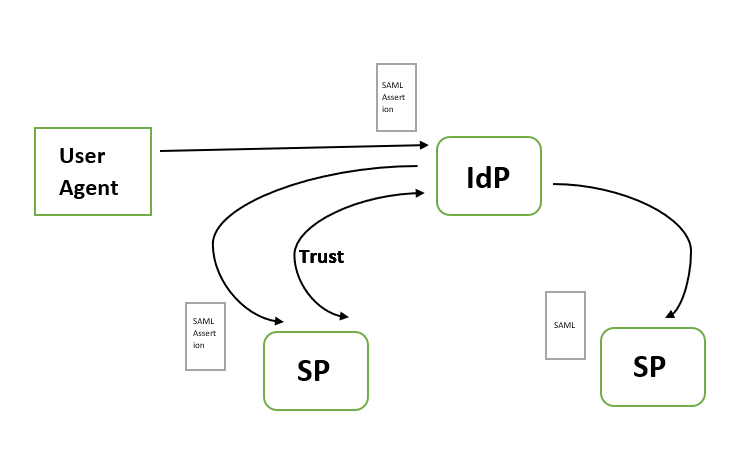
Ping Federate uses SAML protocol to provide authentication (Only when it is not integrated with Ping Access. If used along with Ping Access, then Ping Federate uses OIDC protocol to provide authentication). Ping Federate uses SAML protocol to provide IdP and SP initiated SSO for web applications.

### 3.4.1 Working of SAML Protocol

SAML protocol has 3 entities: IdP (Identity Provider), User Agent and SP (Service Provider). Following steps explains the working of SAML protocol:

* When configuring SAML Federation we establish a trust relationship between SP and IdP.
* User who wants to access SP first needs to authenticate into IdP. It the user can successfully authenticate and authorize, then IdP generates a SAML assertion. Assertion is then sent to the application and since the application trusts IdP, the user can access that application.

***NOTE:*** *Since the user is already authenticated to the IdP, the user can single sign-on to other applications.*



Fig**.**4.2: Working of SAML protocol

### 3.4.2 IdP initiated SSO

Identity Provider (IdP) initiated SSO entails the user clicking a button in the IdP and then being forwarded to an SP along with a SAML message containing an assertion. This flow is typically initiated by a page within the IdP that displays a list of all available SPs to which a user can log in.

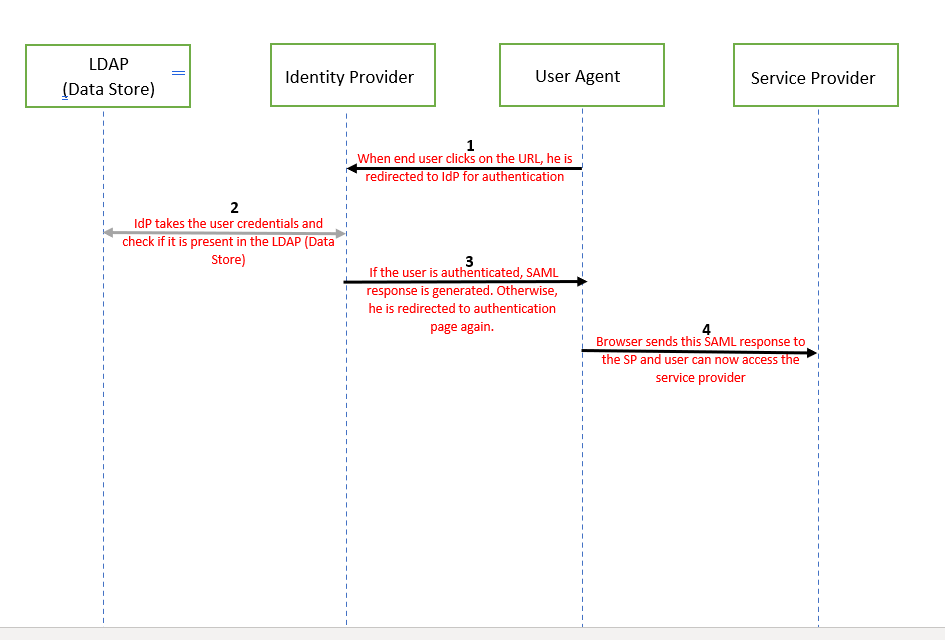


Figure : Sequence Diagram for IdP-Initiated SSO

**STEP 1:** User starts accessing the identity provider

**STEP 2:** User is prompted for authentication to the datastore (LDAP). Once done, user can request a service. If the user is authorized IdP generates a SAML assertion.

**STEP 3:** Using a user agent, assertion is sent to the SP using a post message. It is the user agent who acts as a transport mechanism for the assertion.

**STEP 4:** SP verifies the assertion and maps it to a local user and then the session can start.

### 3.4.3 SP initiated SSO

SP-initiated SSO creates a SAML request and redirects both the user and the request to the identity provider as part of SSO.

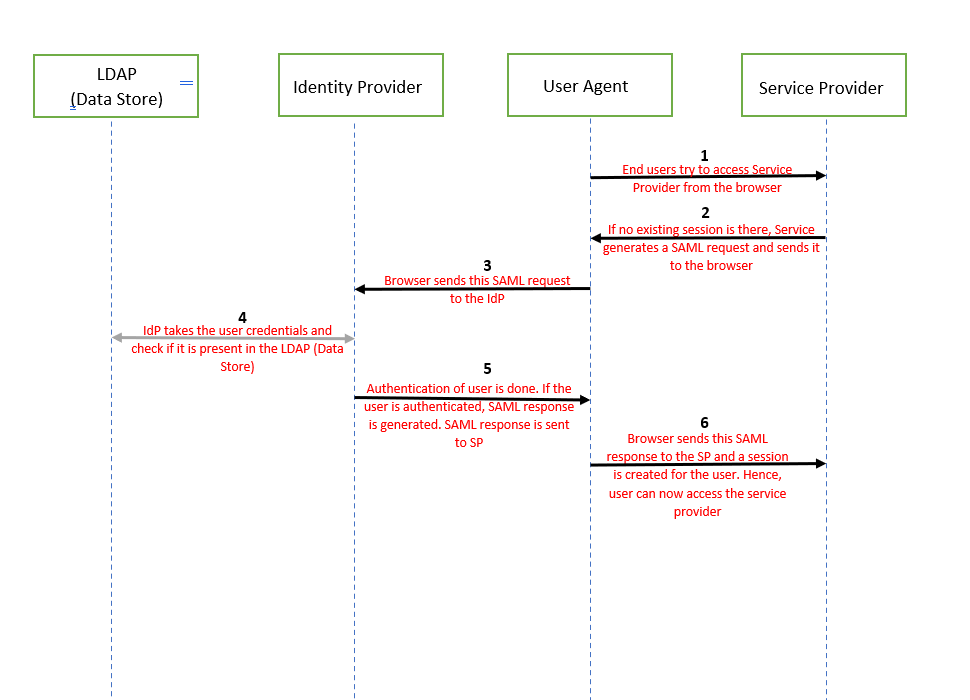


Figure :Sequence Diagram for SP-Initiated SSO

**STEP 1:** User initially reaches out to SP.

**STEP 2:** Since the user is not authenticated so he is redirected to IdP using a request to IdP using an authentication message.

**STEP 3:** IdP further takes the credentials and validate it from Data Store(LDAP).

**STEP 4:** Once the user is validated the IdP generates a SAML assertion.

**STEP 5:** The assertion is sent to SP via the user agent and the session can start.

# CHAPTER-4: PING ACCESS

## 4.1 What is PingAccess?

* Ping Access is an alternative to the Web Access Management Solution.
* Ping Access is used to protect websites, APIs, and other web resources using rules and other authentication criteria. It allows both internal and external users to access web applications securely.

## 4.2 Installation and Configuration

|  |  |  |
| --- | --- | --- |
| **S.No** | **Steps** | **Screenshots** |
| 1. | Request the license key for PingAcces on below URL.  <https://www.pingidentity.com/en/account/request-license-key.html> |  |
| 2. | Go to the link: <https://support.pingidentity.com/s/> and then download the license key under the tab ***Manage License Keys → View → Download*** |  |
| 3. | Inside the VM, go to the URL shown below and download the Linux based Product Distribution (ZIP) file.  <https://www.pingidentity.com/en/resources/downloads/pingaccess.html> |  |
| 4. | Extract the downloaded PingAccess zip file. |  |
| 5. | Go to the bin folder inside the extracted PingAccess folder and add the following line at the beginning of run.sh file:  **JAVA\_HOME="/usr/lib/jvm/java-11-openjdk-amd64"** |  |
| 6. | Now, open the terminal inside the bin folder and type the command shown in the image. |  |
| 7. | Run PingAccess by running the **run.sh** file in the terminal as root |  |
| 8. | Open the link: <https://localhost:9000/> to access the PingAccess console application. Select your previously downloaded license file and import it. |  |

## 4.3 Protocols used in PingAccess

Ping Access uses OAuth 2.0 and OpenID connect. These protocols allow to have a granular access in Ping Access.

### 4.3.1 Terminologies used in OAuth 2.0

1. Resource: It is something that is protected and needs to be accessed by some different service. It is also referred as protected resource.
2. Resource Owner: It is an entity who has the access to the resource.An entity capable of granting access to a protected resource. Generally, user is the resource owner.
3. Resource Server: It is the Server that is hosting the Resource.
4. Client: It is an application making protected resource requests on behalf of the resource owner and with its authorization
5. Authorization Server: Resource server is coupled with an Authorization Server. Authorization Server is responsible for making sure that whoever is accessing the resource server is authorized. Authorization server receives the request for access token from client and after successful authentication, it grants them the access token.

1. Redirect URI: Authorization server redirects the user to a location once application has been successfully authorized and granted an authorization code using URL known as Redirect URI. This process is called as Callback.
2. Access Tokens: Access token is a piece of some data which allows to perform authorization to gain access of resource.
3. Scope: Scopes are list of permissions requested from client side. Scopes are sent from client to authorization server while sending the request to access the resource. They are used to specify the type of access (i.e., read only access or read and modify etc.) to be granted to client. Multiple scopes can also be sent.
4. Consent: Consent is used by Authorization server to user asking whether client can do the actions mentioned in the scope.
5. Back channel: It is a Highly secure channel and used to send request from user’s server to other API server from backend. No one can decrypt the information.
6. Front channel: It is Less secure channel and used to send request from browser. As browser might have loopholes like putting secret key in web app inside html then one can see it using view source with inspect element or with chrome developer tools etc.

### 4.3.2 How OAuth 2.0 Works?

Let’s understand the OAuth 2.0 standard with the help of a use case.

1. Resource owner (i.e. user) will be at Client website and for our use case let’s assume the client website to be **Zomato.com**. Now this website zomato.com needs to access the contacts present in Gmail (Google account. When Resource owner selects the option **‘Connect with Google’** in client website, it is redirected to Authorization server (accounts.google.com be the

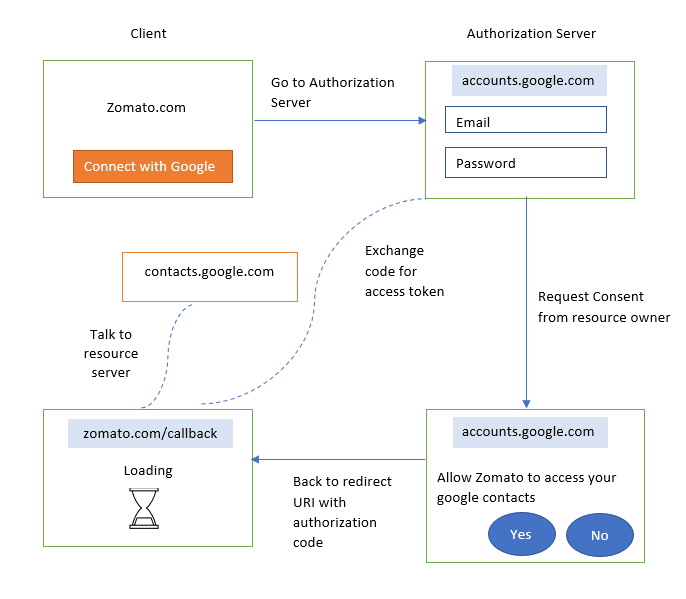


Figure : Working of OAuth 2.0

1. Authorization Server) and While Redirecting, configuration such as Redirect URL, Response type and Scope is passed from Client to Authorization Server.
2. Authorization Server requests consent from resource owner to either grant or deny the permission.



1. If consent is granted from the resource owner, then Authorization server redirects to an URL known as Redirect URL along with authorization code.
2. Client goes back to authorization server and exchanges this code for access tokens.
3. Client contacts the resource server with access token that contains the information regarding actions (read only) and after successful validation of access tokens by the resource server, client can access the requested resource.

### 4.3.3 How OAuth 2.0 and OpenID connect works together?

* Since OAuth was designed only for authorization so there is no standard way for getting user’s information. Hence OIDC is used to authenticate the users.
* It is an extra identity layer added on top of the OAuth. It allows to check the identity of the user based on authorization done by authorization server.

Figure : OAuth2.0 and OIDC Works together

* + In this we can fetch additional info about end user
  + OpenID connect is for Authentication and OAuth 2.0 is for authorization.
  + OpenID adds the following to OAuth 2.0:
* ID token
* If more information is required, then it connects with user information endpoint.
* Standardized implementation
* Standard set of scopes

# CHAPTER-5: USE-CASES

## 5.1 Use-Case I: Group Allocation of users in Ping Directory

In an organization,

|  |  |  |
| --- | --- | --- |
| **S. No** | **Steps** | **Screenshots** |
| 1. | Open Ping Directory CLI in terminal in bin folder and run ldapadd command. |  |
| 2. | Enter password and write ldif queries to create new ou. |  |
| 3. | Add new cn of groups under “position” ou. |  |
| 4. | Link existing user with above created fulltime group. |  |
| 5. | Link existing user with above created intern group. |  |
| 6. | Use ldapadd commands to search the user through the filter of the house |  |

## 5.2 Use-Case II:

IdP and SP initiated SSO using IAM showcase.

### 5.2.1 Creating Adapter and PCV

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Steps** | **Screenshots** |
| 1. | Go to Password Credential Validator (PCV) in System in Ping Federate console. |  |
| 2. | Make the following changes as shown in the figure. New PCV will be created. |  |
| 3. | Now, go to IdP Adapter inside authentication in Ping Federate console to create a new Adapter. |  |
| 4. | Make the following changes in the Create Adapter Instance as shown in the figure. Make sure to connect the adapter the PCV just created.  *Adapter acts as an intermediate between IdP and SP. It is Adapter that takes the user credentials during authentication and checks if it is present in PCV. PCV in simple terms, acts as an intermediate between Adapter and data store. In this case, we are not adding user from data store, instead we are creating a new user in PCV.* |  |

### 5.2.2 Creating SP Connection

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Steps** | **SCRE** |
| 1. | Click on the link:  <https://sptest.iamshowcase.com/> and then click on IdP initiated SSO under Instructions option. |  |
| 2. | Download the metadata file by clicking on the button as shown in figure. |  |
| 3. | Open Ping Federate console and go to Applications. |  |
| 4. | Click on SP connections and then create a new connection. |  |
| 5. | Now select the option to not to use a template for this connection and click on next. |  |
| 6. | Select the browser SSO profiles and click on Next. Then select the Browser SSO option and click on Next. |  |
| 7. | Click on the file option and select the xml meta file that we downloaded from IAMshowcase website. Click on Next. |  |
| 8. | In Browser SSO, select the following settings in IdP Mapping inside Assertion creation. |  |
| 9. | Make the following changes in Assertion creation settings in Browser SSO. |  |
| 10. | Make the following changes in Protocol settings inside Browser SSO |  |
| 11. | View the summary of the settings made in Browser SSO and click on Done. |  |
| 12. | Make the following changes in the Credentials. (If required, create a new certificate). |  |
| 13. | **IdP-initiated connection** is ready. Click on this url. It will redirect directly to IdP for authentication and after authentication will redirect us to SP. |  |
| 14. | Now, export the metadata file of the created IdP connection. |  |
| 15. | Go to the link: https://sptest.iamshow case.com/instructions #spinit and upload the downloaded metadata file. |  |
| 16. | Copy and paste this url in new browser. Hence, **SP-initiated connection** is ready. |  |

## 5.3 Use-Case III:

Protection of Salesforce application using Ping Federate

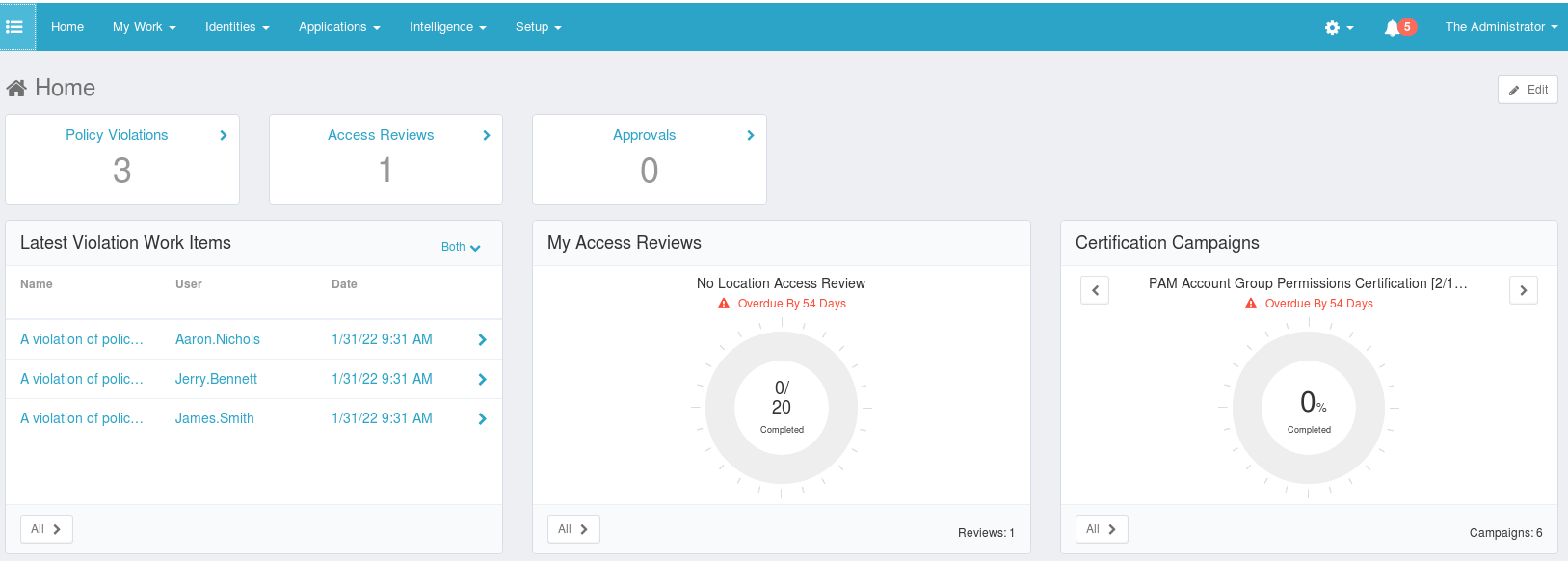
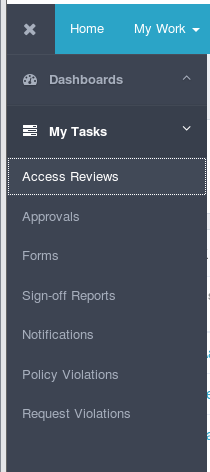
|  |  |  |
| --- | --- | --- |
| **S.No.** | **Steps** | **Screenshots** |
| 1. | Inside Ping Federate console, go to Data Store in System to link our Ping Directory with Ping Federate. |  |
| 2. | Create a new Data Store and connect the Ping Directory with Ping Federate as shown in the figure. |  |
| 3. | Go to Password Credential Validator (PCV) in System in Ping Federate console. |  |
| 4. | Make the following changes as shown in the figure. Make sure toc connect the right data store with PCV. |  |
| 5. | Now, go to IdP Adapter inside authentication in Ping Federate console to create a new Adapter. |  |
| 6. | Create a new adapter with PCV that we create for the data store (Ping Directory Data Store). |  |
| 7. | Now, go to System à Protocol Metadata à Metadata Export. |  |
| 8. | Make the following changes in the settings and export the meta data file of Ping Federate. |  |
| 9. | Login into Salesforce: [Login | Salesforce](https://login.salesforce.com/) and then go to setup à single-sign on. Select the SAML enabled checkbox and create a SAML SSO by inserting the PF metadata file and then download the Salesforce metadata file. |  |
| 10. | Then go to My domain in setup and make the following changes as shown in the figure. |  |
| 11. | Inside my domain, in my domain details, current my domain url is the SP-initiated SSO url. |  |
| 12. | Now, go to Ping Directory and create a new user. |  |
| 13. | Go to setup à users and create a new user. Enter the same details that have been entered for creating a new user in Ping Directory. |  |
| 14. | Create a new SP connection in Ping Federate as shown in figure.  IdP-initiated connection with be built with Salesforce. |  |

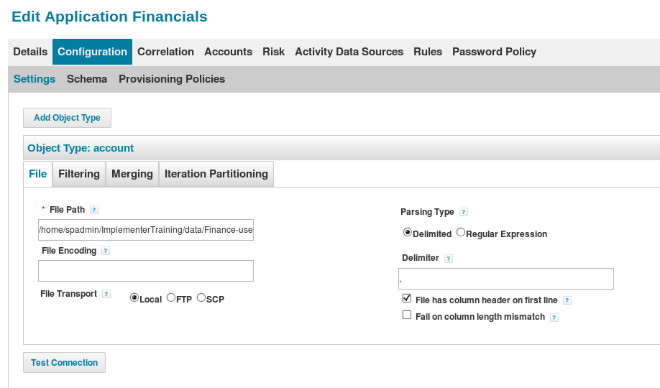
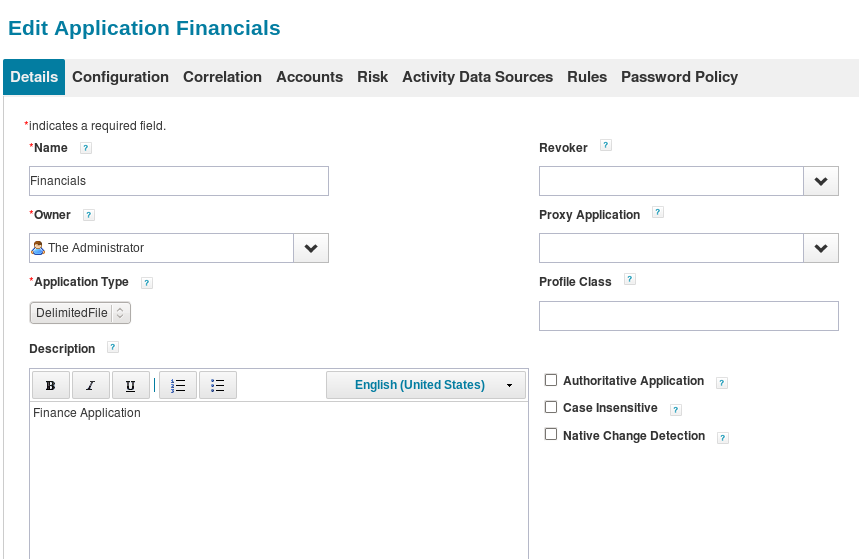
Hence, IdP and SP initiated connection will be created with Salesforce. It will authenticate user from Ping Directory.

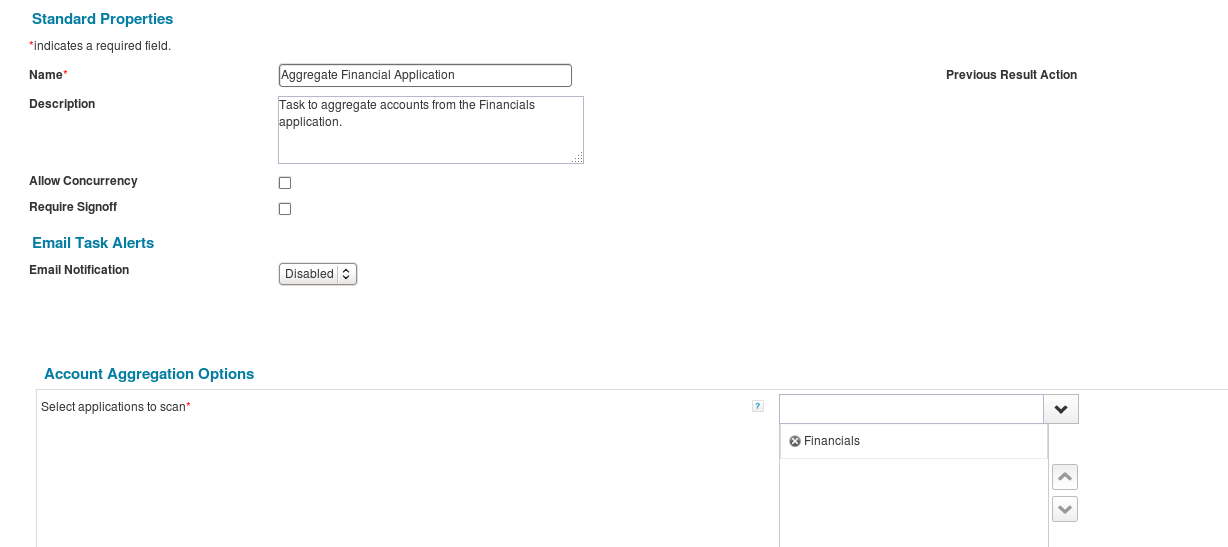
In IdP initiated connection, after clicking on the url we will be redirected to Ping Federate for authentication. After entering the details of the user that we created in Ping Directory and Salesforce, we will be redirected to Salesforce application.

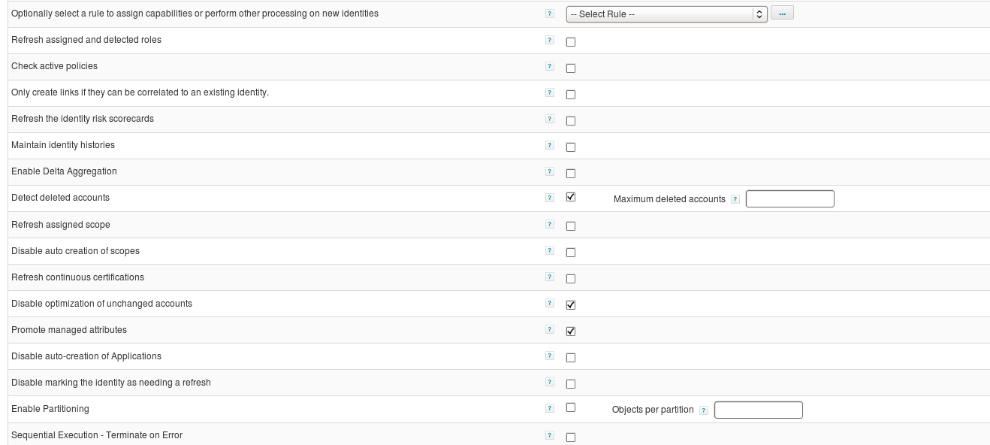
In SP initiated connection, after clicking on the url we will be first redirected to Salesforce login page along with an option for SSO. After clicking on SSO option, it will take us to Ping federate (integrated with Ping Directory) for authentication. After authentication, we will be redirected to Salesforce application account.

|  |  |
| --- | --- |
| **Abbreviation** | **Expanded Form** |
| LDAP | Lightweight Directory Access Protocol |
| SAML | Security Assertion Mark-up Langauage |
| OAuth | Open Authorization |
| OIDC | OpenID Connect |
| SP | Service Provide |
| IAM | Identity Access Management |
| RBA | Role Based Access |
| SSO | Single Sign-on |
| PCV | Password Credentials Validator |

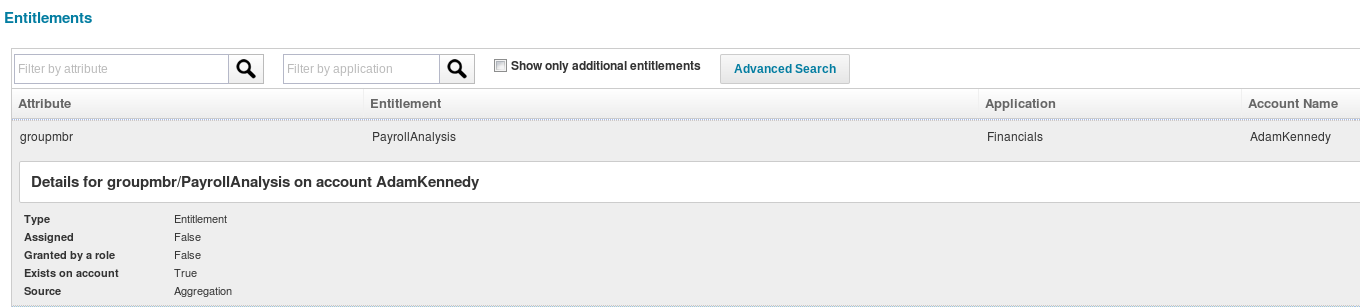


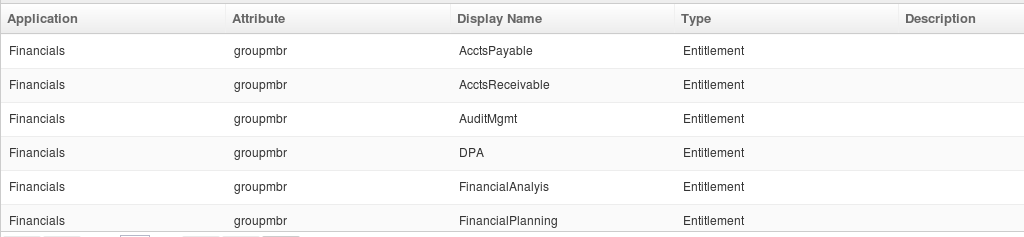


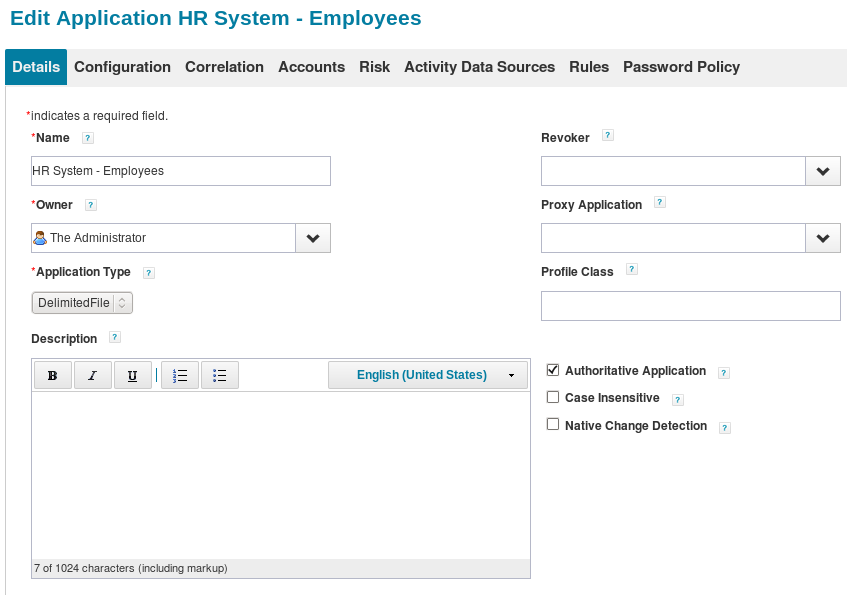
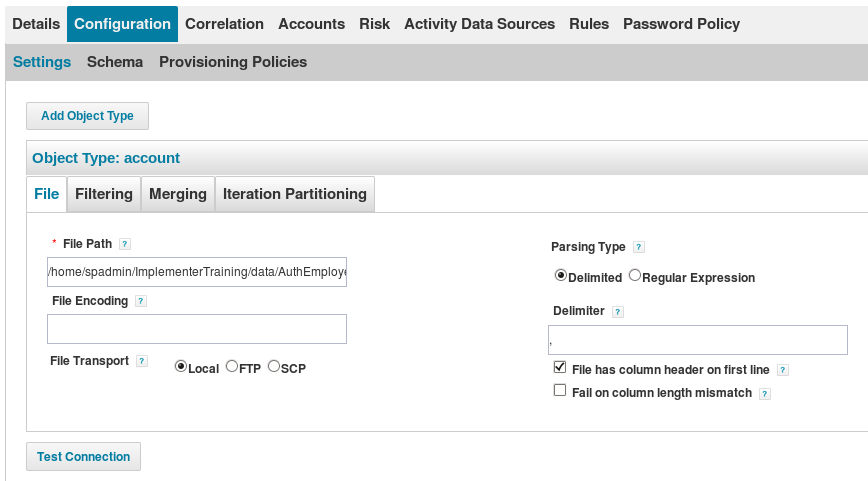
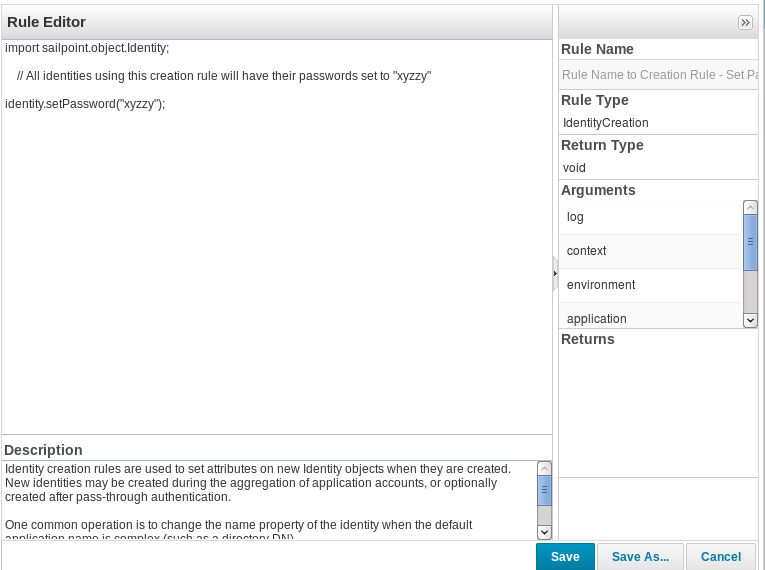
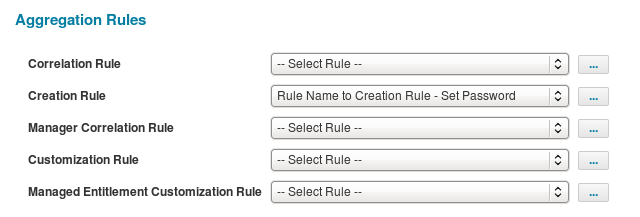
****

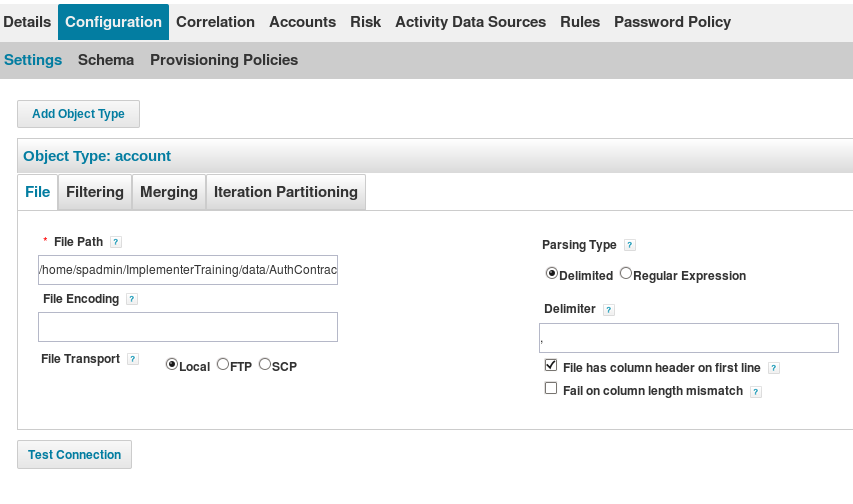
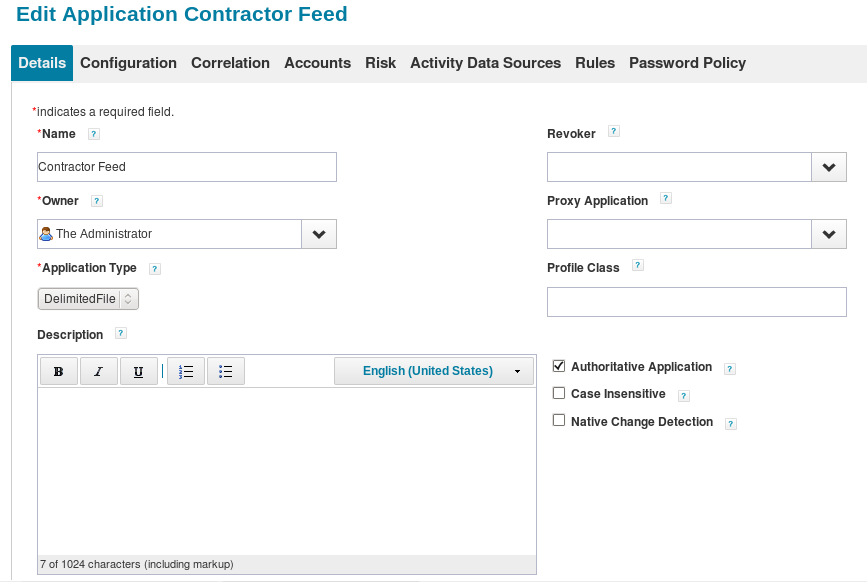
****

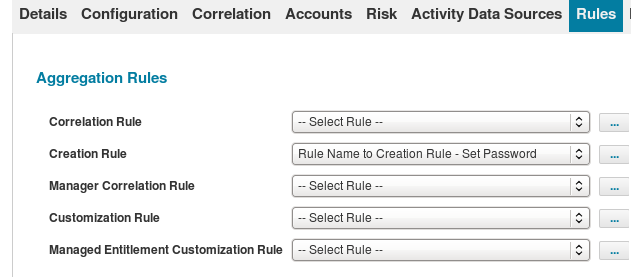
****

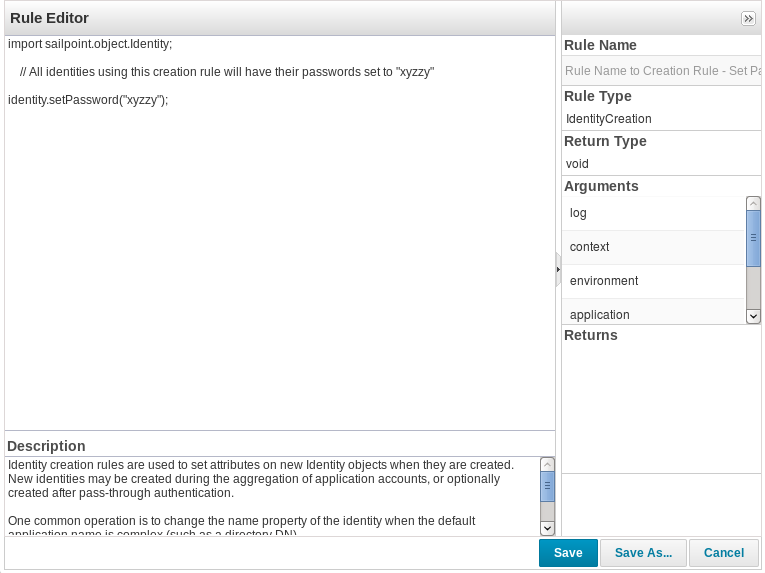
****

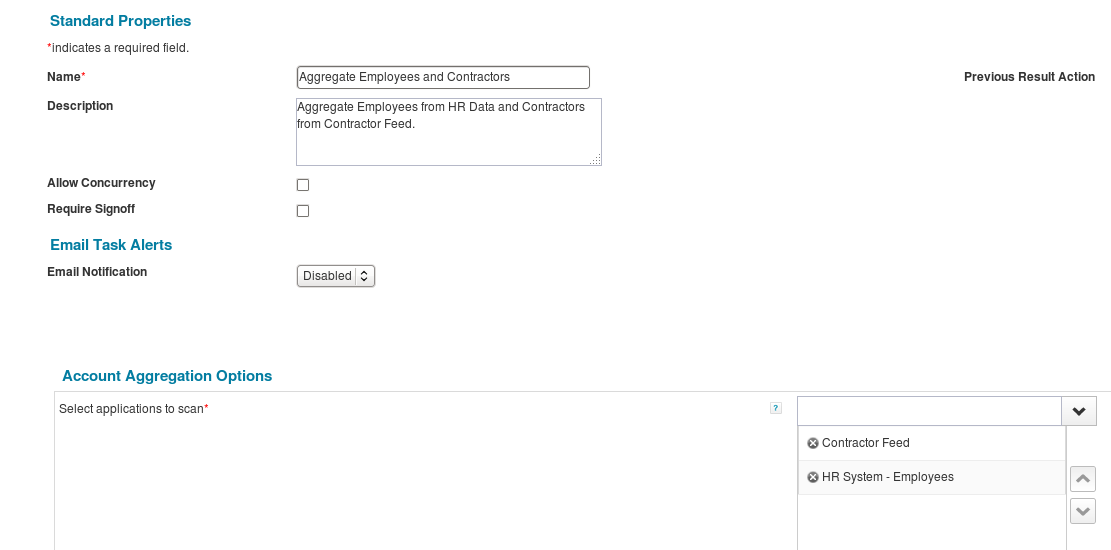
****

****

****

****

****

****

**Model**

Updates

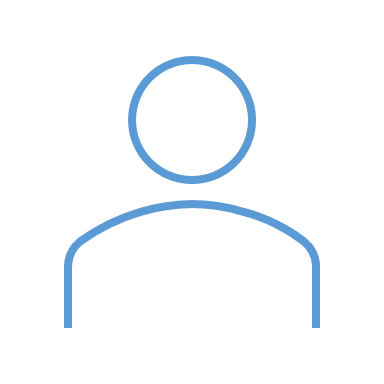
Manipulates

**Controller**

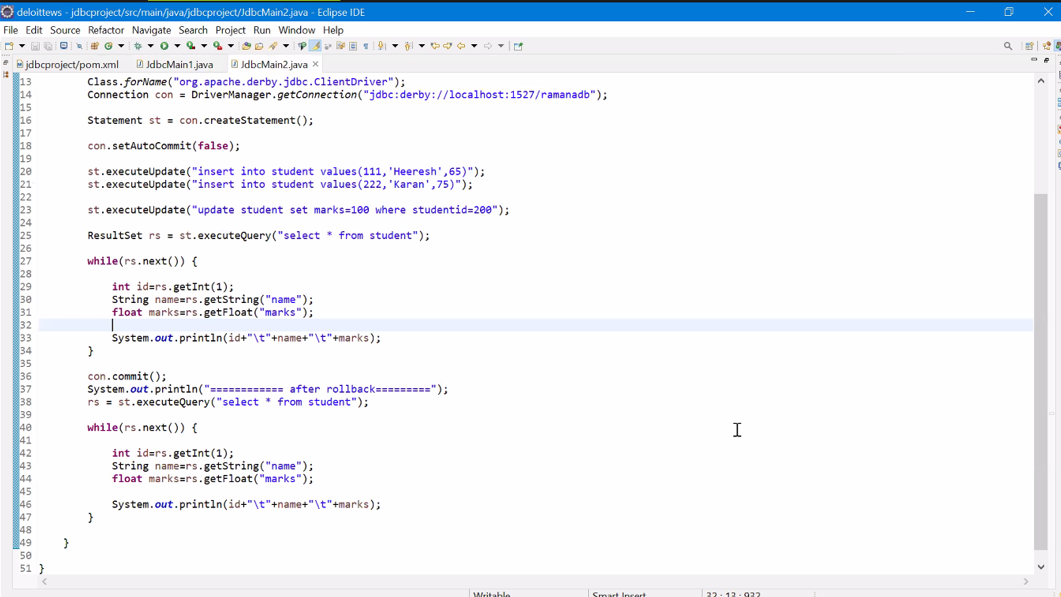
**View**

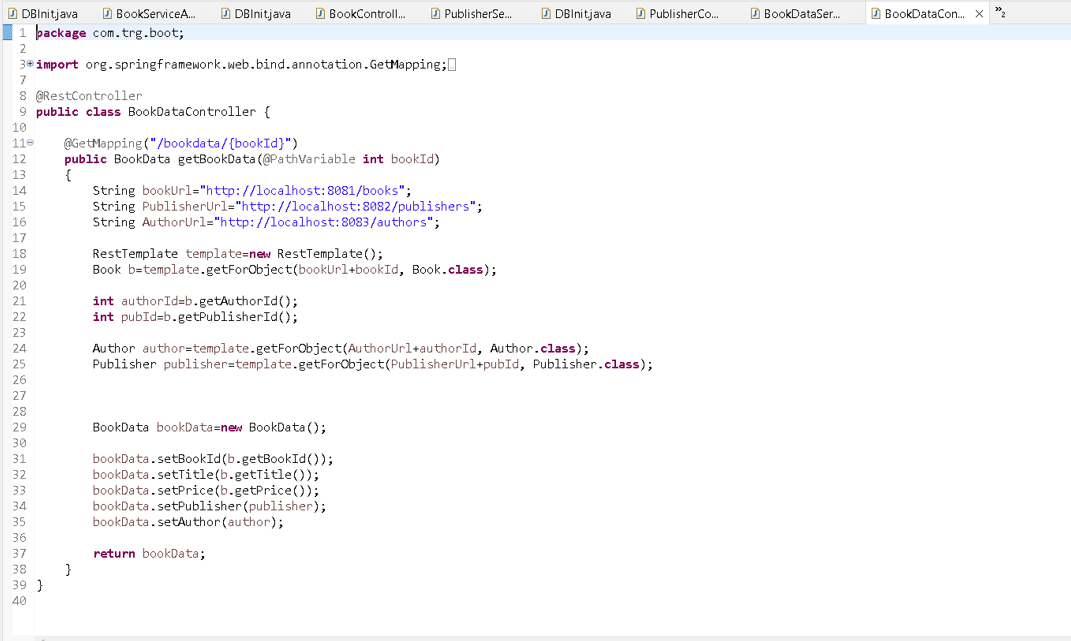
Uses

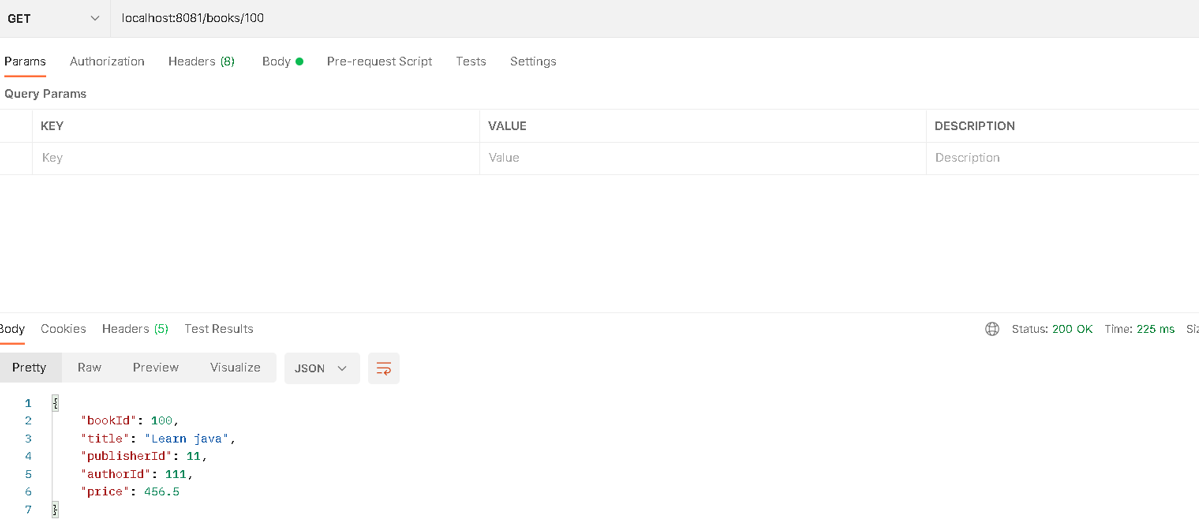
Sees

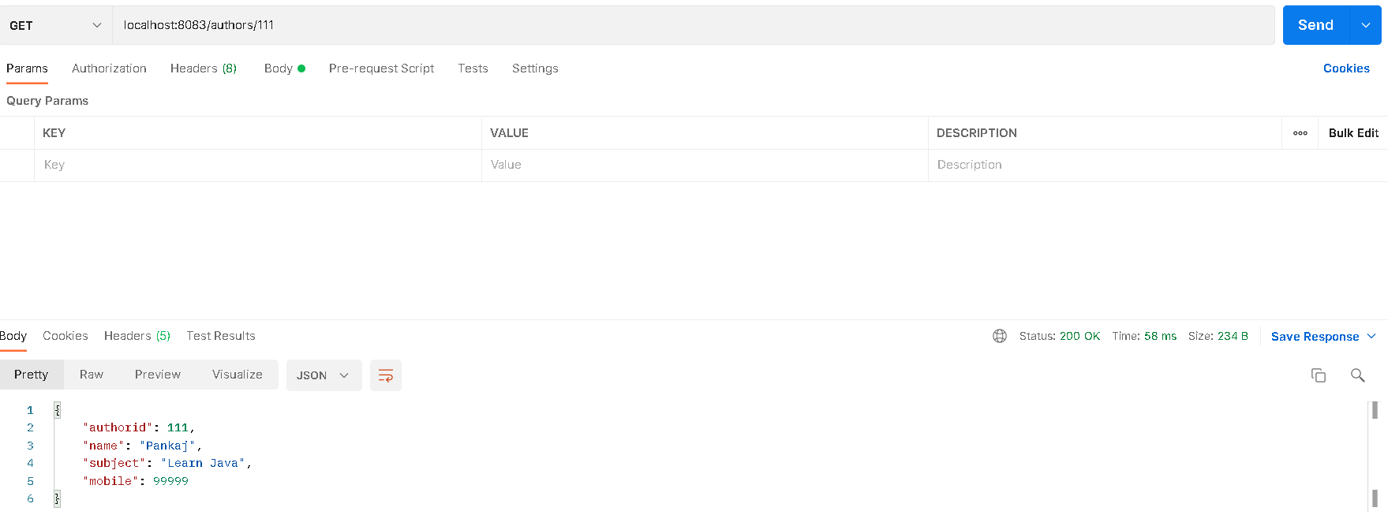
****

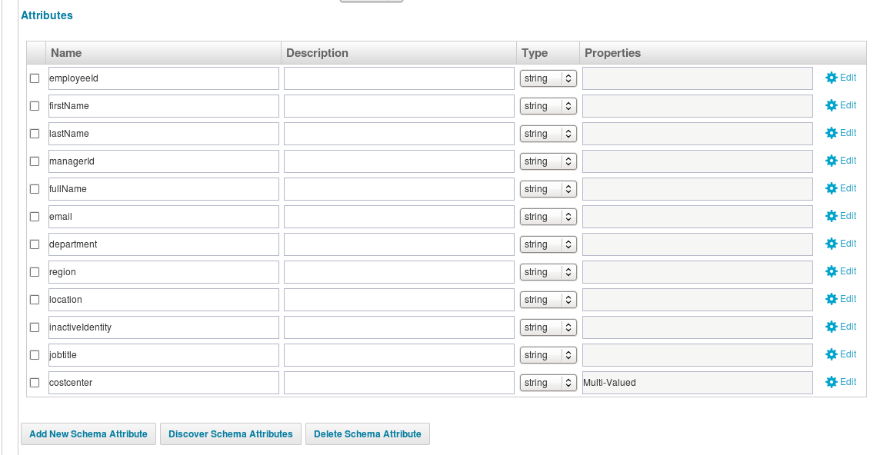
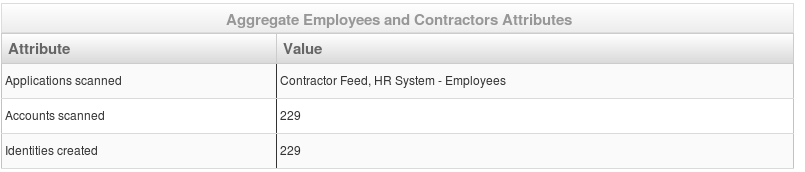
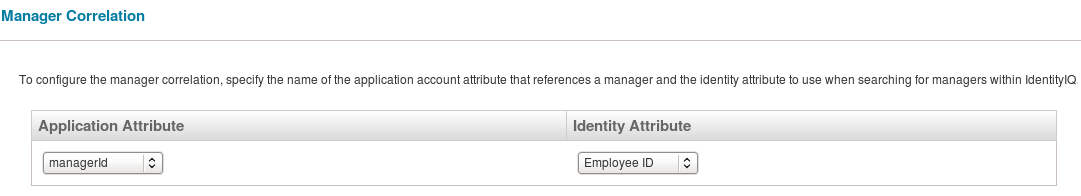
User

****

****

****

****

****