Okta Technical Consultant Boot Camp 2

Lab Guide

Printed April 2017

Contact Okta, Inc @ training@okta.com

Table of Contents

Technical Consultant Labs

Notes:

- All labs are to be performed on your laptop using any web browser.

The Company

Okta Ice has been in the ice cream business for 15 years. In the first year, they only had two stores, but have grown to 30 stores across California. Between the 30 stores there are approximately 800 workers and a recent acquisition of an ice cream truck business means additional head count. While the operators of the ice cream trucks are independent contractors, they are remote and require mobile access to various company applications. Apart from the new head count issues, Okta Ice has a social media presence that they want to grow and promote. Eventually, they also want to expand to the Eastern US and into the gelato market in Europe.

Challenges

- Employees and contractors require access to several applications with separate login pages; requiring them to remember multiple usernames and passwords.
- The ice cream truck owners and employees must be able to access applications from outside of the office using different devices.
- Okta Ice must be able to authenticate and access information outside of the office with an additional authentication factor.
- If an employee or contractor leaves Okta Ice, the associated access to applications must be efficiently disabled; this is currently a lengthy manual process.
- Vendors require access to certain applications for ice cream manufacturing.

Organization Configurations

Okta User Accounts

Okta Administrators	Contractors
	Kay West
Okta Admin	Fred Jones
Bob Jones	Mike Barnes
Karen Smith	Sharon Sims
Jane Young	George Bliss
	Chris Bell

Active Directory Configuration

OUs	Users	Groups
	Adam Willems	EMEA; Sales
	Alex Smit	US West; Engineering
	Ana Walters	US West; Sales; Management
	Catherine Dunn	US West; Marketing
	Edith Jansen	US East; Sales
	Emily Boone	US West; HR; Management
	Erin Richardson	US East; HR
	Frank Molen	EMEA; Sales
	Gerald Miles	US West; Engineering
	Jack Bailey	US West; Engineering
Employees	James Parks	US West; Marketing; Management
	Jennifer Jones	US West; Engineering
	Joseph Baker	US West; Engineering; Management
	Kent Vasquez	US East; Sales
	Martin White	US East; Sales
	Matthew Smith	EMEA; Sales
	Michael Black	US West; Management
	Nate Abbott	US West; Engineering
	Oliver Banks	US West; Marketing
	Sarah James	US East; Engineering; Management
	Stephen Kim	US West; Engineering

Partners	Diane Smith Hope Valley John James Josh West Sam Finnegan Sarah Wood	
Interns	Becky King Faith Hunt Frank Snyder Patrick Peterson Wendy Nelson	EMEA; Marketing US West; Marketing US West; Engineering US West; Engineering EMEA; Sales

Labs

Lab 1-1: Configure an Application as a Master

Objective	Implement Salesforce as a Master and create Active Directory Accounts
Scenario	Okta Ice has decided that their Salesforce tenant is the source of truth for all account data and would like to use it as the Master. They would then like to have those accounts pushed to Active Directory.
Duration	30-45 minutes

Note: Complete this lab in the VM and on your host Windows server for the agent.

Create the Okta Admin Account

1. Login into your Okta Ice Org, as follows:
    a. **Org**: https://oktaice###.oktapreview.com
    b. **Username**: **oktatraining@okta.com**
    c. **Password**: Instructor will provide
2. Answer the forgot password question and select a security image.
3. Click **Create My Account**.
4. In the top menu, point to **Directory** and click **People**.
5. To create an Okta administrator account to use as a service account, click **Add Person**.
6. Complete the mandatory fields, as follows:

Field	Value
First name	**Okta**
Last name	**Admin**
Username	**okta.admin@oktaice.com**
Primary email	**okta.admin@oktaice.com**
Secondary email	**oktaice###@mailinator.com***

*If you are unable to access mailinator.com, then use an email address that you can access.

7. Select **Send user activation email now**.
8. Click **Add Person**.

Activate the Okta Admin Account

1. Open a new browser tab.
    a. If you used a mailinator email address:
        i. In the address bar, type the following:
        https://www.mailinator.com/
        ii. In the **Check Any Inbox** field, type the following and then click **GO!**:
        oktaice###
    b. Access your email account.
2. Open the **Welcome to Okta!** email.
3. In the email, to activate the account, click the activation link.
4. Specify the instructor-provided password in the **Enter new password** and **Repeat new password** fields.
5. In the **Choose a forgot password question** list, select a password reset question and provide an answer.
6. Under **Click a picture to choose a security image**, select an image.
7. Click **Create My Account**.
   You are automatically redirected to the End User home page.
8. Sign out of Okta.

Assign Administrator Permissions

1. Sign back into Okta with the **oktatraining** credentials.
2. In the Okta Administrator application, point to **Security** and click **Administrators**.
3. Click **Add Administrator**.
4. Perform the following tasks in the **Add Administrator** dialog box:
    a. In the **Grant administrator role to** field, type and select the following:
    Okta Admin
    b. Under **Administrator roles**, select **Super Administrator**.
    c. Click **Add Administrator**.

Install the Agent

1. Within your VM, open a web browser.
2. In the Address bar, type your Okta org.
3. On the **Okta Sign In** page, sign in with the **okta.admin** credentials.
4. If necessary, click **Admin**.
5. Point to **Directory** and click **Directory Integrations**.
6. Click **Add Directory** and then click **Add Active Directory**.

7. Review the **Set Up Active Directory** page, scroll to the bottom, and click **Set Up Active Directory**.
8. On the **Download Agent** page, click **Download Agent**.
9. In the Windows Explorer **Downloads** folder, right-click the **OktaADAgentSetup.exe** file and then click **Run as administrator**.
10. If a security warning dialog box appears, click **Run**.
11. Complete the **Okta AD Agent** installation wizard as follows:
    a. In the **Okta AD Agent** dialog box, click **Next**.
    b. Leave the default installation folder and click **Install**.
    c. Leave the domain for the user accounts as oktaice.local and click **Next**.
    d. For this lab, leave the default OktaService account selection and click **Next**.
    e. In the **Password** and **Retype password** fields, type the instructor-provided password.
    f. Click **Next**.
    g. Do not configure any AD agent proxies; click **Next**.
    h. On the **Register Okta AD Agent** page, select **Preview** and in the **Enter Subdomain** field type your subdomain.
    i. Click **Next**.
    j. When prompted, sign into Okta with the **okta.admin@oktaice.com** account.
    k. When prompted to grant access to the Okta API, click **Allow Access**.
    l. When the installation completes, click **Finish**.
12. In the **Agent Installation** dialog box, click **Next**.
    You are returned to the Okta Administrator app to complete the Active Directory configuration.
13. On the **Basic Settings** tab, perform the following:
    a. Review the recognized OUs and select the OUs containing the **Users** (top section) and **Groups** (bottom section) containing the resources.
        i. In the **Users** section, select **Employees**, **Partners**, and **Interns**.
        ii. In the **Groups** section, select **Employees**, **Partners**, and **Interns**.
    b. Leave the default selections including User Principal Name as the Okta username format.
    c. Click **Next**.
14. In the **Import AD Users and Groups** dialog box, click **Next**.

15. On the **Build User Profile** tab, perform the following:
    a. In the **Search** field, type and then select the following attribute: **favoriteIceCreamFlavor**
    b. Click **Next**.
16. On the **Done** tab, review the information and click **Done**.
    The page refreshes to the Settings tab for the Active Directory instance.

Import Users from Active Directory

1. Click the **People** tab.
   Notice that no user records appear. This is because the import has not yet occurred.
2. Click the **Import** tab.
3. Click **Import Now**.
4. In the **Import from Active Directory** dialog box, select **Full import** and then click **Import**.
   A dialog box appears indicating how many users and groups were scanned.
5. Click **OK**.
   If you were configuring Active Directory for JIT provisioning, you would stop here because the accounts have been scanned and now only require users to log in for activation.
6. In the **Show** list, select **50**.
7. In the top-right corner of the table and to the right of the **Okta User Assignment** label, click the **Select All** box.

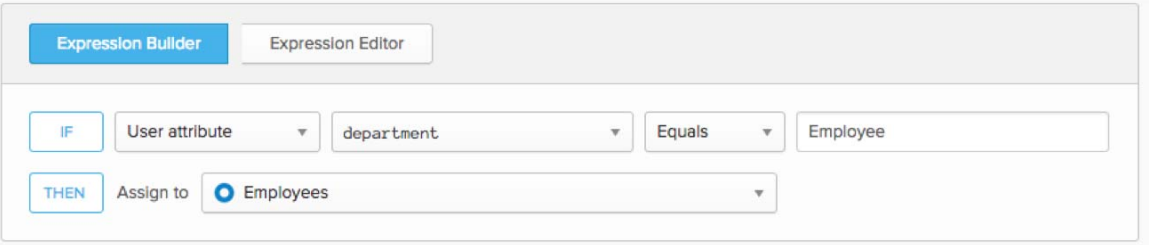
8. Click **Confirm Assignments**.
9. In the **Confirm Imported User Assignments** dialog box, select **Auto-activate new users after user confirmation** and then click **Confirm**.
10. Click the **People** tab.
    All Active Directory users should appear.

Configure Push Accounts to Active Directory

1. Click the **Settings** tab.
2. Under **Provisioning Features**, next to **Create Users**, select **Enable**.
3. In the **Activation email recipient** field, type the following: **oktaice###@mailinator.com**
4. Next to **Update User Attributes**, select **Enable**.
5. Next to **Deactivate Users**, select **Enable**.
6. Click **Save Settings**.

Create Groups and Rules

1. Point to **Directory** and click **Groups**.
2. Click **Add Group**
3. In the Add Group dialog box, perform the following:
    a. In the **Name** field, type the following:
    Contractors
    b. Click **Add Group**.
4. On the **Groups** tab, click **Add Group**.
    a. In the **Name** field, type the following:
    Employees
    b. Click **Add Group**.
5. On the **Groups** tab, click the **Rules** sub tab.
6. Click **Add Rule**.
7. In the **Name** field, type the following:
    Contractors
8. On the **Expression Builder** tab, perform the following:
    a. In the **Select an attribute** list, type and select **department**.
    b. Next to **Equals**, in the **Enter a value** field, type the following:
    Contractors
    c. Next to **Assign to**, in the **Select a group** field, type and then select the Okta **Contractors** group.
    d. Click **Add Rule**.
9. Under the **Status** column, for the **Contractors** rule entry, click **Inactive** and then click **Activate**.
10. Click **Add Rule**.
11. Repeat steps 8 to create an **Employees** rule.
    The rule should be as follows:



12. Under the **Status** column, for the **Employees** rule entry, click **Inactive** and then click **Activate**.
13. Click the **All** sub tab.
14. Search for and select the **Employees** group.
15. Click **Manage Directories**.

16. Click **Add All**.
17. Click **Next**.
18. Under **Default** Attributes, in the **Organizational Unit** list, select **employees**.
19. Click **Confirm Changes**.

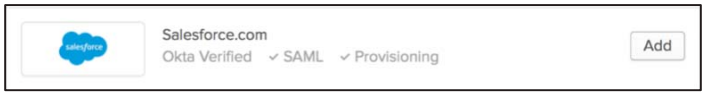
Promote the Okta Active Directory Agent Service Account

This portion of the lab is done to support Active Directory account creates.

1. In your VM, in the **Windows taskbar**, click the **Start** icon.
2. In the top-right corner, click the **Search** (magnifying glass) icon.
3. In the search field, type and select the following:
   Okta AD Agent Manager
4. In the **Okta AD Agent Manager Utility** window, expand **Service Account**.
5. Click **Add to Domain Admins group**.
   The agent stops and restarts.
6. Close the **Okta AD Agent Manager Utility** window.

Create the Salesforce Application

1. Return to the Okta Administrator application.
2. Point to **Applications** and click **Applications**.
3. Click **Add Application**.
4. In the **Search for an application** field, type the following:
   salesforce
5. Next to the **Salesforce.com** entry, click **Add**.



   A dialog box appears.
   a. On the **Enter basic info** page:
      i. Change the default **Application label** to **SFDC as a Master**.
      ii. In the **Instance Type** list select **Sandbox** and click **Next**.
   b. On the **Configure sign on settings** page:
      i. In the **Default username** list, select **Custom**.
      ii. In the custom string field, type the following:
      ${f:substringBefore(user.email, "@")}@oktaice###.com
      Where ### is your Okta Ice org number.
      iii. Click **Save and Assign**.
      The wizard completes and the Okta Administration app refreshes to the Assignments tab of the Salesforce application.
6. Click the **General** tab.

   a. Next to **App Settings**, click **Edit**.
   b. Next to **Application Visibility**, perform the following:
      i. Select **Do not display application icon to users**.
      ii. Select **Do not display application icon in the Okta Mobile App**.
   c. Click **Save**.
7. Click the **Provisioning** tab.
   a. Click **Enable Provisioning**.
   b. Select **Enable provisioning features**.
      The page expands down.
   c. Under **API Credentials**, enter your administrator credentials.
      i. **Username**: **oktatraining@okta.com.sandbox100**
      ii. **Password**: Instructor will provide
      iii. **Token**: Instructor will provide
   d. Click **Test API Credentials**.
   e. Next to **Profile Master** select **Enable**.
   f. Click **Save**.

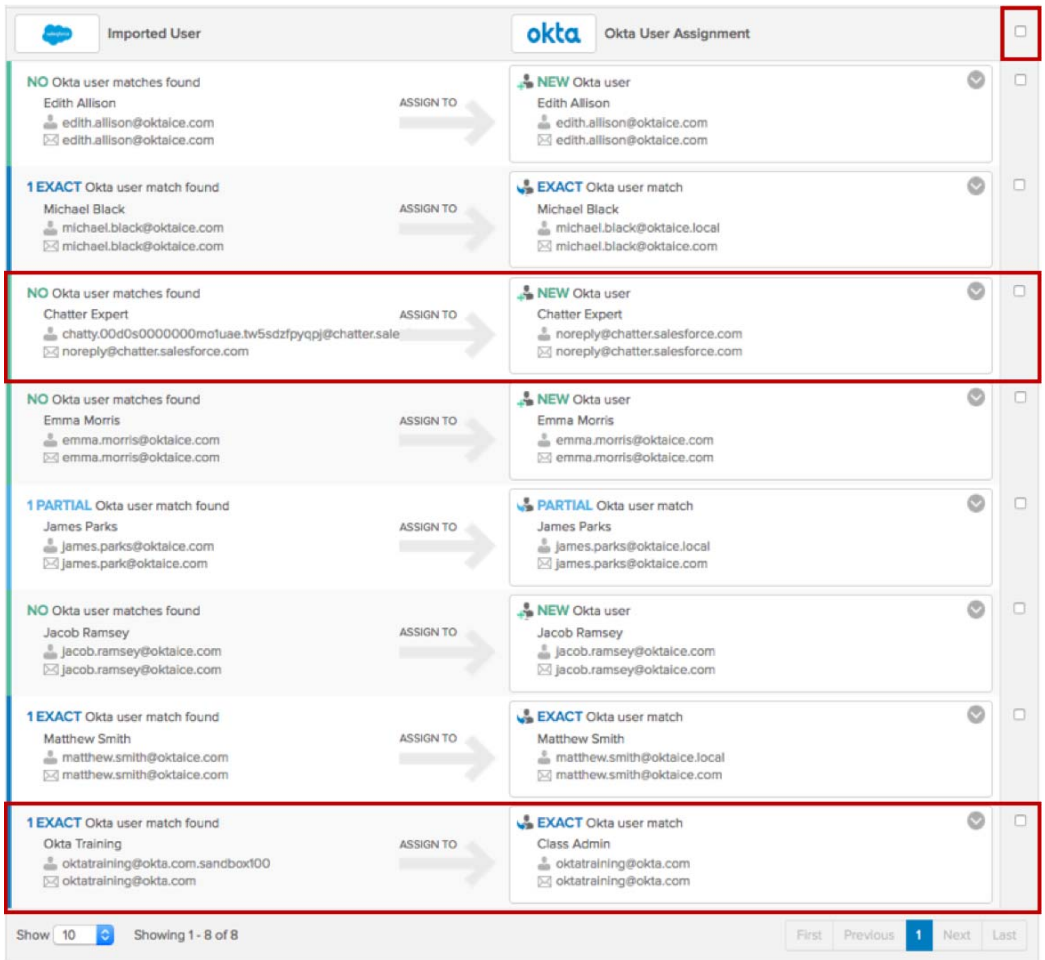
Set the Profile Master

1. Point to **Directory** and click **Profile Masters**.
2. Next to the **Salesforce.com** entry, under the **Priority** column, to make the application the first priority, click the **up arrow**.
3. Click **Confirm**.

Import Users from Salesforce.com

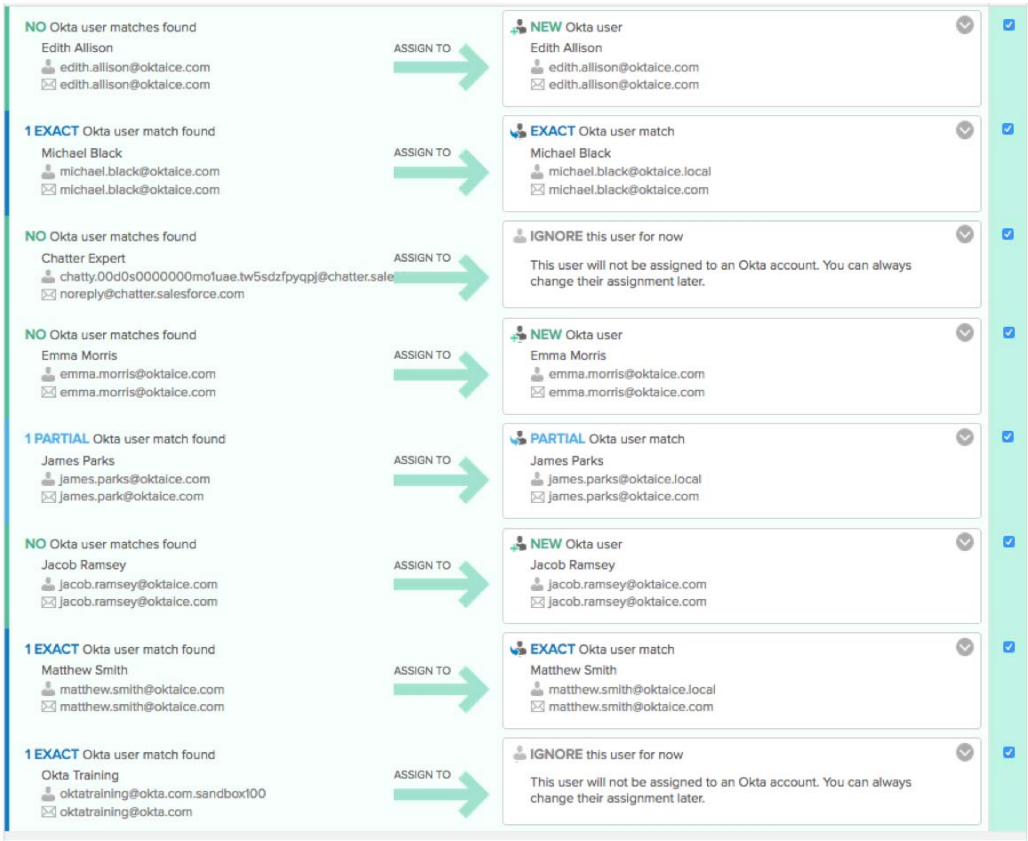
1. Point to **Applications** and click **Application**.
2. Click **SFDC as a Master**.

3. Click the **Import** tab.
    a. Click **Import Now**.
    A dialog box appears indicating how many users and groups were scanned.
    b. Click **OK**.
    Your results should look like this:



    c. Next to the **Chatter Expert** and **Okta Training** users, click the **drop-down arrow** next to the name and click **Ignore**.

d. To confirm the remaining users, select the **top-right box** and then click **Confirm Assignments**.



e. In the Confirm Imported User Assignments dialog box, perform the following:
   i. Select **Auto-activate users after confirmation**.
   ii. Click **Confirm**.

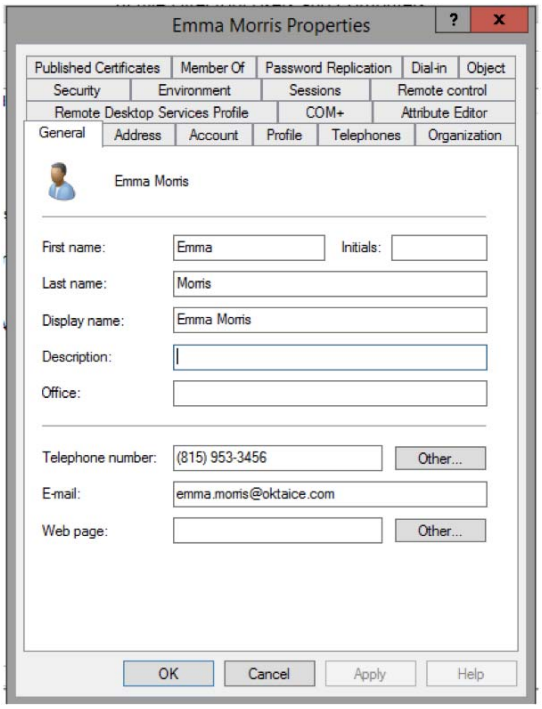
Verify Results

1. In the Okta Administrator app, point to **Directory** and click **People**.
   Edith Allison should be in password reset mode and Emma Morris and Jacob Ramsey should be in active status.
   Note: If the account statuses have not changed, confirm the Okta AD Agent is started; you might have to restart it.
2. In your VM, to verify Emma and Jacob are in your **Employees** OU open the Active Directory **Users and Computers**  .
3. Navigate to mailinator.com and enter in the admin email address: oktaice###@mailinator.com
   There should be two emails that have a subject of:
   Okta user pushed to Active Directory

Lab 1-2: Configure an Application as a Master - Troubleshooting

Objective	Troubleshooting
Scenario	The Active Directory Administrator is verifying the data that was created in the new accounts from Salesforce.com and the extension is not being sent to Active Directory
Duration	15-20 minutes

The Active Directory administrator has sent you the following screenshot wanting to know why the Extension for the phone number was not sent to Active Directory.



Where do you start to troubleshoot this issue?

How do you resolve this issue?

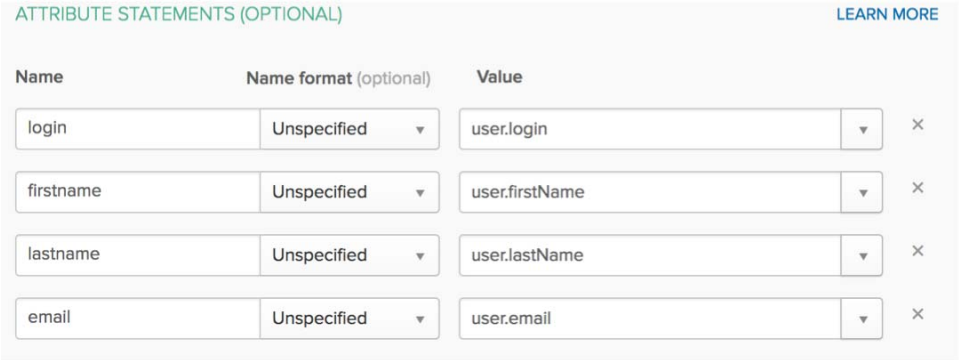
Which step from the previous lab did we not do that would have prevented this issue?

Lab 3-1: Configure Inbound Federation with SAML JIT

Objective	Setup Inbound Federation to access Salesforce for vendors.
Scenario	In the Okta Central org, Salesforce has been configured and each Okta Ice org must access the central Salesforce instance.
Duration	45 - 60 minutes

Create the AIW Application

1. Login to your oktaice###.oktapreview.com org
2. Point to **Applications**, **Applications**
3. Click **Add Application**.
4. Click **Create New App**.
5. Change the **Sign on method** to **SAML 2.0**.
6. Click **Create**.
7. **App Name** - Oktaice###
8. Click **Do not display application icon to users** and **Do not display app icon in Okta Mobile App**.
9. Click **Next**.
10. Complete the fields as follows:
    a. **Single signon URL** - https://placeholder.com
    b. **Audience URI (SP Entity ID)** - https://placeholder.com
    c. **Name ID format** – EmailAddress
    d. In the **Application username** list, select **Custom**.
    e. In the custom string field, type the following:
    ${f:substringBefore(user.email, "@")}@oktaice###.com
    Where ### is your Okta Ice org number.
    f. Add the following attribute statements:



11. Click **Next**.

12. Select **I'm an Okta customer adding an internal app**.
13. Click **This is an internal app that we have created**.
14. Click **Finish**.
15. Click **Assignments**.
16. Click **Assign**, **Assign to People**.
17. Search for Sarah James.
18. Click **Assign**.
19. Click **Save and Go Back**.
20. Click **Done**.
21. Click the **Sign On tab** and then click **View Setup Instructions**.
    You need this information to Configure your Inbound SAML

Configure Inbound SAML

1. Open another browser tab
2. Login to oktacentral.oktapreview.com
    a. **Username**: **oktatraining@okta.com**
    b. **Password**: Instructor will provide
3. Point to **Security**, **Identity Providers**.
4. Click **Add Identity Provider**, **Add SAML 2.0 IdP**.
5. **Name** – your oktaice org name and number.
6. Under the **Authentication Settings**, select **IdP Username** set to **idpuser.subjectNameId** and leave the other default values.
7. Under **JIT Settings**, select the following:
    a. Select **Profile master** and then click **Update attributes for existing users**.
    b. Under **Group Assignments**, click **Assign to specific groups**.
    c. **Specific Groups** - Oktaice
8. Under **SAML Protocol Settings**, perform the following
    a. **IdP Issuer URI** – Copy from the setup instructions from the SAML app that you created – copy the **Identity Provider Issuer**
    b. **IdP Single Sign-on URL** -- Copy from the setup instructions from the SAML app that you created – copy the **Identity Provider Single Sign-On URL**
    c. **IdP Signature Certificate** – download the X.509 Ceritificate from the SAML app that you created – upload it to the **IdP Signature Certificate**
9. Click **Add Identity Provider**.
10. Copy the **Assertion Consumer Service URL**.

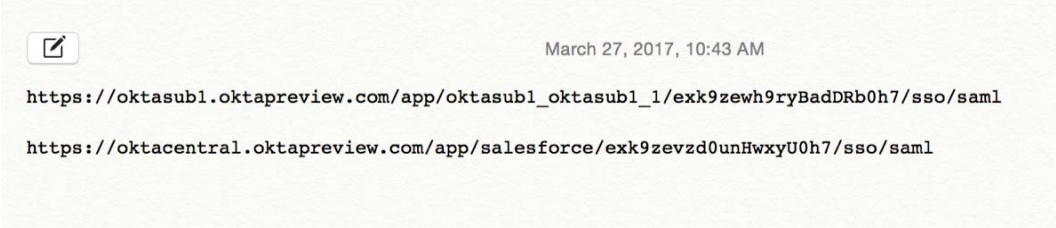
Update the AIW Application

1. Return to your oktaice org.
2. Point to **Applications**, **Applications**.
3. Click the **SAML** application.
4. Click the **General** tab.
5. Next to SAML Settings, click **Edit**.
   This reopens the AIW.
6. Click **Next**.
7. Paste the **Assertion Consumer Service URL** into the **Single sign on URL**
8. Return to the oktacentral org.
9. Copy the **Audience URI** into the **Audience URI (SP Entity ID)** field.
10. Click **Next**.
11. Click **Finish**.

Create the Bookmark Application

1. In Okta Central org, point to **Applications**, **Applications**
2. Click **SFDC – Central**.
3. Click the **Sign On** tab.
4. Click **View Setup Instructions**.
5. Copy the **Identity Provider Login URL** into Notepad.
6. Return to your oktaice org.
7. Navigate to **Applications**, **Applications**
8. Click the **SAML** application.
9. Click the **Sign on** tab.

10. Click **View Setup Instructions**.
    a. Copy the **Identity Provider Single Sign-on URL** into notepad
       For Example:



    b. To the end of the URL that is your oktaice org add the following:
       ?RelayState=
       For example:
       `https://oktasub1.oktapreview.com/app/oktasub1_oktasub1_1/exk9zewh9ryBadDRb0h7/sso/saml?RelayState=`
    c. Next, add the **Signon URL** from Okta Central.
       For example:
       `https://oktasub1.oktapreview.com/app/oktasub1_oktasub1_1/exk9zewh9ryBadDRb0h7/sso/saml?RelayState=`https://oktacentral.oktapreview.com/app/salesforce/exk9zevzd0unHwxyU0h7/sso/saml
11. Return to your Okta Ice org.
12. Navigate to **Applications**, **Applications**.
13. Click **Add Application**.
14. Search for **Bookmark App.**
15. Click **Add**.
16. In the **Enter basic info** dialog box, perform the following:
    a. In the **Application Label** field, type the following:
       SFDC Okta Ice Central
    b. In the **URL** field, paste in the URL from Notepad.
    c. Click **Next**.
17. Click **Save and Assign**.
18. Click **Assignments**.
19. Click **Assign**, **Assign to People**.
20. Search for Sarah James.
21. Click **Assign**.
22. Click **Save and Go Back**.
23. Click **Done**.

Verify Results

1. Launch an incognito window in Chrome or restart your browser.
2. Login as Sarah James to your Okta Ice org and access Okta Central SFDC.

Lab 3-2: Configure Inbound Federation with Org2Org

Objective	Setup Org2Org provisioning.
Scenario	In the Okta Central org Office365 has been setup and each Okta Ice org must provision users and groups to the Central Office365 application.
Duration	45 - 60 minutes

Create the API Token

1. Login to the Okta Central Org
2. Navigate to **Security**, **API**
3. Click **Create Token**
4. Enter in your Okta Ice org as the name
5. Click the **Token** tab and then click **Create Token**
6. Copy the token value into Notepad
7. Click **OK, got it**

Configure the Org2Org Application

1. Login to your Okta Ice org.
2. Point to **Applications**, **Applications**.
3. Click **Add Applications**.
4. Search for Org2Org.
5. Click **Add.**
6. In the **Enter basic info** dialog box, perform the following:
    a. In the **Application Label** field, type the following:
    Oktaice### to OktaiceCentral
    b. In the **Base Url** field, type the following:
    https://oktacentral.oktapreview.com
    c. Click **Next**.
7. On the **Configure sign on settings** dialog box, perform the following:
    a. Leave the default **SAML 2.0** selection.
    b. In the **Default username** list, select **Custom**.
    c. In the new field, type the following:
    ${user.firstName}.${user.lastName}oktaice###@${f:substringAfter(user.email, "@")}
    Where ### is your Oktaice org number.
8. Click **Save and Assign**.

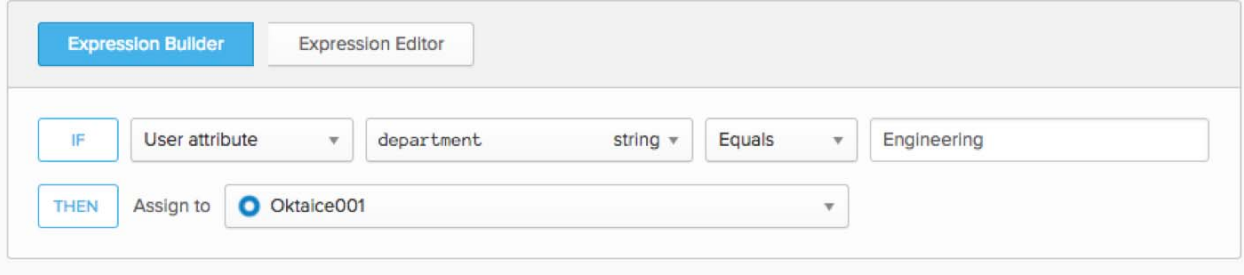
9. Click the **General** tab.
10. Next to **App Settings,** click **Edit.**
11. Select **Do not display application icon to users**.
12. Select **Do not display application icon in the Okta Mobile App**.
13. Click **Save**.
14. Click the **Provisioning** tab.
15. Click **Edit**.
16. Click **Enable provisioning features**.
17. In the **Security Token** field, paste the API token that you saved into Notepad.
18. Click **Test API Credentials.**
19. Next to **Create Users** and **Update User Attributes**, select **Enable**.
20. Click **Save**.
21. Click the **Sign On** tab.
22. Click **Edit**.
23. Click **View Setup Instructions**.
24. Follow the view setup instructions in oktacentral.oktapreview.com org – name your Identity Provider – oktaice###org2org
    a. Under the **Authentication Settings**, select **IdP Username** set to idpuser.subjectNameId and Leave the other default
    b. Under **JIT Settings**, perform the following:
        i. Leave the default **Profile master** not selected.
        ii. **Group Assignments**: None
    c. Under **SAML Protocol Settings**, perform the following:
        i. **IdP Issuer URI** - Copy from the setup instructions from the SAML app that you created – copy the **IdP Issuer URI**
        ii. **IdP Single Sign-on URL** - Copy from the setup instructions from the SAML app that you created - copy the **IdP Single Sign-On URL**
        iii. **IdP Signature Certificate** - download the X.509 Ceritificate from the SAML app that you created - upload it to the **IdP Signature Certificate**
    d. Click **Add Identity Provider**.

Transform Data

1. Return to Oktaice### preview org and navigate to **Directory**, **Profile Editor**, **Org2Org** application, and click **Mappings**
2. Click the **Okta to Oktaice### to OktaiceCentral** tab.
3. Locate and transform the user **email** attribute, as follows: **user.firstName +"." + user.lastName + "oktaiceXXX@" + substringAfter(user.email, "@")**
4. Click **Save Mappings**.
5. Click **Apply Updates Now**.

Configure Groups and Rules

1. Navigate to **Directory**, **Groups**.
2. Click **Add Group**.
3. Name: **Oktaice###**.
4. Click **Add Group**.
5. Click **Rules**.
6. Click **Add Rule**.
7. Create an **Engineering** rule as follows:



8. Click **Add Rule**.
9. Change the status from **Inactive** to **Active**.

Assign Groups and Push Groups

1. Navigate to **Applications**, **Applications**.
2. Click **OktaXXX to OktaIceCentral**.
3. Click the **Assignments** tab.
4. Click **Assign**.
5. Click **Assign to Groups**.
6. Find the **Oktaice###** group and click **Assign**
7. Click **Save and Go Back**.
8. Click **Done**.
9. Click the **Push Groups** tab.
10. **Push Groups**, **Find Groups by name**.

11. Find Oktaice###.
12. Click **Add Group**.

Verify Results

1. Open a new browser tab.
2. Login into oktacentral.oktapreview.com.
3. Navigate to **Directory**, **People**
4. Verify at least 1 account from your org was provisioned to the Central org. For example, Stephen Kim: Stephen.KimoktaiceXXX@oktaice.com.
5. Navigate to **Applications**, **Applications**.
6. Click **Microsoft Office 365**.
7. Click the **Push Groups** tab.
8. Verify that your Oktaice### group is active.

Create Groups and Rules

1. Navigate to **Directory**, **Groups**.
2. Click **Rules**.
3. Click **Add Rule**.
4. Name: Oktaice###
5. Change the dropdown from **User attribute** to **Group membership**.
6. **Enter a group** of Oktaice###.
7. Click **Select a group** and set it to Oktaice.
8. Click **Add Rule**.
9. Change the status from **Inactive** to **Active**.

Verify Results

1. Open a new browser tab.
2. Login to portal.office.com.
3. Username: **admin@okta###.onmicrosoft.com**
   Note: The instructor will give you the number
4. **Password**: Instructor Provided
5. Click **Admin**.
6. Click **Users**, **Active Users**.
7. Verify at least 1 account was provisioned to Office 365.
8. Click **Groups**, **Groups**.
9. Verify that your Oktaice### group was provisioned.
10. Select your Oktaice### group.
11. Verify user membership was provisioned to Office 365.

Lab 4-1: Encrypt a SAML Assertion

Objective	Encrypt SAML
Scenario	Okta Ice wants to integrate with Salesforce, they have decided that all SAML assertions will need to be encrypted
Duration	20-30 minutes

1. If not already logged in, open a new browser tab and navigate to the following website:
   https://test.salesforce.com
2. Sign in to Salesforce using the unique credentials provided from the instructor.
3. In Salesforce, perform the following:
   a. In the top bar, click **Setup**.
   b. In the left pane, under **Administer**, click **Security Controls**.
   c. Click **Certificate and Key Management**.
   d. Click **Create Self-Signed Certificate**.
   e. Label: oktaice###.
   f. Unique Name: oktaice###.
   g. Click **Save**
   h. Click **Download Certificate**.
   i. Click **Ok**.
   j. In the left pane, under **Administer**, click **Security Controls**.
   k. Click **Single Sign-on Settings**.
   l. Under **Federated Single Sign-On Using SAML**
   m. Click **Edit**.
   n. Select **SAML Enabled**.
   o. Click **Save**.
   p. Under **Single Sign-On Settings**, click **New**.
   q. In the top-right corner, click **Help for this Page**.
4. In the new web page, under **Configure SAML Settings for Single Sign-On**, click the **Provide information to your identity provider** link.
   a. Scroll down to the **Entity ID** field.
   b. Because we are not using custom domains with Salesforce, you must use the following default value for the Entity ID:
      https://saml.salesforce.com
   c. Close the help.salesforce.com web page.
5. On the Okta admin page, point to **Applications, Applications**.

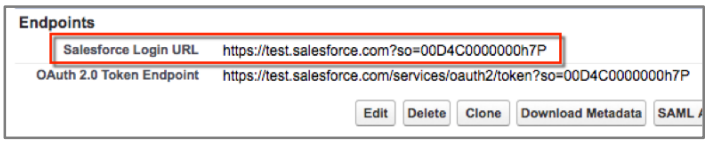
6. Click **Add Application**.
7. Under **Can't find an app?**, click **Create New App**.
8. In the **Create a New Application Integration** dialog box, perform the following:
    a. Leave the default **Web** selection for the **Platform**.
    b. Next to **Sign on method**, select **SAML 2.0**.
    c. Click **Create**.
9. On the **General Settings** page, perform the following:
    a. In the **App name** field, type: **AIW SAML**.
    b. Click **Next**.
10. On the **Configure SAML** page, perform the following:
    a. In the **Single sign on URL** field, type the following:
    https://placeholder.com
    b. In the **Audience URI (SP Entity ID)** field, type the Entity ID value of:
    https://saml.salesforce.com
    c. Click **Show Advanced Settings**.
    d. Change **Assertion Encryption to Encrypted**.
    e. In the **Encryption Certificate** field upload the certificate from Salesforce.
    f. Browse for and select a **oktaice###.crt** file.
    g. Click **Open**.
    h. Click **Preview SAML Assertion**.
    You should see CipherData and CipherValue.
    i. Close the browser tab that opened when you previewed the SAML Assertion.
    j. Click **Next**.
11. On the **Feedback** page, perform the following:
    a. Select **I'm an Okta customer adding an internal app**.
    b. In the expanded section, because we are using Salesforce as a test private application, select **This is an internal app that we have created**.
    c. Click **Finish**.
    The add application wizards completes and the Okta Administration app refreshes to the Sign On tab of the application.
    d. Under **Sign On Methods**, click **View Setup Instructions**.

    You are going to configure this application connector to Salesforce.
12. Return to the browser tab for Salesforce **SAML Single Sign-On Settings**.

13. In Salesforce, perform the following:
    a. Using the **View Setup Instructions** copy the following into the required fields for Salesforce:

Field	Value
Name	**AIW SAML**
API	**AIW_SAML**
Issuer	Paste the **Identity Provider Issuer** copied from the **View Setup Instructions** page
Identity Provider Certificate	From the View Setup Instructions page, click **Download Certificate** and then inside Salesforce, browse to and upload the new certificate.
Entity ID	https://saml.salesforce.com
Assertion Decryption Certificate	Oktaice###

    b. Click **Save**.
    c. Copy the **Salesforce Login URL** value.



14. In Okta, return to the definition of the **AIW SAML** app.
15. Click the **General** tab.
16. Under **SAML Settings**, click **Edit**.
    The SAML configuration wizard re-opens.
17. On the **General Settings** page, click **Next**.
18. On the **Configure SAML** page, perform the following:
    a. In the **Single sign on URL** field, overwrite the place holder value. Paste the **Salesforce Login URL** copied in a previous step.
    b. Click **Next**.
19. On the **Feedback** page, click **Finish**.
    Now you want to assign it to a person and test the configuration.
20. Click the **Assignments** tab.
21. Click **Assign**, then **Assign to People**.
22. Next to the **Class Admin** account, click **Assign**.
23. In the **Assign AIW SAML to People** dialog box, type your *Salesforce Admin username*.

24. Click **Save and Go Back**.
25. Click **Done**.

Launch SAML Tracer and Test

1. Access your Windows VM and launch Firefox.
2. Click the menu bar  in the top-right corner of Firefox.
3. Click **SAML Tracer**.
4. Log into your Oktaice Org with the oktatraining@okta.com credentials.
5. On the Okta Application homepage, click the **AIW SAML** app.
6. Verify Salesforce successfully logs in to the Salesforce Sandbox account.
7. Close the **Salesforce** browser tab.
8. In the SAML Tracer tool, find and click the **Post**.
9. Click the **SAML** tab.
10. Verify your assertion is encrypted.

Lab 4-2: Install the Okta RADIUS Agent

Objective	Install the Okta RADIUS Agent
Scenario	Okta Ice would like to add a RADIUS authentication option for primary authentication.
Duration	20-30 minutes

Configure the Okta Sign-on Policy

1. Login into your Oktaice Org.
2. In the Okta Administrator Application navigate to **Security**, **Policies**.
3. Click the **Okta Sign-on** tab.
4. Click **Add New Okta Sign-on Policy**.
5. Complete the fields as follows:
    a. **Policy Name**: Radius
    b. **Policy Description**: Radius
    c. **Assign to Groups**: Engineering
6. Click **Create Policy and Add Rule**.
7. Perform the following:
    a. **Rule Name**: Radius
    b. Change **And Authenticates via** to RADIUS.
    c. Click **Create Rule**.

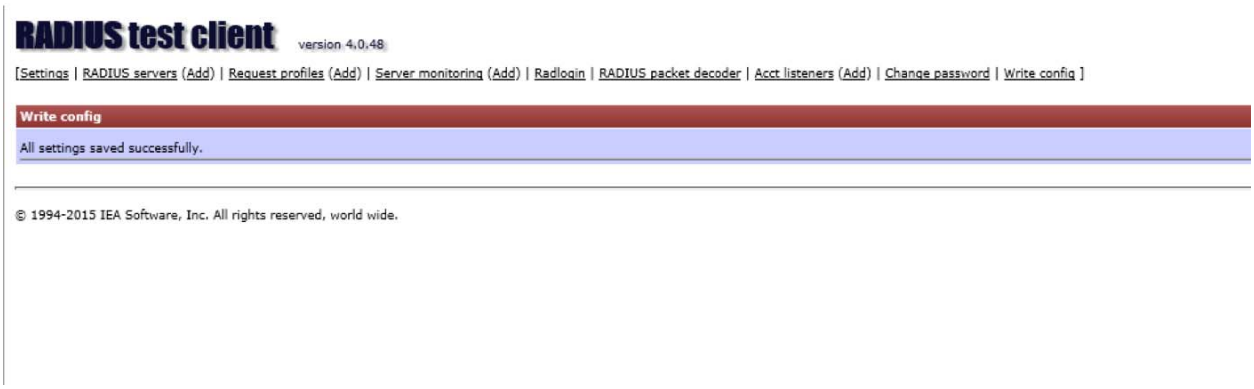
Install the Okta RADIUS Agent

1. Login to your VM.
2. Open a browser and login to your Oktaice org
3. In the Okta Administrator Application, point to **Settings**, **Downloads**.
4. Under **Admin Downloads** locate and download the Okta RADIUS Server agent.
5. Click **Save file**.
6. Navigate to the downloads folder.
7. Open the **OktaRadiusAgentSetup-2.3.1.exe** file.
8. Click **Run**.
9. Click **Next**, **Next**, **Next** and then click **Install**.
10. **RADIUS Share Secret**: icecream
11. Click **Next** and **Next**.
12. Click **Preview**.
13. Enter subdomain: oktaice###
14. Click **Next**.
15. In the web browser login with the **okta.admin@oktaice.com** credentials.

16. Click **Allow Access**.
17. Click **Finish**.

Test Using a RADIUS Client

1. Click the Windows logo.
2. Click search icon.
3. Search for Radius test client.
4. Click Radius test client to launch the application.
5. In the Windows Security window enter the following:
    a. User name: **Administrator**
    b. Password: **Tra!nme4321**
6. Click **OK**.
7. Next to the **RADIUS servers** link, click **Add** (RADIUS servers (Add)).
    a. Server address: **127.0.0.1**
    b. Shared secret: **icecream**
8. Click **Continue**.
9. Click Write config



10. Click **Radlogin**(| Radlogin |).
    a. RADIUS Server select: **127.0.0.1**
    b. Login: **jack.bailey**
    c. Password: **Tra!nme4321**
11. Click **Continue**.

You should see a Request and a Response

Request

Attribute		Data
Standard	Acct-Session-Id	"1491154304I12mkk"
Standard	NAS-IP-Address	127.0.0.1
Standard	NAS-Identifier	"Localhost"
Standard	NAS-Port	0
Standard	Calling-Station-Id	"1115551212"

Response

Status: Good

Resp Time: 797 ms

Attribute		Data
Standard	Reply-Message	"Welcome oktatraining@okta.com!"

Request

Attribute		Data
Standard	Acct-Session-Id	"1491154304I12mkk"
Standard	NAS-IP-Address	127.0.0.1
Standard	NAS-Identifier	"Localhost"
Standard	NAS-Port	0
Standard	Calling-Station-Id	"1115551212"

Response

Status: Good

Resp Time: 797 ms

Attribute		Data
Standard	Reply-Message	"Welcome oktatraining@okta.com!"

Lab 6-1: Configure the Marketing Structure for Contractors

Objective	Configure the social accounts and groups required by marketing for outsourcing the social engagement to contractors.
Scenario	Okta Ice has a 150% peak in social engagement every summer due to the ice cream sales seasonality. Okta Ice hired two companies to support their social accounts – Pinterest and LinkedIn – during the summer: • **Vanilla Social:** A marketing company located on the east coast that will cover the US and British market. • **Biscuit Publicité:** A marketing company located in French that will cover the European market. Okta Ice wants to onboard their marketing contractors using API as a master and grant access to their social accounts automatically. The reasons behind their choice are: • Each contractor may use a different HR system. Integrating with a third-party HR for each contractor takes time. • Each contractor agreed to share a list of users for provisioning. • Okta Ice wants a simple integration model in case they expand or hire new marketing contractors. In this lab, you will setup a basic structure – apps, group, and rule – to receive the marketing contractors. **Note:** You are executing this lab to understand the importance of scoping before receiving data from multiple masters.
Duration	15 minutes

Register LinkedIn and Pinterest Apps

1. Log into your Oktaice### org, access your Okta org as **okta.admin**.
2. Click **Admin**.
3. Click **Applications**.
4. Click **Add Application**.
5. Search for and add the **LinkedIn** app.
6. Click **Next**.
7. Complete the fields as follows and then click **Save and Assign**.

Attribute	Value
Who sets the credentials?	Users share a single username and password set by administrator.
Shared Username	oktatraining@okta.com
Shared Password	*provided by the instructor

8. Repeat the previous steps to register the **Pinterest** application with the following sign on settings:

Attribute	Value
Who sets the credentials?	Users share a single username and password set by administrator.
Shared Username	oktatraining@okta.com
Shared Password	*provided by the instructor

Create the Marketing Contractors Group

Note: The marketing contractors group will provide access to social networks for all contractors. Concentrating the assignment to a group helps with assigning and removing social apps for all contractors.

1. Point to **Directory**, **Groups**.
2. Create the following group:

Name	Group Description
Marketing Contractors	For contractors working on marketing projects.

3. Click **Marketing Contractors**.
4. Click **Manage Apps**.
5. Assign **LinkedIn** and **Pinterest**, and then click **Done**.

Create the Marketing Contractors Rule

Note: The marketing contractors rule guarantees that contractors (users with the costCenter starting with `mkt` and with the `organization` fulfilled with their external org will have immediate access to the Marketing Contractor group.

1. Click **Directory**, **Groups**.
2. Click the **Rules** tab and then click **Add Rule**.
3. In the **Name** field, type the following:
   Marketing Contractors
4. Click **Expression Editor** and provide the following Rule:
   Tip: To learn more about the expression used in this rule, check the Conditional Expressions under the Okta Expression Language doc (http://developer.okta.com/reference/okta_expression_language.)

Attribute	Value
IF	`user.organization != NULL AND String.startsWith(user.costCenter,"mkt")`
THEN	Assign to `Marketing Contractors`

5. Click **Add Rule**.
6. Change the Rule status to **Active**.

Lab 6-2: Configure API as a Master

Objective	Configure Okta and Postman for API as a Master.
Scenario	In this lab, you: • Create and obtain an API Token. This token is used by the master to perform user management tasks via Okta API. • Configure Postman to act as API Master. **Notes:** For this lab, you use Postman as master. Although Postman is not a typical API master system, Postman: • Provides detailed insights about the REST API calls executed in the API as a Master integration. • Can be used on the field to help with the API as a Master development. • Can reduce the development time in tasks such as defining the JSON body for API requests, testing the API endpoints, and troubleshooting API as a Master issues.
Duration	15 minutes

Get an API Token

1. In the Windows VM, log into the Oktaice### org.
2. In the Okta Admin interface, click **Security**, **API**.
3. Click the **Token** tab.
4. Click **Create Token**.
5. In the **Name** field, type the following and then click **Create Token**:
   Postman as API Master
6. Record the token value retrieved by Okta in Notepad.
   Important: This is the only time you have access to the token value. In case you lose the token value, re-create the token.
7. Click **OK, Got it**.

Configure Postman

1. Open Postman.
2. Observe that Postman displays few collections on the left pane and some environments in the combo box on the top right corner of the window.

   Tip: These collections and environments are samples that you can use to make Okta API requests. These samples are available in the Get Started With the Okta APIs doc (http://developer.okta.com/docs/api/getting_started/api_test_client.html.)
3. In the top-right corner, click **Settings (gear icon)** > **Manage Environments**.
4. Click **example.oktapreview.com**.
5. Update the environment variables as follows and then click **Update**.

Attribute	Value
url	Your Okta ICE org url. For example, `https://oktaice`**`XXX`**`.oktapreview.com`
apikey	The **API Token** retrieved by Okta in the previous task. For example, `fzyV0kSHGAQ8k2h23e-y2I8Y`

6. Close the **Manage Environments** pop up.

Explore API Requests

1. Under Collections, click **API as a Master > 1 Create** > **Create Users**.

   Tip: The Create Users request in this Lab is based on the " *Create Activated User with Password & Recovery Question*" sample in the Okta API doc.

2. Click **Header** and confirm that the **{{apikey}}** is sent in the **Authorization** header.

   Tip: {{apikey}} is a dynamic variable. Postman replaces dynamic variable with values from your environment – or from input files – every time you send REST requests.

3. Click **Body** and check the body contents.

```
{
  "profile": {
    "firstName": "{{FIRSTNAME}}",
    "lastName": "{{LASTNAME}}",
    "email": "{{MAIL}}",
    "login": "{{MAIL}}",
    "organization": "{{ORGANIZATION}}",
    "costCenter": "{{COSTCENTER}}"
  },
  "credentials": {
    "password": { "value": "{{PASSWORD}}" },
    "recovery_question": {
      "question": "{{QUESTION}}",
      "answer": "{{ANSWER}}"
    }
  }
}
```

   Tip: The Create User JSON body has dynamic variables for the user data (in red). This data will be replaced with the user information during the tests.

4. **Optionally,** explore the remaining requests under the **API as a Master** collection.

   Tip: Each request will present differences in the request method (GET, POST, PUT…), URL, JSON body, and parameters.

Lab 6-3: Onboard Users with API as a Master

Objective	Onboard marketing contractors with API as a Master.
Scenario	In the previous labs, you prepared the Okta Ice org with apps, groups, rules, and an API token to onboard marketing contractors from Postman. Now it's time to use API as a master to onboard contractors from Vanilla Social and Biscuit Publicité.
Duration	10 minutes

Onboard Users

1. In Postman, click **Runner**.
   The Collection Runner Window appears.
2. Under current run, select the **API as Master > 1 Create** collection and then update the attributes as follows:

Attribute	Value
Environment	`example.oktapreview.com`
Iteration	`10`
Data	Select and open: `C:\labs\api-master\create.csv`
Data File Type	Text/csv

3. **Optionally**, click **Preview** to check the users that will be created in Okta through API as a Master.
4. Click **Start Run**.
5. Wait until the test is completed.
   You should have 10 users created at Okta.
6. Close the Runner window.

Verify the Results

1. Return to the Okta Admin app.
2. Navigate to **Directory** > **Groups** and then open the **Marketing Contractors** group.
3. Verify that the group contains 10 active members – five from Vanilla Social and five from Biscuit Publicité.
   This confirms that the Marketing Contractors rule is working.
4. Sign out as **okta.admin**.

Access an App as a Contractor

1. Sign into Okta as the **callie.nelson@vanillasocial.com** marketing contractor account.
2. Ask the Instructor for the password.
3. Confirm that **LinkedIn** and **Pinterest** are available for the marketing user.
4. Launch one of the apps to access the social account.
5. Sign out of Okta.

Lab 6-4: Update Users with API as a Master

Objective	In this lab, you update marketing contractors with API as a Master.
Scenario	Okta Ice decided that, for information purposes, Vanilla Social and Biscuit Publicité must send the title for each user accessing their social accounts. So, you upload the user profiles using API as a Master.
Duration	5 minutes

Update Users

1. Return to Postman Runner.
2. Under current run, select the **API as Master > 2 Update** collection and then update the attributes as follows:

Attribute	Value
Environment	`example.oktapreview.com`
Iteration	`10`
Data	Select and open: `C:\labs\api-master\update.csv`
Data File Type	Text/csv

3. Click **Start Run**.
4. Wait until the test is completed.
   You should have 10 users updated with new titles at Okta.
   Important: If Postman returns the error *"socket hang up"* or *"An error occurred while running this request. Open DevTools for more info"*, restart the Runner and repeat the update.
5. Close the Runner window.

Verify the Results

1. Return to your Okta org as **okta.admin**.
2. Click **Admin**.
3. Navigate to **Directory** > **People**.
4. Search and open **Zachary Bryan**.
5. Click **Profile** and confirm that Zachary's title is Marketing Account Director.

Lab 6-5: Deactivate and Delete Users with API as a Master

Objective	Deactivate and delete marketing contractors with API as a Master.
Scenario	After the summer, Okta Ice decided to end their contract with Vanilla Social and Biscuit Publicité. Now, it's time to deactivate the contractors using API as a Master.
Duration	5 minutes

Deactivate Users

1. Return to Postman Runner.
2. Under current run, select the **API as Master > 3 Deactivate** collection and then update the attributes as follows:

Attribute	Value
Environment	`example.oktapreview.com`
Iteration	`10`
Data	Select and open: `C:\labs\api-master\deactivate.csv`
Data File Type	Text/csv

3. Click **Start Run**.
4. Wait until the test is completed.
   You should have 10 users deactivated at Okta.

Verify the Results

1. Return to your Okta org and navigate to **Directory** > **People**.
2. In the left menu, select **Deactivated**.
3. Confirm that you can see users from Biscuit Publicité and Vanilla Social.

Optional: Delete Users

Important: Deleting a user is an action that cannot be recovered.

1. Return to Postman Runner.
2. Under current run, select the **API as Master > 4 Delete** collection and then update the attributes as follows:

Attribute	Value
Environment	`example.oktapreview.com`
Iteration	`10`
Data	Select and open: `C:\labs\apimaster\deactivate.csv`
Data File Type	Text/csv

3. Click **Start Run**.
4. Wait until the test is completed.
   You should have 10 users deleted at Okta.
5. Return to Okta as **okta.admin** and confirm that you cannot see the contractor users under **Directory** > **People**.

Lab 7-1: Configure the Okta Sign-in Widget

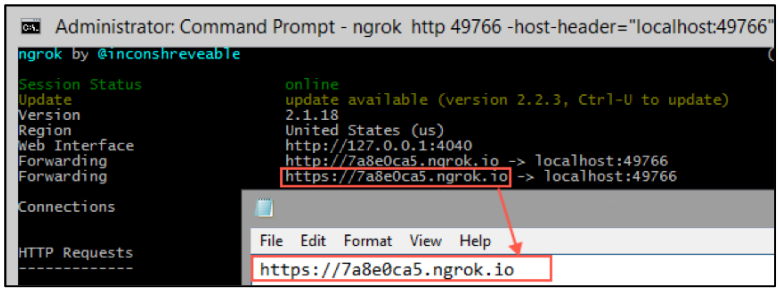
Objective	Configure a custom sign-in experience
Scenario	Okta Ice would like allow vendors to login with a custom sign-in experience
Duration	30-40 minutes

Configure the Login Widget

1. Log in to your VM.
2. Open a new browser tab.
3. In the address bar, type the following:
   developer.okta.com
4. In the top bar, click **Code**.
5. Click the **JavaScript** icon.
6. Click **Okta Sign-In Widget**.
7. Scroll down to **Creating an HTML file with the widget code**.
8. Copy and paste the HTML into Atom and save the file with the following name: login.html
9. After saving the file, replace the following 2 strings with your oktaice###.oktapreview.com:
   a. var baseUrl = 'https://example.okta.com'; should now be https://oktaice###.oktapreview.com
   b. var redirectURL = 'https://localhost.8000/signed-in.html' should now be https://oktaice###.oktapreview.com/app/UserHome
10. Save the changes.
11. Copy your file into the C:\inetpub\wwwroot folder.

Expose the Local Host to the Internet

1. Launch ngrok:
   a. Launch the command prompt.
   b. To launch ngrok, enter the following command:
      ngrok http 80
   c. Record the forwarding URL retrieved by ngrok in Notepad.
      For example, https://7a8e0ca5.ngrok.io.



Enable CORS in Your Okta Org and Test

1. If you are not already signed into Okta, open a new browser tab and log into your Okta org.
2. If necessary, navigate to the **Admin** interface.
3. Point to **Security** and click **API**.
4. Click the **Trusted Origins** tab.
5. Click **Add Origin**.
6. Populate the fields as follows:

   Name: Sign In Widget
   Origin URL: type in the forwarding URL from above.
   E.g. **https://f0a29874.ngrok.io**
   CORS: checked
   Redirect: cleared
7. Click **Save**.
8. Sign out of Okta.
9. In a web browser go to your forwarding URL plus the login.html for example: **http://f02a29874.ngrok.io/login.html**
10. Sign in using your Okta credentials.

Perform Additional Challenges with the Login Widget

1. Change the background image.
2. Change the logo image.

Lab 8-1: Configure IDP Discovery

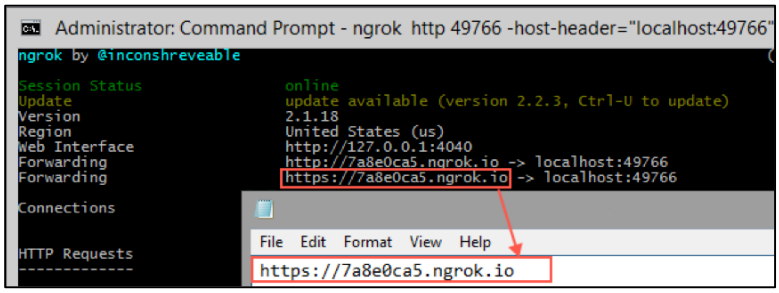
Objective	Configure **oktacentral**, your **oktaice** org, and the **code sample** for the IDP discovery.
Duration	15 minutes

Note: This lab should be executed from your Windows Virtual Machine.

Open and Launch the Code Sample

1. In your Windows VM, launch **Visual Studio 2013**.
2. Click **File** > **Open** > **Project/Solution**.
3. Open the file **C:\labs\idp-discovery\code\CustomLogin_wHomeRealm.sln**.
   The CustomLogin_vHomeRealm files are displayed in the Solution Explorer pane, located in the right hand.
4. To launch the application, click **Debub** > **Start Debugging**.
   Visual Studio launches a browser with the CustomLogin_vHomeRealm app.

5. Launch ngrok:
   a. Launch the command prompt.
   b. To launch ngrok, enter the following command:
      ngrok http 49766 –host-header="localhost:49766"
   c. Record the forwarding URL retrieved by ngrok in Notepad.
      For example, https://7a8e0ca5.ngrok.io.

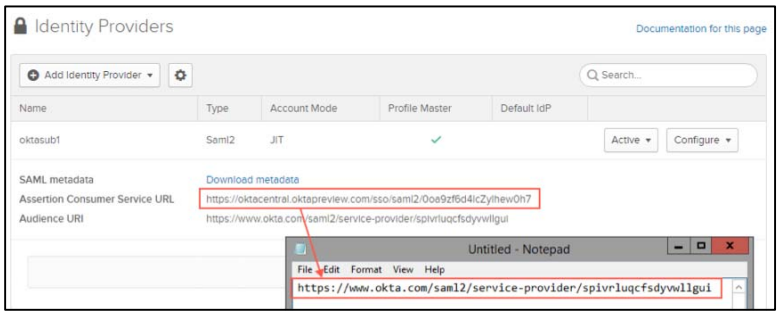


   d. Launch a browser and access CustomLogin_vHomeRealm app with the ngrok url.
      For example, https://7a8e0ca5.ngrok.io/Home.

The CustomLogin_vHomeRealm app should be displayed.

Configure the Okta Central Org

1. Launch a browser and access the **oktacentral** org as admin.
2. Click **Admin**.
3. Record the Assertion Consumer Service URL:
   a. Click **Security**, **Identity Providers**.
   b. Locate the Identity Provider connection to your org (oktaiceXXXorg2org), and then copy the **Assertion Consumer Service URL** to Notepad.



4. Obtain an API token:
   a. Click **Security**, **API**.
   b. Click **Create Token**.
   c. In the Name field, type the following and then click **Create Token**: CustomLogin_iceXXX
   d. Copy the token value to Notepad.
   e. Click **OK, got it**.

5. Add CustomLogin_vHomeRealm as a Trusted Origin:
    a. Click the **Trusted Origins** tab.
    b. Click **Add Origin**.
    c. Enter the information as follows and click **Save**.

Attribute	Value
Name	CustomLogin_iceXXX
Origin URL	Your ngrok url. For example: https://7a8e0ca5.ngrok.io
Type	CORS: selected Redirect: cleared

6. Sign out of **oktacentral**.

Configure your Okta Ice Org

1. Access your **oktaice** org as **okta.admin**.
2. Click **Admin**.
3. Click **Security**, **API**.
4. Add CustomLogin_vHomeRealm as a Trusted Origin:
    a. Click the **Trusted Origins** tab.
    b. Click **Add Origin**.
    c. Enter the information as follows and click **Save**.

Attribute	Value
Name	CustomLogin_iceXXX
Origin URL	Your ngrok url. For example: https://7a8e0ca5.ngrok.io
Type	CORS: selected Redirect: cleared

5. Add oktacentral as Trusted Origin:
    a. Click **Add Origin**.
    b. Enter the information as follows and click **Save**.

Attribute	Value
Name	Oktacentral
Origin URL	https://oktacentral.oktapreview.com
Type	CORS: cleared Redirect: selected

6. Sign out of oktaice org.

Configure the Code Sample

1. Return to Visual Studio.
2. In the right pane, open the **CustomLogin_wHomeRealm** > **App_Data** > **IdentityProviders.json** file.

   Tip: The IdentityProviders.json file identifies in what Okta org a user will authenticate based on the e-mail domain (idpDomain) extracted form his/her username.
3. Update the code as follows:

```
[
  {
    "idpName": "OktaHub",
    "idpDomain": "okta.com",
    "idpUrl": "https://oktacentral.oktapreview.com",
    "idpACS": "none"
  },{
    "idpName": "oktaiceXXX",
    "idpDomain": "oktaice.com",
    "idpUrl": "https://oktaiceXXX.oktapreview.com",
    "idpACS": "https://oktacentral.oktapreview.com/sso/saml2/dsW33ed"
  },{
    "idpName": "oktaiceXXX",
    "idpDomain": "oktaice.local",
    "idpUrl": "https://oktaiceXXX.oktapreview.com",
    "idpACS": "https://oktacentral.oktapreview.com/sso/saml2/dsW33ed"

  }
]
```

4. Update the items in red as follows:

Attribute	Value
`idpName`	Your Identity Provider connection name registered in oktacentral. For example, `oktaice001`.
`idpUrl`	Your okta ice url.
`idpACS`	Your Identity Provider **Assertion Consumer Service URL** obtained in oktacentral during lab 8-1.

5. Save and close the **IdentityProviders.json** file.
6. Open the **CustomLogin_wHomeRealm** > **Web.config** file.
7. Update the okta.ApiUrl and okta.ApiToken according to the table:

```
<add key="okta.ApiUrl" value="https://oktacentral.oktapreview.com" />
<add key="okta.ApiToken" value="tokenvalue" />
```

Note: Use the API **token value** obtained in oktacentral during Lab 8-1. For example: `oAkd2io3e0e2d0o3e`

8. Save and close the **Web.config** file. If you are prompted to stop the debugger select Yes.
9. **Debug** > **Start Debugging**.
   This restarts the CustomLogin application for the final tests.

Lab 8-2: Test the IDP Discovery

Objective	In this lab, you test the IDP discovery.
Duration	15 minutes

Test the Access as an Okta Central User

1. To make sure that you are not logged in any Okta org, restart your browser.
2. Navigate to the CustomLogin application using your ngrok url – obtained in Lab 8-1.
   For example: https://7a8e0ca5.ngrok.io/Home
   The Home page is displayed.
3. Enter oktatraining@okta.com as Username and click Submit.
4. Right-click the Sign In page, and then click View page source.
5. In line **67**, confirm that the oktaorg value is **oktacentral.oktapreview.com**.
   This confirms that the CustomLogin app sets oktacentral as the org for authentication when the user login mail domain is @okta.com.
6. Close the source code.
7. Sign in as oktatraining@okta.com.
8. The **oktacentral** home page for oktatraining user is displayed.
9. Access oktaiceXXX.oktapreview.com.
   A login page appears because the oktatraining user is logged only in oktacentral.
10. Return to oktacentral.oktapreview.com.
11. Sign out of oktacentral.

Test the Access as an Okta Ice User with the @oktaice.local Domain

1. Navigate to the CustomLogin application using your ngrok url.
   For example: https://7a8e0ca5.ngrok.io/Home
2. Enter sarah.james@oktaice.local as **Username** and click **Submit**.
3. Right-click the **Sign In** page, and then click **View page source**.
4. In line **67**, confirm that the oktaorg value is **oktaiceXXX.oktapreview.com**.
   This confirms that the CustomLogin app sets oktaiceXXX as the org for authentication when the user login mail domain is `@oktaiceXXX.com`.
5. In line **68**, confirm that the intRelayState value matches the **Assertion Consumer Service URL** obtained in oktacentral during lab 8-1.
   This happens because the user will be redirected to oktacentral right after log into oktaiceXXX in a SAML SP initiated sign-on.
6. Close the source code.
7. Sign in as gerald.miles@oktaice.local.

8. The **oktacentral** home page for Gerald is displayed.
9. Access oktaiceXXX.oktapreview.com. The okta ice home page for Sarah is displayed. This happens because Gerald is logged in both oktaiceXXX – logged in via widget – and oktacentral – logged in via SAML SP initiated sign on.

Optional: Test Access as an Okta Ice User with @oktaice.com Domain

1. Access your oktaice org as administrator and assign the OktaiceXXX to OktaCentral app **to okta.admin**.
2. Sign out of your oktaice org and repeat steps from the previous section to login as okta.admin@oktaice.com.
   Okta admin should login with the same steps as gerald.miles@oktaice.local.

Stop the Code Sample

1. Close your browser.
2. In Visual Studio, click **DEBUG** > **Stop Debugging**.
3. Close Visual Studio.
4. Close the command prompt terminal where ngrok is running.

Lab 9-1: Launch and Test the SCIM Server

Objective	Start the SCIM Server and test with Runscope. Runscope is a useful tool for both running scripted tests and capturing and analyzing live HTTP traffic.
Duration	20 minutes

Deploy the SCIM Server Code

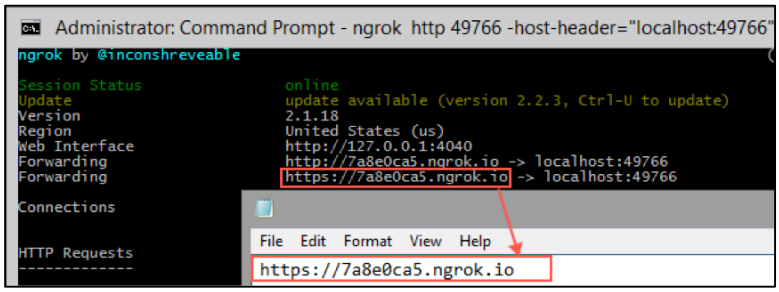
1. If not already started, launch the remote VM in ReadyTech.
    a. Open your browser.
    b. Navigate to http://okta.instructorled.training/.
    c. Enter your unique assigned access code.
    d. Enter the following credentials to login to Windows:
    User: OKTAICE\Administrator
    Password: Shared Password
2. Inside the VM, create an account at Runscope.
    Running the tests with Runscope is optional. The steps that involve Runscope can be skipped, but the tool is extremely useful, very informative and free for how we will be using it.
    a. Inside the VM, launch **Chrome**.
    b. Navigate to http://www.runscope.com/.
    c. If you already have an account, select **Sign In**. If not, select **Sign Up** and complete the registration process.
3. Copy the SCIM Server source code.
    a. Launch **Windows Explorer**.
    b. Navigate to **Local Disk (C:) – labs\native-scim\python\TCBC2**.
    c. Right click on **scim-server.py** and select **Copy.**
    d. Navigate to **Local Disk (C:) – labs\native-scim\python**.
    e. Right click in the directory and select **Paste**.

Test the SCIM Server with Runscope

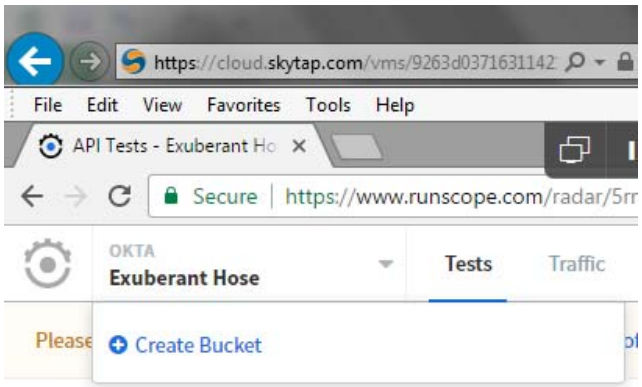
1. Start the SCIM server.
    a. In the VM, from the Windows task bar, start a **Command Prompt**.
    b. In the terminal window, enter the command "**cd c:\labs\native-scim\python**" and press Enter.
    c. To start a virtual environment for Python, enter the command "**venv\scripts\activate.bat**" and press Enter.

   d. Start the Python SCIM server by typing the command "**python scim-server.py**" and then Enter.

2. Start NGROK to enable tunneling.
   a. In the VM, start a new command prompt window by right clicking the icon on the task bar and selecting **Command Prompt**.
   b. Enter the command "**.\ngrok http 5000**".
   c. In the terminal window, locate the Forwarding link for HTTPS. E.g. https://abcd1234.ngrok.io. Write down the uniquely generated URI for your NGROK server.



3. Load the pre-built SCIM tests into Runscope.
   a. Inside the VM, log into Runscope with Chrome.
   b. In the upper left corner, click the down arrow in the **BUCKET** area. Select **Create Bucket**.



   c. Name the bucket **SCIM**.
   d. In **Create from File** section, select the **Import Test button**.
   e. In the list of available test format types, select **Runscope API Tests**.
   f. Below, click **Choose File**.
   g. In the file browsing window, navigate to **C:\ – labs\native-scim\python**.
   h. Select **Okta SCIM 2.0 Tests.json** and click **Open**.
   i. Click **Import API Test**.
   j. After the import finishes, click **Close**.

Verify Tests on Runscope

1. In the central top area, click **Tests**.
2. In the **API Tests** area, click **Okta SCIM 2.0 Tests**.
3. Click **Edit Test**.
4. Expand **Test Settings** by clicking the **arrow**.
5. In **Initial Variables**, click **Add Initial Variable**.
6. For the **Variable Name**, enter **SCIM Base URL**.
7. For the **Value**, enter in the **NGROK URI** written down earlier, appending **/scim/v2** to the end.
   For example: https://abcd1234.ngrok.io/scim/v2.
8. In **Initial Variables**, click **Add Initial Variable**.
9. For the **Variable Name**, enter **AuthToken**.
10. For the Value, enter **goodtoken**.
11. Click **Save**.
12. Click the **Run Now** link.
13. On the left side, in the Recent Test Results area, click the results.
    The tests should have succeeded, but certain elements may fail due to performance.
    Examine the calls, including the requests and responses.
14. Click **View Results**.
15. Leave the Runscope tab open in the browser.
16. Keep all NGROK, Runscope, and your SCIM sessions open.

Lab 9-2: Define a Native SCIM Application in Okta

Objective	Define an application connector in Okta, and verify the API credentials from there. Okta verifies the availability of the SCIM Server by requesting a list of User and Group Resources. In SCIM, resources are returned in a ListResponse.
Duration	30 minutes

Test the API Connection from Okta

1. Open a new tab in your browser.
2. Log into Okta with your Okta admin credentials.
3. Click **Admin**.
4. Point to **Applications, Applications**.
5. Click **Add Application**.
6. In the search window, type in "**scim**".
7. Next to **SCIM 2.0 Test App (Header Auth)**, click **Add**.
8. In the **Enter basic info** window, for the **Application label** field, update the value to be **SCIM 2.0 Training App**.
9. Click **Next**.
10. In the **Configure sign on settings** window, leave the default values and click **Save and Assign**.
11. Click the **Provisioning** tab.
12. Click **Edit**.
13. Select **Enable provisioning features**.
14. Under API Credentials area, complete the fields as follows:
    a. **Base URL**: enter in the NGROK URI written down earlier, appending /scim/v2 to the end.
       For example: https://abcd1234.ngrok.io/scim/v2
    b. **API Token**: goodtoken
15. Click **Test API Credentials**.
    This should return a success message.
16. Change the value for **API Token** to **badtoken**.
17. Click **Test API Credentials** again and verify that the call fails because it is unauthorized.
18. Change the value of the **API Token** back to **goodtoken**.
19. Click **Save.**

Enable Provisioning in Okta and Capture Traffic in Runscope

1. Enable Runscope to Capture Traffic.
    a. Return to the browser tab for **Runscope**.
    b. In the top-middle menu, click **Traffic**.
    If needed, skip the tutorial.
    c. Under **Try It Now!**, type in your URL for your SCIM server.
    For example: https://14a16910.ngrok.io/scim/v2
    d. In the **Sample Code** area, click **Other**.
    e. Copy the generated URL.
    For example: https://14a16910-ngrok-io-xb961j6ibzha.runscope.net/scim/v2.
2. Enable User Provisioning in Okta.
    a. Return to the Okta Admin app.
    b. Click **Applications**, **Applications**.
    c. Click the **SCIM 2.0 Training App** link.
    d. In the list of tabs, click **Provisioning**.
    e. Click **Edit**.
    f. In the **Base Url** field, paste in the Runscope-generated URL.
    g. Next to **Create Users**, select **Enable**.
    h. Next to **Update User Attributes**, select **Enable**.
    i. Scroll down and click **Save**.
3. Test Provisioning from Okta.
    a. Click **Directory**, **People**.
    b. Click **Add Person**.
    c. Populate the following fields:
        i. **First name**: Scim
        ii. **Last name**: Test
        iii. **Username**: scimuser@test.com
        iv. **Primary email**: scimtest###@mailinator.com
    d. Click **Add Person**.
    e. In the list of people, click the new user **Scim Test**.
    f. In the **Applications** tab, click **Assign Applications**.
    g. Next to your SCIM app definition, **SCIM 2.0 Training App**, click **Assign**.
    h. Click **Save and Go Back** and then click **Done**.
    i. Back in Runscope, click the **Traffic** link at the top of the page.
    j. Inspect the HTTP GET and the HTTP POST sent as a part of provisioning the new user in the endpoint application. Be sure to look at the request and response.

Lab 9-3: Extend Native SCIM with Custom Attributes

Objective	Define a new attribute associated with the SCIM application connector and test with Okta. Verify the message with Runscope. (OPTIONAL) Review the code in a SCIM server that will implement the queries for both a list of users and a specific user, and insert a user into the endpoint database.
Duration	20 minutes

Setup Custom Attributes in Okta

1. Create a new attribute on the User profile in Okta.
    a. Return to the Okta Admin app.
    b. Point to **Directory** and click **Profile Editor**.
    c. Next to **Okta**, click **Profile**.
    d. In the Profile Editor page, click **Add Attribute**.
    e. In the Add Attribute page, enter in the following field values:
        Display name: Tenant ID
        Variable name: appTenantId
        Description: Unique organization identifier.
        Data type: string
        Attribute Length: Equals, 6
        Attribute required: unchecked
    f. Click **Add Attribute**.
2. Create a new field for the SCIM application connector in Okta.
    a. Point to **Directory** and click **Profile Editor**.
    b. To the right of the **SCIM 2.0 Training App User**, click **Profile**.
    c. In the Profile Editor page, click **Add Attribute**.
    d. In the Add Attribute page, enter in the following field values:
        Display name: Multi-tenant ID
        Variable name: tenantId
        External name: tenantId
        External namespace: TrainingApp
        Description: Identifier for an organization.
        Data type: String
        Attribute Length: Equals, 6
        Attribute required: unchecked
        Scope: unchecked
    e. Click **Add Attribute**.
3. Map the local field to the endpoint, in Okta.

   a. Click **Map Attributes**.
   b. In the SCIM 2.0 Training App User Profile Mappings page, click **Okta to SCIM 2.0 Training App**.
   c. Locate the row for mapping to **tenantId**.
   d. On the left side for the associated Okta User attribute, set the value to **String.toUpperCase(user.appTenantId)**.
   e. The arrows indicate when the mapping is performed. Confirm the mapping is set to **Apply mapping on user create and update**, which is associated with the green arrow.
   f. Click **Save Mappings**.
   g. In the lower **Mappings saved!** area, click **Don't apply updates**.

Test the Custom Attribute Mapping

1. Modify the user to test the mapping.
   a. Point to **Directory** and click **People**.
   b. Click the user **Scim Test**.
   c. Click the **Profile** tab.
   d. Click **Edit**.
   e. In the field **Tenant ID**, type the following:
      abcdef
   f. Click **Save**.
2. Verify custom field in message in Runscope.
   a. Back in Runscope, click the **Traffic** link at the top of the page.
   b. Expand the latest HTTP **PUT** message, which shows the Response.
   c. Click **Request**.
   d. Verify the message body contains a JSON object "TrainingApp", with a key "tenantId" and that the value is "**ABCDEF**".
4. Sign out of Runscope.
5. Close the command prompt windows where python and ngrok are running.

(OPTIONAL) Review the SCIM Server Code

1. Open the SCIM Server source code in a text editor.
    a. In the VM, launch **Windows Explorer**.
    b. Navigate to **Local Disk (C:) – labs\native-scim\python**.
    c. Right click on **scim-server.py** and select **Open.**
2. In Atom, review the existing code. Note: in Atom, you can jump to a specific line using Ctrl + g.
    a. Around line 20, review the imported modules:
        - Flask is a web application framework, which will be used to create the HTTP endpoints for SCIM.
        - SQLAlchemy is an object-relational mapping framework, which will be used to read and write data to the SQLite database.
    b. Around line 30, Review the inline code which will execute when the python script is run. These instructions will launch the web server and connect to the local database.

```
app = Flask(__name__)
database_url = os.getenv('DATABASE_URL', 'sqlite:///test-users.db')
app.config['SQLALCHEMY_DATABASE_URI'] = database_url
db = SQLAlchemy(app)
socketio = SocketIO(app)
```

3. Review the class that represents **a single user resource**.
    a. Around line 40, locate the class called **User**, which extends db.Model. This class represents a User resource in SCIM, but also the User table in the database. The member variables not only represent the key values that will be passed from Okta to the provisioning server, but also the values returned.

```
class User(db.Model):
    __tablename__ = 'users'
    id = db.Column(db.String(36), primary_key=True)
    active = db.Column(db.Boolean, default=False)
    userName = db.Column(db.String(250),
                         unique=True,
                         nullable=False,
                         index=True)
    familyName = db.Column(db.String(250))
    givenName = db.Column(db.String(250))

    def __init__(self, resource):
        self.update(resource)
```

    b. Around line 60, locate the **to_scim_resource** function.
    This function serializes the User object into a SCIM-compliant JSON object. The rv variable (as in "return value") will hold a Python dictionary

object containing the SCIM representation of the User resource.

```
def to_scim_resource(self):
    rv = {
        "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User"],
        "id": self.id,
        "userName": self.userName,
        "name": {
            "familyName": self.familyName,
            "givenName": self.givenName
        },
        "active": self.active,
        "meta": {
            "resourceType": "User",
            "location": url_for('user_get',
                                user_id=self.id,
                                _external=True)
        }
    }
    return rv
```

4. Review the class that represents **a list of user resources**.
    a. Around line 80, locate the **ListResponse** class.
    b. The **__init__** function is the constructor, and will assign pagination values to local variables depending on the current page.

```
class ListResponse():
    def __init__(self, list, start_index=1, count=None, total_results=0):
        self.list = list
        self.start_index = start_index
        self.count = count
        self.total_results = total_results
```

    c. Locate the **to_scim_resource** function, which will return the list of results as a dictionary object, ready to be converted to the JSON representation in SCIM. The startIndex, itemsPerPage, and totalResults variables are used for pagination. The variable "resources" will hold the list of users that will be queried from the application's User table called

"resources".

```
def to_scim_resource(self):
    rv = {
        "schemas": ["urn:ietf:params:scim:api:messages:2.0:ListResponse"],
        "totalResults": self.total_results,
        "startIndex": self.start_index,
        "Resources": []
    }

    resources = []
    for item in self.list:
        resources.append(item.to_scim_resource()

    if self.count:
        rv['itemsPerPage'] = self.count

    rv['Resources'] = resources
    return rv
```

5. Review the **authorization** logic.
    a. Around line 103, locate the function **is_authorized**.
    This is a helper method to validate the token passed from Okta in the HTTP request header. If the local_token and remote_token (the one passed in the request) match, return True, if not, return False. In this lab, the token is stored unencrypted in the source code. In your SCIM server, you should store the token value in an encrypted data store.

```
def is_authorized(request_headers):
    local_token = 'goodtoken'
    request_token = str(request_headers.get('Authorization'))
    if local_token == request_token:
        return True
    else:
        return False
```

6. Review the function which supports **queries for Group resources**.
    a. Around line 110, locate the **groups_get** function, which will return an empty ListResponse.
    b. This is necessary in the SCIM server so that Okta may validate the connection, even though it not currently implemented by Okta.

```
@app.route("/scim/v2/Groups", methods=['GET'])
def groups_get():
    rv = ListResponse([])
    return flask.jsonify(rv.to_scim_resource())
```

7. Review the function which supports **queries for multiple users**, based on filters.
    a. Around line 117, locate the definition of the function called **users_get**.
    This function is called when Okta needs to perform a filtered query. This function uses regular expressions to parse the filter parameter, and

performs a query on the database, and contains pagination logic. This function uses the ListResponse class to generate the JSON response.

```
# Completed HTTP GET with filtered lookup.
@app.route("/scim/v2/Users", methods=['GET'])
def users_get():
    if not(is_authorized(request.headers)):
        return scim_error('Unauthorized', 401)
    query = User.query
    request_filter = request.args.get('filter')
    match = None
    if request_filter:
        match = re.match('(\w+) eq "([^"]*)"', request_filter)
    if match:
        (search_key_name, search_value) = match.groups()
        search_key = getattr(User, search_key_name)
        query = query.filter(search_key == search_value)
    count = int(request.args.get('count', 100))
    start_index = int(request.args.get('startIndex', 1))
    if start_index < 1:
        start_index = 1
    start_index -= 1
    query = query.offset(start_index).limit(count)
    total_results = query.count()
    found = query.all()
    rv = ListResponse(found,
                      start_index=start_index,
                      count=count,
                      total_results=total_results)
    return flask.jsonify(rv.to_scim_resource())
```

8. Review the function which **queries for a specific User**, based on their Id.
    a. Around line 145, locate the **user_get** function.
    This function uses the SQLAlchemy object-to-relational mapping capabilities to perform a database query for a row, based on the "id" column. It returns an HTTP 404 error code if no rows are found.

```
@app.route("/scim/v2/Users/<user_id>", methods=['GET'])
def user_get(user_id):
    try:
        user = User.query.filter_by(id=user_id).one()
    except:
        return scim_error("User not found", 404)
    return render_json(user)
```

9. Review the function **to create a new user**.
    a. Around line 170, locate the function called **users_post**.
    This function generates a unique Id for the user. It then inserts the user into the database as a new row. The new user is then returned back to

Okta as a SCIM resource.

```
@app.route("/scim/v2/Users", methods=['POST'])
def users_post():
    user_resource = request.get_json(force=True)
    user = User(user_resource)
    user.id = str(uuid.uuid4())
    db.session.add(user)
    db.session.commit()
    rv = user.to_scim_resource()
    send_to_browser(rv)
    resp = flask.jsonify(rv)
    resp.headers['Location'] = url_for('user_get',
                                       user_id=user.id,
                                       _external=True)
    return resp, 201
```

10. **Close Atom**.

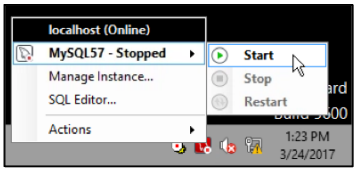
Lab 9-4: Deploy an On Premise Provisioning Connector

Objective	In this lab, you deploy and test the MySQL provisioning connector, a sample provided with the Connector SDK. This connector can provision users to MySQL tables and be used as boilerplate for developing other custom connectors. After deployment, you test the connector using the tester, a utility provided with the SDK.
Scenario	Okta Ice wants to leverage the provisioning capabilities from their Okta org to their internal legacy systems, so admins can provision users from a single pane of glass. One of their legacy system stores users in MySQL tables. We need to integrate Okta provisioning to the user tables, so users can be provisioned to the legacy.
Duration	15-20 minutes

Important: This lab is entirely performed in your Windows Server.

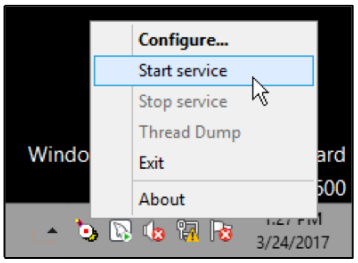
Launch MySQL and Tomcat Servers

1. Access your Windows Server as Administrator.
2. On the task manager bar, right-click the MySQL icon, and then click **MySQL57 - Stopped** > **Start**.



Wait until the MySQL icon turns white. This confirms that MySQL is running.

3. Right-click Tomcat and then click **Start service**.

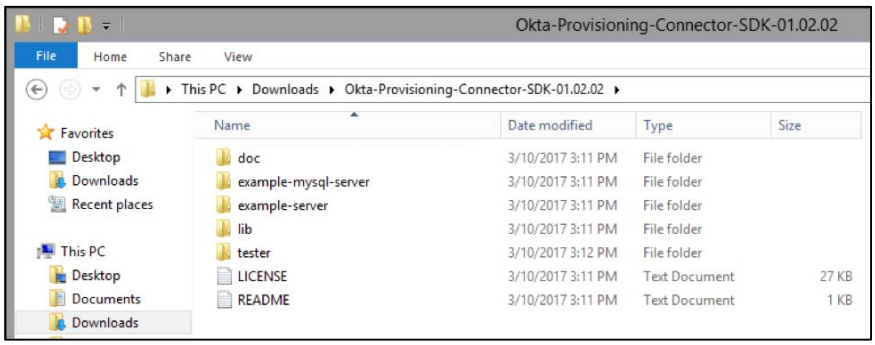


Wait until the Tomcat icon turns green. This confirms that Tomcat is running.

Download the Connector SDK

Tip: The Okta Provisioning Connector SDK contains APIs, code samples, and instructions on how developers can build on premise connectors.

1. Launch a browser and access your Okta org as **okta.admin**.
2. Click **Admin**.
3. Click **Settings** > **Downloads**.
4. Under **Admin Downloads**, download the **Okta Provisioning Connector SDK**.
5. Launch Windows Explorer and navigate to the **Downloads** folder.
6. Right-click the **Okta-Provisioning-Connector-SDK-01.02.02.zip**, click **Extract All**, and then click **Extract**.
7. Open the **Okta-Provisioning-Connector-SDK-01.02.02** folder and verify its contents:



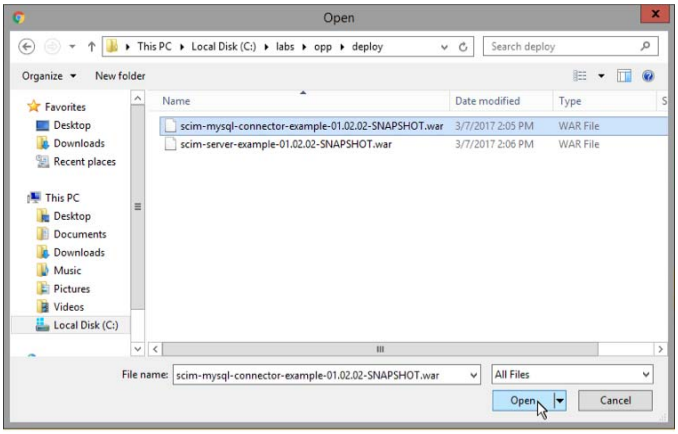
8. **Optionally**, review the main files and folders provided with the Connector SDK:

File/Folder	Description
doc	SDK Javadoc. Typically used by developers as reference to use the SDK APIs.
example-mysql-server and **example-server**	Sample connector source code that can provision users to MySQL and Text Files.
tester	Utility for testing the connector SDK operations before integrating with the Okta Agent.
README	Describes how to compile and use the SDK API.

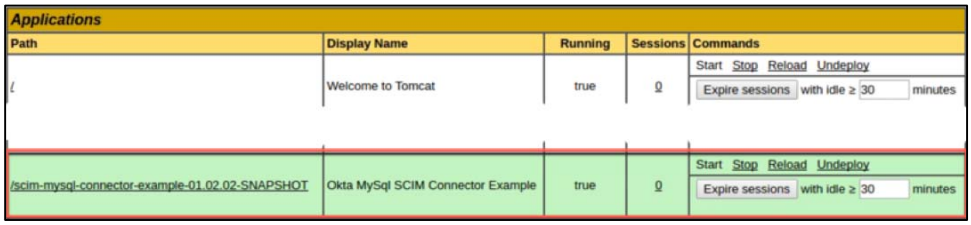
Deploy the MySQL Sample Connector in Tomcat

Note: In this section, you deploy the MySQL sample connector provided with the Okta Provisioning Connector SDK in Tomcat. To save time, this course provides the connector pre-packaged (war file). To learn more about how to package the MySQL sample connector, read the file **example-mysql-server/README.txt** with the SDK.

1. Inside your VM, navigate to:
   http://legacy.oktaice.com:8080/
   The Tomcat home page appears.
2. Click **Tomcat Manager** and log in with the **tomcat** credentials.
   The Web Application Manager page appears.
3. Scroll down to the **Deploy** table and click **Choose File**.
4. Navigate to **C:\labs\opp\deploy** and open the file starting with **scim-mysql-connector**.



5. Click **Deploy**.
6. Confirm that **Okta MySql SCIM Connector Example** is displayed under the Applications table.

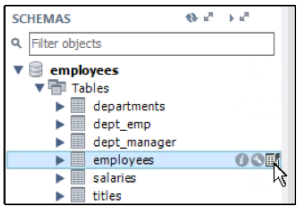


The sample provisioning connector is deployed and running.

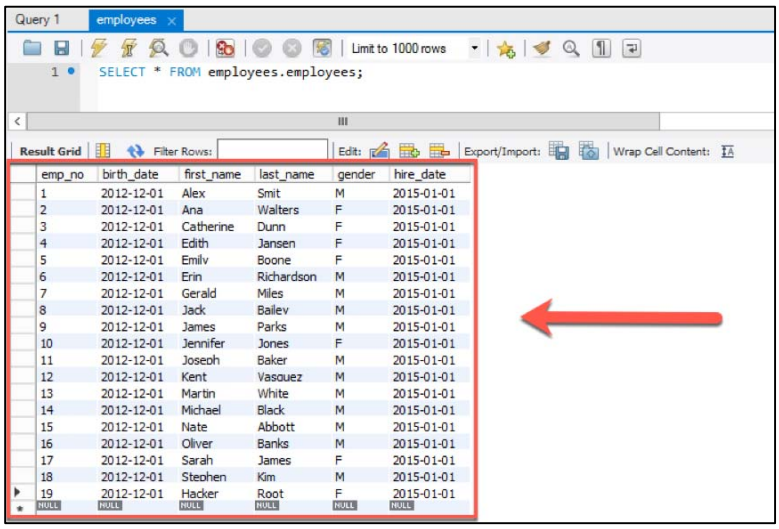
Launch MySQL Workbench

Tip: You will use MySQL Workbench to access MySQL tables.

1. On the Windows task manager bar, right-click the MySQL icon, and then click **Manage Instance**
   MySQL Workbench is launched.
2. Click **Local instance MySQL57**.
3. On the left pane, expand **employees** > **Tables**, click **employees**, and then click the data editor icon ( ).



4. Confirm that the employees' table data is displayed.

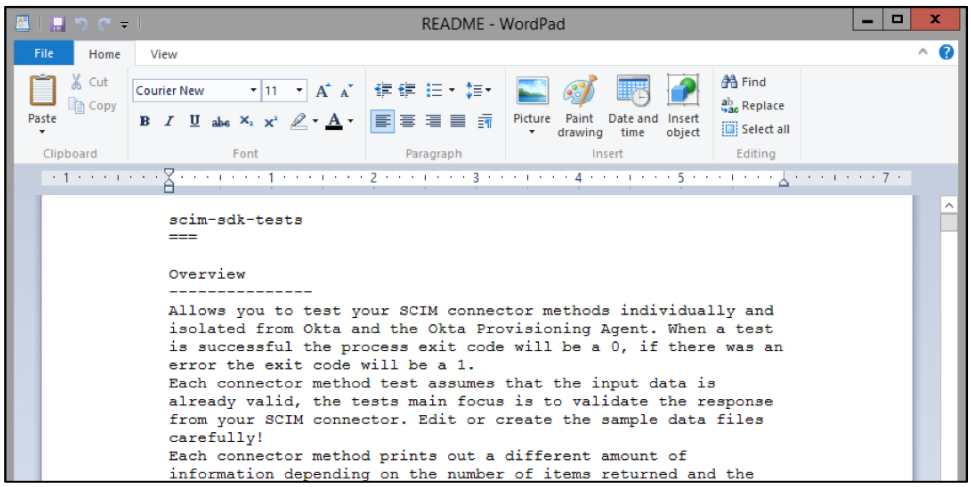


5. Leave MySQL Workbench opened.
   You will use the workbench in the next tasks to confirm that the provisioning operations are working.

Test the Connector

In this section, you test the connector using the tester utility provided with the Connector SDK.

1. Launch Windows Explorer and navigate to:
   C:\Users\Administrator\Downloads\Okta-Provisioning-Connector-SDK-01.02.02\tester
2. Optionally, review the README.txt contents in WordPad. It covers how you can use the tester utility.



3. Leave Windows Explorer open.
   You will return to this window to verify the tester results.
4. Open the command prompt and navigate to the SDK test folder:
   cd C:\Users\Administrator\Downloads\Okta-Provisioning-Connector-SDK-01.02.02\tester

   Tip: To save time and avoid typos, you open the C:\labs\opp\commands.txt to copy and paste the commands.
5. Export the variables by typing the following:

```
set URL="http://legacy.oktaice.com:8080/scim-mysql-connector-example-01.02.02-SNAPSHOT"
set TEST="C:\labs\opp\test-data"
```

6. To test the **user import**, execute the following command:

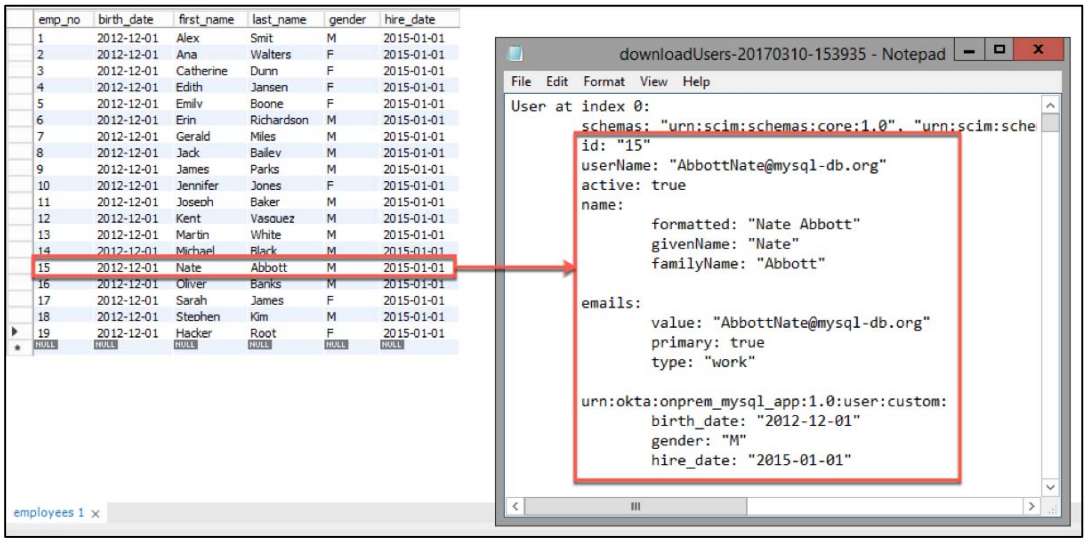
```
java -jar scim-sdk-tests.jar -url %URL% -method downloadUsers
```

   The tester will return the message "**19 Users returned**" and create a text file starting with **downloadUsers**.

7. Return to the tester folder (`C:\Users\Administrator\Downloads\Okta-Provisioning-Connector-SDK-01.02.02\tester`), open the file starting with **downloadUsers** in Notepad, and review its contents.

The file displays an SCIM-formatted list of users from MySQL. This confirms that the connector is working.

8. **Optionally**, compare the downloadUsers results with the employee data displayed in MySQL Workbench.



Note: The downloadResults file shows one user register equivalent to each entry in the employees table. The email and username are set by the sample code as `last_name+first_name+"@mysql-db.org"`.

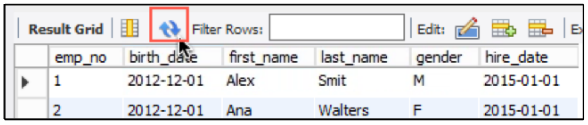
Tip: Observe how the table registry is translated to the SCIM payload in JSON format.

9. Close the text editor.
10. To test the **user creation**, execute the following command:

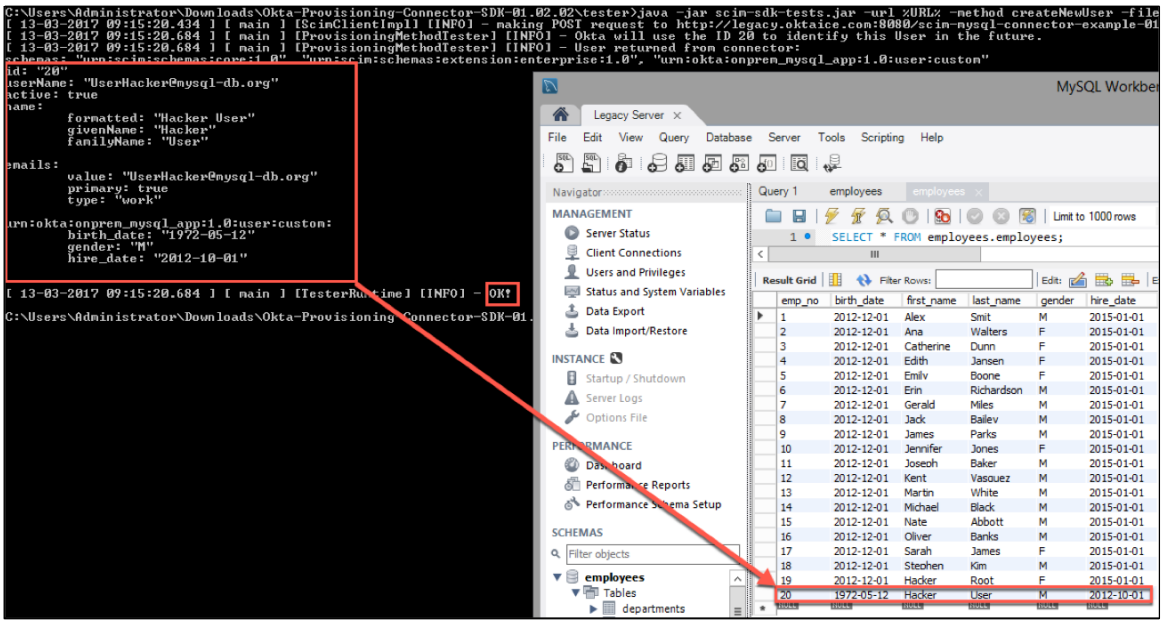
```
java -jar scim-sdk-tests.jar -url %URL% -method
createNewUser -file %TEST%/createNewUser.json
```

Tester creates a new user (Hacker User) in MySQL, display the user information in JSON format, and the success message "**OK!**".

11. In MySQL Workbench, click refresh.



12. Confirm that the Hacker User entry is created:

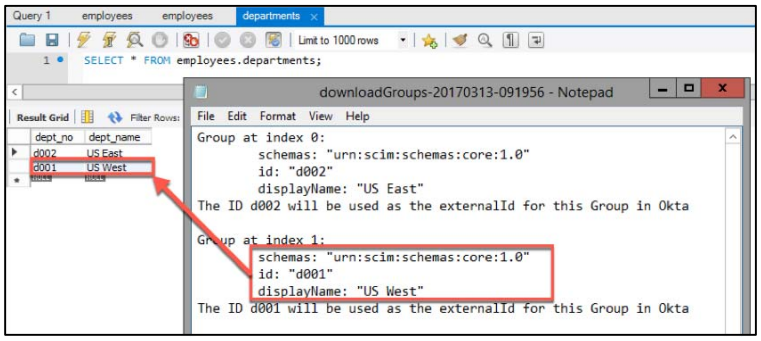


13. **Optionally**, return to the command prompt and test the group import:

```
java -jar scim-sdk-tests.jar -url %URL% -method
downloadGroups
```

The tester will return the message "**2 Groups returned**" and create a text file starting with **downloadGroups-**.

14. **Optionally**, verify the **downloadGroups** content against the departments table in MySQL Workbench.



15. Close the command prompt.

Lab 9-5: Install and Configure the Okta Provisioning Agent

Objective	In this lab, you install and configure the Okta Provisioning Agent. The agent acts as a broker. It receives provisioning requests from Okta and propagates the operations to on premise systems.
Scenario	Okta Ice has a custom connector fully tested and running on their premises. This connector can execute provisioning operations in their legacy MySQL system. Now it's time to setup the Provisioning Agent. This agent works similarly to the Active Directory agent, and provide a secure connection between the Okta Ice org in the cloud and the connectors on premises.
Duration	5-10 minutes

Download the Provisioning Agent

1. In your Windows VM, access your Okta org as **okta.admin**.
2. Click **Admin**.
3. Click **Settings** > **Downloads**.
4. Under **Admin Downloads**, download the **Okta Provisioning Agent (Windows x64 EXE)**.

   Tip: The provisioning agent connects your Okta org with on premises applications via Okta SDK connectors or SCIM servers.
5. Keep your browser opened.

Install and Configure the Provisioning Agent

1. Launch the Okta Provisioning Agent installer. When prompted, click **Run**.
2. Click **Next**, **Next**, and then click **Install**.

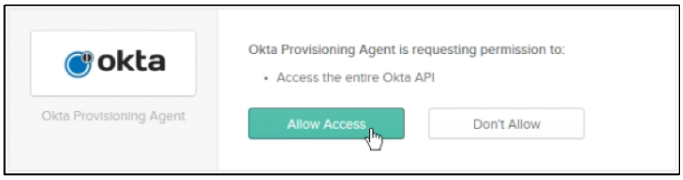
   Wait for few seconds until the binary installation is completed.
3. Provide the information about your environment as follows and then click **Next**.

Attribute	Value
Okta Environment	Preview
Okta Customer Domain	https://oktaiceXXX.oktapreview.com

4. Ignore the proxy settings and click **Next**.

   A login form for your Okta org appears.
5. Sign in as **okta.admin**.

6. Click **Allow Access**.

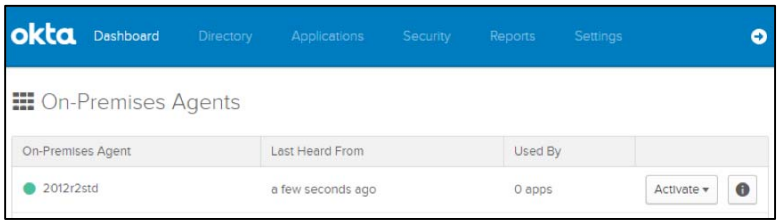


7. Wait until the setup is completed and then click **Finish**.

Verify the Agent Status

1. Access your Okta org as **okta.admin**.
2. Click **Admin**.
3. Click **Dashboard** > **Agents**.

   The On-Premises Agent page displays the agent 2012r2std with a green status.



   This confirms that the agent is up and running.

Lab 9-6: Integrate the Custom Application Provisioning

Objective	In this lab, you integrate your Okta org with the on-premises system for provisioning users. After configuration, you test the import and provisioning operations and confirm the results in your MySQL database.
Scenario	Okta Ice has a custom connector and the provisioning agent running on their premises. Finally, it's time to register the legacy application and test the provisioning functionality.
Duration	10-15 minutes

Launch ngrok

Note: The Okta On Premise Agent requires the On Premise Connector to be accessed only via HTTPS with a valid certificate. In this section, you launch ngrok to use a valid HTTPS certificate.

1. In your windows VM, close all command prompt terminals.
2. Launch a new command prompt terminal and enter:
   ngrok http 8080 –host-header="localhost:8080"
3. Record the https url in Notepad

Register the MySQL Application

1. Access your Okta org as **okta.admin**.
2. Click **Admin**.
3. Click **Applications** and then click **Add Application**.
4. Click **Create New App**.
5. Select **Secure Web Application (SWA)** and then click **Create**.
6. Provide the following information:

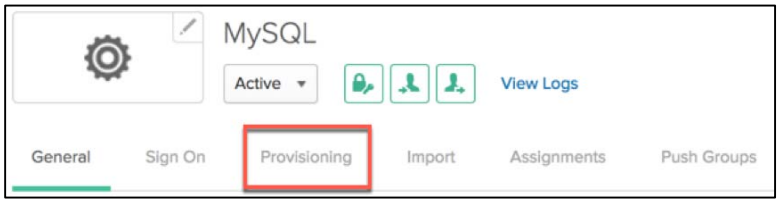
Attribute	Value
App Name	MySQL
App's Login Page URL	http://legacy.oktaice.com
Do not display application icon to users	selected
Do not display application icon in the Okta Mobile app	selected
This is an internal application that we created	selected

7. Click **Finish**.

The application is registered.

Enable and Configure Provisioning

1. In MySQL page, click **General**.
2. Under **App Settings**, click **Edit**.
3. Select **Enable on-premises provisioning configuration** and click **Save**.
   The Provisioning option appears in the Application Menu.



4. Click **Provisioning**.
5. Click **Configure SCIM Connector**
6. In the Connector **Configuration section** provide the following information:

Attribute	Value
SCIM connector base URL	NGROK_HTTPS_URL/scim-mysql-connector-example-01.02.02-SNAPSHOT Replace NGROK_HTTPS_URL with your ngrok url. For example: https://384996ed.ngrok.io/scim-mysql-connector-example-01.02.02-SNAPSHOT
Authorization type	None
Unique user field name	userName
Store updates to the user's app profile returned by the connector	selected
Timeout for API Calls	30 seconds
Connect to these agents	2012r2std

7. Click **Save**.
8. Click **Enable Provisioning**.
9. Select **Enable provisioning features**.
10. Enable **Create Users** and **Update User Attributes**, and then click **Save**.

Import Users

1. Click the **Import** tab.
2. Click **Import Now**.
3. Wait until the import is completed.
4. Confirm that the import retrieves 20 users and 2 groups, and then click **OK**.

This confirms that your Okta org can import users from MySQL.

5. **Optionally**, compare the results against MySQL Workbench.

Provision Users

1. Click **Assignments**.
2. Click **Assign** > **Assign to People**.
3. Search **okta.admin** user, click **Assign**, click **Save and Go Back**, and then click **Done**.
   A confirmation message is displayed.
4. Return to MySQL Workbench and confirm that the **okta.admin** account is provisioned.

Stop the Services

1. In your windows VM, close MySQL Workbench and the terminal running ngrok.
2. On the task manager bar, right-click the MySQL icon, and then click **MySQL57** > **Stop**.
3. Right-click Tomcat and then click **Stop service**.

Lab 10-1: Implement Social Authentication with Facebook

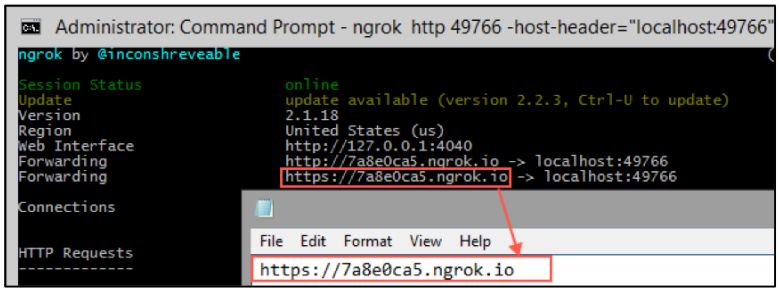
Objective	Use the Okta Sign-In Widget to create a login web page, and use that to verify a CORS connection that you will define.
Duration	40 minutes

Configure the Okta Sign-in Widget and Start the Web Server

If not already started, launch the remote VM in ReadyTech to begin the labs. Everything should be done within the remote virtual machine (VM).

1. Inside the VM, launch **Windows Explorer**.
2. In the left folder pane, click **Local Disk (C:).** Navigate to **c:\labs\social-auth**.
3. Right click on "**login-to-okta.html**" and select **Open with > Atom**.
4. Around line 25, locate the declaration and assignment of the 'orgUrl' variable:
   ```
   var orgUrl = 'https://example.okta.com';
   ```
5. Replace 'https://example.okta.com' with '**https://oktaice###.oktapreview.com**', substituting your unique assigned org number for '###'.
6. On the next line, locate the declaration and assignment of the 'redirectUrl' variable:
   ```
   var redirectUrl = 'http://localhost:8000/signed-in.html';
   ```
7. Replace ''http://localhost:8000/signed-in.html' with '**https://oktaice###.oktapreview.com/app/UserHome**', substituting your unique assigned org number for '###'.
8. In Atom, from the menu, click **File | Save**.
9. Start a **Command Prompt** window by clicking the icon on the Windows task bar.
10. Change directories by entering the command **cd \labs\social-auth**.
11. Enter in the command **python –m SimpleHTTPServer**. If prompted for approval by Windows Firewall, click "**Allow access**". Leave this window open so the process can continue to run.
12. Open a new Command Prompt window by right-clicking on the Command Prompt icon in the Windows task bar and selecting **Command Prompt**.
13. At the command line prompt, enter the following command: **ngrok http 8000**. Leave this window open so the process can continue to run.

14. Write down the generated HTTPS forwarding address. You will need that for CORS.



Enable CORS in Your Okta Org and Test

1. If you are not already signed into Okta, open a new browser tab and log into your Okta org.
2. If necessary, navigate to the **Admin** interface.
3. Point to **Security** and click **API**.
4. Click the **Trusted Origins** tab.
5. Click **Add Origin**.
6. Populate the fields as follows:

   Name: Social Auth Sign In
   Origin URL: type in the forwarding URL from above.
   E.g. **https://f0a29874.ngrok.io**
   CORS: selected
   Redirect: cleared
7. Click **Save**.
8. Sign out of Okta.
9. In the same web browser tab, go to your forwarding URL plus the login-to-okta.html. For example: **http://f02a29874.ngrok.io/login-to-okta.html**
10. Sign in using your Okta credentials.
11. Leave this browser tab open.

Configure a Facebook App for Facebook Login

1. In the VM, open a new browser tab.
2. Log in to Facebook using your own credentials.
3. Afterwards, change the URL in the browser's address field: **https://developers.facebook.com/apps/**.
4. In the upper right corner, click **My Apps** > **Add a New App**.
5. Populate the fields as follows:

   Display Name: Okta Social Auth

Contact Email: enter in your email address
Category: Utilities

6. Click **Create App ID**.
7. If a Security Check appears, follow the directions to get through the bot detection.
8. In the left pane, click **App Review**.
9. In the area **Make Okta Social Auth public?**, toggle the **Your app is in development and unavailable to the public** option to **Yes**.
10. In the **Make App Public?** confirmation window, click **Confirm**.
11. In the left pane, under Dashboard, click **Settings**.
12. Record the **App ID** and **App Secret** in Notepad.
    Note: We will need those when we are creating our Identity Provider in the next section.
13. Leave this browser tab open.

Configure Social Authentication in Your Okta Org

1. Back in the browser tab currently logged into Okta, change to the **Admin** view.
2. From the admin menu, select **Security** and then click **Identity Providers**.
3. Click **Add Identity Provider** and then click **Add Facebook**.
4. In the **Name** field, type the following: **Log in with Facebook**
5. In the **IdP Username** field, click the **down arrow** and then click **idpuser.email**.
6. In the **JIT Settings** section, check the box for **Profile Master: Update attributes for existing users**.
7. Scroll down to **Facebook Settings**.
8. From the browser tab for Facebook, copy the **App ID** value to the **Client Id** field on the Add Identity Provider configuration page in the browser tab for Okta.
9. From the Facebook browser tab, next to the **App Secret** field, click **Show**. If prompted for your password, enter it and click **Submit**.
10. Copy the Facebook **App Secret** value to the **Client Secret** field on the Add Identity Provider configuration page in Okta.
11. In the browser tab for Okta, click **Add Identity Provider**.
12. Copy the **Authorize URL** and the **Redirect URI** into Notepad.
13. In the VM, launch **Notepad**.
14. Paste the **Authorize URL** value into Notepad.
15. Back in the browser tab for Okta, copy the **Redirect URI.**
16. Return to the browser tab running **Facebook Developer**.
17. In the left pane, click **+ Add Product**.
18. In the center pane, next to **Facebook Login**, click **Get Started**.

19. In the left pane, click on **Facebook Login**, then underneath, click **Settings**.
20. In the **Valid OAuth redirect URIs** field, paste in the **Redirect URI** from above.
21. In the lower right corner, click **Save Changes**.

Creating the OIDC Application using the AIW

1. Return to the browser tab with the Okta Admin page.
2. Point to **Applications** and click **Applications**.
3. Click **Add Application**.
4. Under **Can't find an app?**, click **Create New App**.
5. In the **Create a New Application Integration** dialog box, perform the following:
    a. For the **Platform** picklist, select **Single Page App (SPA)**.
    b. Next to **Sign on method**, leave the default to **OpenID Connect.**
    c. Click **Create**.
6. On the **General App Settings** page, in the **App name** field, enter **Facebook Social Auth OIDC Client** and then click **Next**.
7. On the **Configure OpenID Connect** page, next to **Redirect URIs**, click **Add URI**. Enter in: **https://oktaice###.oktapreview.com/app/UserHome**, replacing '###' with your unique number.
8. Click **Finish**.
9. Under the Client Credentials, copy the **Client ID.**

Edit the Authorized URL in Notepad

1. Back in **Notepad**, edit your **Authorized URL:**
   `https://oktaice###.oktapreview.com/oauth2/v1/authorize?idp=0oa6wn6ebfgNwI5Nk0h7&client_id={clientId}&response_type={responseType}&response_mode={responseMode}&scope={scopes}&redirect_uri={redirectUri}&state={state}&nonce={nonce}`
2. Paste the Client ID in replacing the **{clientId}** with value you just copied from above.
3. Replace **{responseType}** with **id_token**.
4. Delete **&response_mode={responseMode}**.
5. Replace **{scopes}** with **openid%20email%20profile**.
6. Replace **{redirectUri}** with **https://oktaice###.oktapreview.com/app/UserHome**, replacing '###' with your unique org number.
7. Replace **{state}** with **someState**.
8. Replace **{nonce}** with **someNonce**.
9. Your final URL will look something like this:
   `https://oktaice000.oktapreview.com/oauth2/v1/authorize?idp=0oa6wn6ebfgNwI5Nk0h7&client_id=belKlGhaGy71hQr8wWPI&scope=openi`

```
d%20email%20profile&response_type=id_token&redirect_uri=https
://oktaice000.oktapreview.com/app/UserHome&state=someState&no
nce=someNonce
```

10. **Select all** the text with your mouse (or use **Ctrl + a**) and then **Copy** your URL with **Ctrl + c** or the right click menu.

Modify Your Login Widget File

1. Back in **Atom**, in your **login-to-okta.html** file, locate the following line of code:
```
var oktaSignIn = new OktaSignIn({baseUrl: orgUrl});
```
2. Before the first reference to baseUrl, place the cursor and hit **Enter**.
3. After the second reference to baseUrl, add a **comma** and type **Enter**. Your code should look similar to below:
```
var oktaSignIn = new OktaSignIn({
     baseUrl: orgUrl,
});
```
4. Update the JSON to include a new property called **helpLinks**, which will contain a custom URL that will initiate the social authentication process.
```
var oktaSignIn = new OktaSignIn({
  baseUrl: orgUrl,
  helpLinks: {
    custom: [
      { text: 'Login with Facebook', href: '' }
    ]
  }
});
```
5. Update the empty value of the **href** by pasting in your social authorization URL from above. The final code will look like this:
```
var oktaSignIn = new OktaSignIn({
  baseUrl: baseUrl,
    helpLinks: {
      custom: [
{ text: 'Login with Facebook', href:
'https://oktaice###.oktapreview.com/oauth2/v1/authorize?idp
=0oa6wn6ebfgNwI5Nk0h7&client_id=belKlGhaGy71hQr8wWPI&scope=
openid%20email%20profile&response_type=id_token&redirect_ur
i=https://oktaice###.oktapreview.com/app/UserHome&state=som
eState&nonce=someNonce' }
      ]
    }
});
```

6. Select **File - Save**.

Test Social Authentication

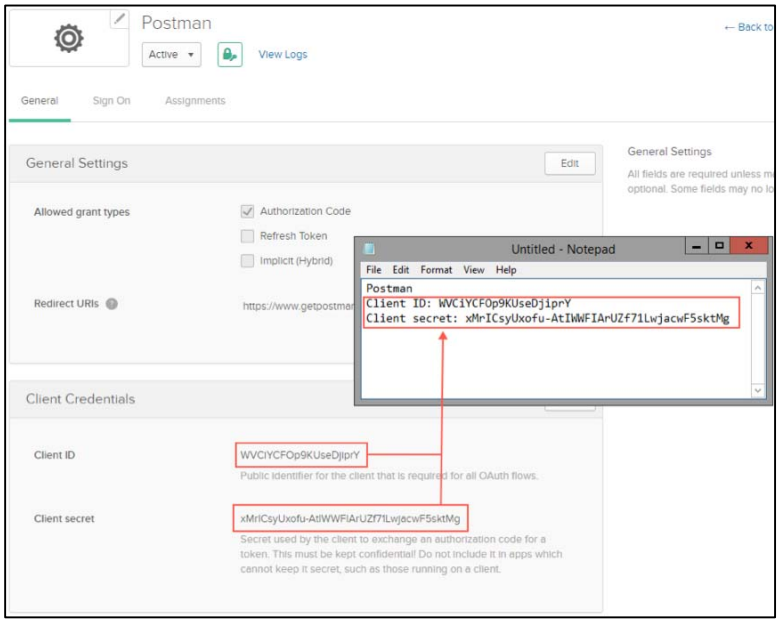
1. In the browser tab running Okta, **Sign out**.
2. In the browser running Facebook, **Log Out of Facebook**.
3. In the address bar, enter in your forwarding URL plus the login.html.
   For example: **https://f02a29874.ngrok.io/login-to-okta.html**
4. Click **Need help signing in?**, then click **Login with Facebook**.
5. Log in with your Facebook credentials. When prompted to consent to sharing your profile data, allow access.
6. You should be redirected to the Okta apps home page.
7. **Sign out** of Okta.
8. Sign back into Okta using your administrator credentials.
9. Click **Admin**.
10. From the menu, select **Directory**, then click **People**.
11. Click the new user account associated with your Facebook account.
12. You should now see yourself as a user in Okta, with **Profile mastered by Facebook IdP**.

Lab 10-2: Create an OAuth and OIDC Application Using the AIW

Objective	Register an OAuth and OIDC application through the Okta AIW.
Scenario	Okta ICE hired a development team to create an innovative platform to promote their ice creams. The platform roadmap includes: • Start with a Web App accessed only by the Marketing and Sales team. • Get the Marketing and Sales team feedback. • Publish a Mobile Application for Android and iOS. • Open the platform for public users. • Expand the platform with new services based on the Marketing/Sales needs. • Support innovation in the API economy, so the application can be exposed in new places with API (no UI) integration. For example, as a Facebook Chatbot. The roadmap relies on Okta SSO and Platform to provide: • SSO and authentication for different kinds of apps, including apps that don't support SAML or cookies. • Authentication and authorization for APIs. • Security in app-to-app communications. • Flexibility to support an API-first microservices architecture, independent of platform provider. During the entire Lab 10, you perform a sample configuration, test the OAuth and OIDC in your Okta org, and prepare a development kit. This kit will help the development team creating the platform. In Lab 10-1, you register a sample application to test the OAuth and OIDC provided with Okta SSO. This sample registration can be used with the initial Web App that will be available for the Marketing and Sales team.
Duration	5 minutes

1. Access your Okta org as **okta.admin**.
2. Click **Admin**.
3. Click **Applications**.
4. Click **Add Application**.
5. Click **Create New App**.
6. Select **OpenID Connect** and then click **Create**.

7. Enter `Postman` as **Application Name** and then click **Next**.
8. Click **Add URI**.
9. Enter **https://www.getpostman.com/oauth2/callback** as **Redirect URIs** and then click **Finish**.
   The Postman application is registered.
10. Record the **Client ID** and the **Client secret** located under client credentials.



Note: Client credentials are a set of credentials used by applications – in this lab, Postman – to authenticate against Okta. These credentials are required be sent by the application in some of the OAuth/OIDC requests, such as in the request for an access token.

11. Click **Assignments**.
12. Assign the application to the **Marketing** and the **Sales** groups, as well as the **okta.admin** user.

Lab 10-3: Test the OIDC Single Sign-On

Objective	Test the OIDC SSO flow using Postman.
Scenario	After registering the sample OAuth/OIDC App in Okta, you test the OAuth/OIDC Okta SSO with Postman. This test will be shared with the development team, so they can verify the authentication any time while developing or troubleshooting the app.
Duration	10 minutes

Launch and Configure Postman

Note: In this lab, you configure Postman, a popular client for testing REST APIs. Postman can test OAuth/OIDC Single Sign-On and authorization flows. The Okta developer's portal (http://developer.okta.com/docs/api/resources/oauth-clients.html) provides sample Postman collections that you can use for testing SSO and API AM requests. To save time, the sample collections are loaded in your course environment.

1. From your windows VM desktop, launch **Postman**.
2. In the upper right corner, click **Settings (gear icon)** > **Manage Environments**.
3. Click **Okta ICE**.
4. Update the environment variables as follows and then click **Update**.

Attribute	Value
url	Your Okta ICE org url. For example, `https://oktaice`**`XXX`**`.oktapreview.com`
clientId	The **Postman**'s client id (obtained in lab 11-1). For example, `fzyV0kSHGAQ8k2hy2I8Y`
clientSecret	The **Postman**'s client secret (obtained in lab 11-1). For example, `2ztky6XUAS1aLA4X7yK6t4fkZ3Xz`
redirectUri	`https://www.getpostman.com/oauth2/callback`

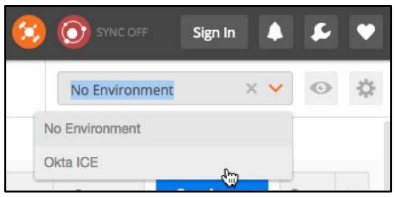
5. Close the **Manage Environments** window.

Get the OIDC Discovery Document

Notes:

- The OIDC Discovery Document is a public REST API request – defined by the OIDC discovery standard – that provides metadata about the OIDC configuration set in your Okta org in JSON format.
- This information can be used by client applications – and developers – for communicating with your Okta org.
- To learn more about the discovery document, visit: http://developer.okta.com/docs/api/resources/oidc.html

1. In Postman, select **Okta ICE** as environment.



2. Expand the **OAuth 2.0 (Okta API)** Collection and then click **OpenID Connect** > **Get OpenID Provider Metadata**.
3. Observe the following attributes in the **Get OpenID Provider Metadata** request:

Attribute	Comment
Request URL	The request URL contains dynamic variables: `{{url}}` and `{{clientId}}`. Postman replaces these variables with values from your environment (Okta ICE) every time you send requests.
Authorization Type	The Authorization type is selected as **No Auth** because the request for OIDC provider metadata is public.

4. Click **Send**.
5. Verify that the response body displays your Okta org OIDC discovery documentation in JSON format.

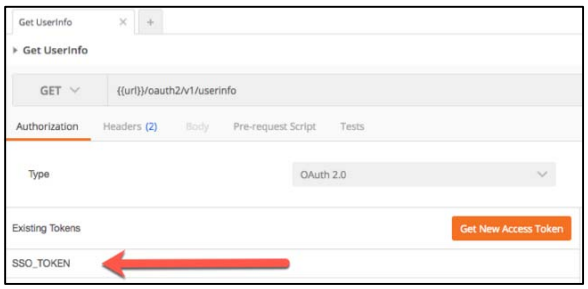
Obtain open_id and Access Tokens

1. Under Collections, click **OpenID Connect** > **Get User Info**.
2. In the Authorization tab, select **OAuth 2.0** and then click **Get New Access Token**.
3. Enter the information as follows and then click **Request Token**.

Attribute	Value
Token Name	`SSO_TOKEN`
Auth URL	`{{url}}/oauth2/v1/authorize`
Access Token URL	`{{url}}/oauth2/v1/token`
Client ID	`{{clientId}}`
Client Secret	`{{clientSecret}}`
Scope	`openid profile`
Grant Type	`Authorization Code`
Request access token locally	`Deselected`

Postman launches a popup with the login form from your Okta org.

4. Sign in as **martin.white**.
   Postman closes the popup after a successful authentication and displays the SSO_TOKEN under Existing Tokens.



5. Click **SSO_TOKEN**.
   Postman presents the access_token and the id_token in the right pane.

Test the Access Token

1. Select Add token to **Header** contents, and then click **Use Token**.
   This updates the request header with the access_token from your token request.
2. Click **Header** and confirm that the Authorization header is set.
3. Click **Send**.
4. Confirm that the results display information about **martin.white**.

```
{
  "sub": "00u9x7c4ebvGTTfXO0h7",
  "name": "Martin White",
  "locale": "en-US",
  "preferred_username": "martin.white@oktaice.com",
  "given_name": "Martin",
  "family_name": "White",
  "zoneinfo": "America/Los_Angeles",
  "updated_at": 1490134115
}
```

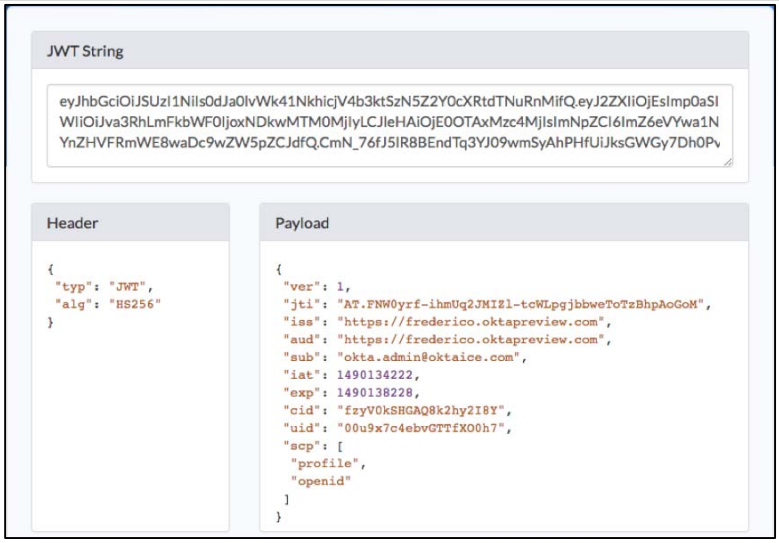
   This confirms that the access token is fully functional.
5. **Optionally**, repeat the **Get UserInfo** requests with different values for the `Authorization` header and observe the Request Status and headers retrieved by Okta.

Authorization Header	Status or HTTP Response	Headers
`"Bearer "` (blank)	`400 Bad Request`	WWW-Authenticate: Bearer error="invalid_request", error_description="The access token is missing."
`Bearer 123.456.7890`	`401 Unauthorized`	WWW-Authenticate: Bearer error="invalid_token", error_description="The access token is invalid."

Check the access_token and the open_id Token Contents

1. Under Get UserInfo, click **Authorization** > **SSO_TOKEN**.
2. Select and right-click the access_token contents, and then click **Copy**.
3. In your browser, access https://www.jsonwebtoken.io/
4. Paste the access_token in the JWT String field.
5. Observe the Header and the Payload fields.

Tip: The header contains information about how the JWT is signed while the payload contains the JWT main information. The access_token payload usually carries information about what can be accessed with the token – scopes under the scp field.



6. **Optionally**, repeat the previous steps to verify the open_id token contents with the following exceptions:
    a. In step 2, make sure you select and copy open_id contents.
    b. In step 5, verify that the open_id payload carries information about **martin.white**.

Checkpoint: At this point, you explored the OIDC Single Sign-On feature provided by Okta SSO natively.

Okta SSO provides OIDC via a default Resource Server. This resource server supports authentication features that are comparable to the SAML SSO.

Okta API Access Management (API AM), allows you to define your own Authorization Servers with extended features for OIDC and OAuth that enable the consumption by APIs. With API AM, you can:

- Define Access Policies specific for API access.
- Define custom OAuth tokens and scopes.
- Define custom OIDC Claims for UD attributes or based on Okta's expression language.
- Provide security in APP-to-APP communications.
- Provide security in Microservices, B2B, and IoT scenarios.

In the next labs, you explore the extended capabilities provided by API AM.

Lab 10-4: Configure API AM

Objective	In this lab, you configure features that are delivered by Okta's API AM. This includes registering and configuring: • An OAuth Service Application. • An Authorization Server. • A custom scope. • A custom claim. • An access policies with rules.

Scenario	After testing the OIDC/OAuth SSO, you expand the configuration to address the following requirements:
	• The WebApp need to know whether an authenticated user is from the Sales or the Marketing group. The marketing group has access to more information than the sales group.
	• The Web App need to make an API call to an external Resource Server (`https://api.oktaice.com/promos`.) This API will returns promotions based on the user department.
	• The `/promos` Resource Server will be accessed by another APP (Service App) that will update REST API with new specials.
	To meet the requirements, you implement a sample configuration that includes:
	• Implement a sample Resource Server that:
	  • Provide the group information via a custom claim.
	  • Secure the new REST API endpoint (`/promos`) using a custom scope.
	  • Generate tokens only for the WebApp (accessed by users) and the Service App (that updates the specials).
	• Register a Service App that will update the promotions.
	This sample configuration will be used by the development team to test the security while developing the app.

Duration	15 minutes

Register an OAuth Service Application

OAuth Service Applications are apps that either:

- **Provide REST APIs services for other apps:** These apps – also known as Resource Servers – act in the back-end receiving, validating, and answering to REST API calls from other apps. Secure REST API services, when called, must confirm that a token issued by Okta's API AM is provided with the request, check the token authenticity – via digital signature or introspection, and the token authorization and user information.
- **Consume REST APIs in App-to-App integrations:** These apps can request tokens get access to REST APIs using the OAuth client credentials flow. This flow does not require a user authentication and is typically used in app-to-app communication where there's no user, like in B2B and batch processing.

Although these applications do not have a direct interface with end-users, they can communicate with Okta API AM to either request or validate OAuth tokens.

1. Access your Okta org as **okta.admin**.
2. Click **Admin**.
3. Click **Applications**.
4. Click **Register OAuth Service**.
5. Enter `Service App` as **Service Name** and then click **Save**.
6. Record the **Client ID** and the **Client secret** located under client credentials in Notepad.

   Tip: In the next lab, you will use Postman to emulate the Service App authentication using the client credentials flow.

Register an API AM Authorization Server

Notes:

- The API AM Authorization Server is a REST API service – located under `https://oktaicexxx.oktapreview.com/oauth2/id` – that can issue, validate, and revoke OAuth/OIDC tokens.
- Authorization Servers are the main entry in API AM and from where all API AM features are managed – with exception managing OAuth and OIDC applications.
- In this lab, you register a new Authorization Server in API AM.
- This Authorization Server will support custom scopes and claims, as well as access policies for OIDC and Service Applications.

1. Click **Security** > **API**.
2. Click **Add Authorization Server**.

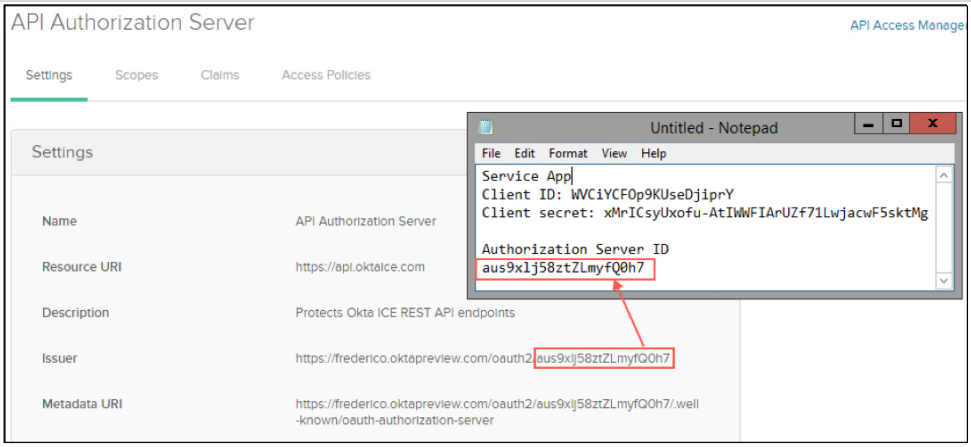
3. Enter the information as follows and then click **Save**.

Attribute	Value
Name	`API Authorization Server`
Resource URI	`https://api.oktaice.com`
Description	`Protects Okta ICE REST API endpoints`

The API Authorization Server page is displayed.

4. Record the Authorization server ID.

Note: The Authorization server ID can be extracted from the Issuer URL: `https://oktaicexxx.oktapreview.com/oauth2/{{authorizationServerId}}`

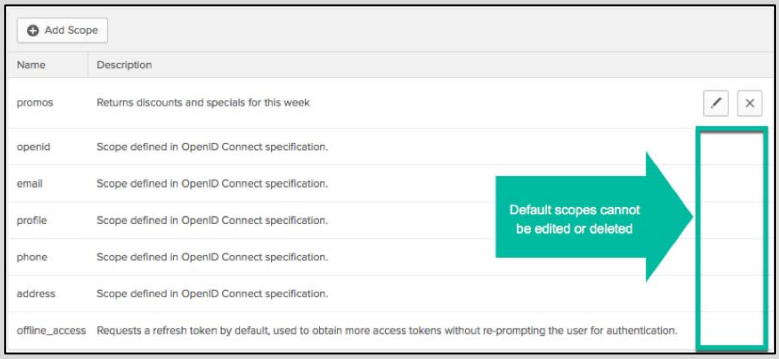


Register a Custom Scope

1. In the API Authorization Server page, click **Scopes**.
2. Click **Add Scope**.
3. Enter `promos` as **Name**, `Return discounts and specials for this week` as **Description**, and then click **Create**.

   The promos scope is listed in the table.

Tip: Scopes provided by default with OAuth and OIDC – such as `openid`, or `offline_access` – cannot be edited or deleted.



Register a Custom Claim

1. Click **Claims**.
2. Click **Add Claim**.
3. Enter the information as follows and then click **Create**.

Attribute	Value
Name	`department`
Claim type	`Access Token`
Value Type	`Expression`
Mapping	`user.department`
Include in	`The following scopes: promos`

The department claim is listed in the table.

Register an Access Policy and Rules

Note: The API AM access policies and rules controls who can request tokens in the Authorization Server. This includes:

- What applications can request tokens
- What OAuth flows are supported for the token request, and
- What scopes, tokens, and claims will be granted by the Authorization Server.

In this lab, you'll configure an access policy with rules, so the Authorization Server will issue tokens with custom scopes just for the `Service App` and the `Postman` applications.

1. Click **Access Policies**.
2. Click **Add Policy**.
3. Enter the information as follows and then click **Create Policy**.

Attribute	Value
Name	`Okta Ice Custom Policy`
Description	`Policy for Okta Ice for custom APIs`
Assign to	`Postman and Service App`

4. Click **Add Rule**.
5. Enter the information as follows and then click **Create Rule**.

Attribute	Value
Rule Name	`App to App Rule`
IF Grant type is	Select only `Client credentials`.
THEN Grant these scopes	The following scopes: `promos`

6. Create a second rule with the following attributes:

Attribute	Value
Rule Name	`User to App Rule`
IF Grant type is	Select only `Authorization Code` and `Implicit`.
THEN Grant these scopes	All scopes.

Lab 10-5: Test API AM requests

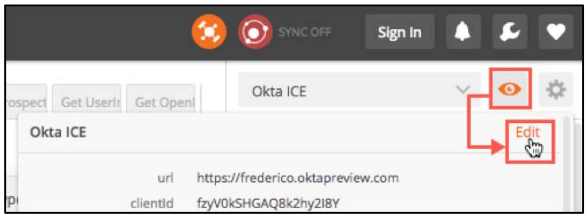
Objective	In this lab, you test the OAuth and OIDC authentications provided by an Okta API AM Authorization Server.
Scenario	After registering the OAuth/OIDC Resource Server in in Okta, you test the configuration with Postman. This test will be shared with the development team, so they can verify the authentication any time while developing or troubleshooting the app.
Duration	10-15 minutes

Update the Okta Ice Environment Variables

1. **Restart** Google Chrome and Postman.

   Tip: You restart Chrome and Postman to flush the cache. Otherwise, Postman will not show you the sign-in form when requesting new tokens.

2. Click the eye icon (located in the top-right corner) and then click **Edit**.



3. Update the environment variables as follows and then click **Update**.

Attribute	Value
serviceClientId	The **Service App**'s client id (obtained in lab 11-3). For example, `95qBlAgLoqZOoyLw2XCf`
serviceClientSecret	The **Service App**'s client secret (obtained in lab 11-3). For example, `ijnPGOEx1Vy2KGirQ88QqKp1d`
authorizationServerID	The **API Authorization Server** id (obtained in lab 11-3). For example, `aus9w0z4988od3YdS0h7`

4. Close the **Manage Environments** pop up.

Get the OIDC Discovery Document

1. Open **API Access Management – OpenID Connect** > **Get OpenID Provider Metadata**.
2. Click **Send**.
3. From results, confirm that the JSON discovery document displays a different issuer from the discovery document obtained in the Lab 11-2.

   Note: In this task, you requested the discovery document from the Authorization Server you've registered during Lab 11-3. The metadata for this authorization server (`https://oktaicexxx.oktapreview.com/oauth2/id`) is different from the default authorization server provided with Okta SSO (`https://oktaicexxx.oktapreview.com/oauth2`).

Test the OIDC SSO with Custom Scopes and Claims

1. Under Collections, click **API Access Management – OpenID Connect** > **Get User Info**.
2. Under Authorization, select **OAuth 2.0** and then click **Get New Access Token**.
3. Enter the information as follows and then click **Request Token**.

Attribute	Value
Token Name	`APIAM_TOKEN`
Auth URL	`{{url}}/oauth2/{{authorizationServerId}}/v1/authorize`
Access Token URL	`{{url}}/oauth2/{{authorizationServerId}}/v1/token`
Client ID	`{{clientId}}`
Client Secret	`{{clientSecret}}`
Scope	`openid profile promos`
Grant Type	`Authorization Code`
Request access token locally	`Cleared`

   Postman launches a popup with the login form from your Okta org.
4. Sign in as **martin.white**.

   Postman closes the popup after a successful authentication and displays the APIAM_TOKEN under Existing Tokens.
5. Click **APIAM_TOKEN**.
6. Postman presents the access_token and the id_token in the right pane.
7. Copy the access_token and paste it in https://www.jsonwebtoken.io.

Under the payload, confirm that the scope **promos** and the claim **department** with Sales value are presented.


```
{
 "ver": 1,
 "jti": "AT.jqQmDLbPZFW7rjRaJ0W1iVpv3e0DZeytBgD3qnuYDds",
 "iss": "https://oktaice.oktapreview.com/oauth2/aus9w0z4988od3YdS0h7",
 "aud": "https://api.oktaice.com",
 "iat": 1490136227,
 "exp": 1490139903,
 "cid": "fzyV0kSHGAQ8k2hy2I8Y",
 "uid": "00u9x97ogs84yyY010h7",
 "scp": ["openid", "profile", "promos"],
 "sub": "martin.white@oktaice.com",
 "department": "sales"
}
```


This confirms that the API AM Resource Server set the custom scopes and claims successfully.

8. **Optionally**, set the access_token as Header and click **Send**. The userinfo request will return information about Martin White.

Test the Client Credentials Flow

1. Under the Get User Info collection, click **Get New Access Token**.
2. Enter the information as follows and then click **Request Token**.

Attribute	Value
Token Name	APIAM_CLIENTCREDS_TOKEN
Auth URL	{{url}}/oauth2/{{authorizationServerId}}/v1/authorize
Access Token URL	{{url}}/oauth2/{{authorizationServerId}}/v1/token
Client ID	{{serviceClientId}}
Client Secret	{{serviceClientSecret}}
Scope	promos
Grant Type	Client Credentials
Request access token locally	Deselected

Postman closes the popup after a successful authentication and displays the APIAM_CLIENTCREDS_TOKEN under Existing Tokens.

3. Click **APIAM_CLIENTCREDS_TOKEN**.
   Postman presents the access_token in the right pane. The openid token is not displayed because it's was not included in the OAuth token request.
4. Copy the access_token and paste it in https://www.jsonwebtoken.io.
   Under the payload, confirm that the scope **promo** is set, and that the sub value is not a user id.


```
{
 "ver": 1,
 "jti": "AT.8e23w88OvY_u3Hyf4lxBb41UtzoQZsm8fJGLodLjffs",
 "iss":
"https://frederico.oktapreview.com/oauth2/aus9w0z4988od3YdS0h7",
 "aud": "https://api.oktaice.com",
 "iat": 1490137006,
 "exp": 1490140606,
 "cid": "95qBlAgLoqZOoyLw2XCf",
 "scp": ["promos"],
 "sub": "95qBlAgLoqZOoyLw2XCf"
}
```


This confirms that the API AM Resource Server can issue tokens for applications via client credentials flow.

Note: The value in the **sub** claim is the `Service App` client id. This happens because the client credentials authentication does not involve an end-user.

Lab 10-6: Enable the Development Team

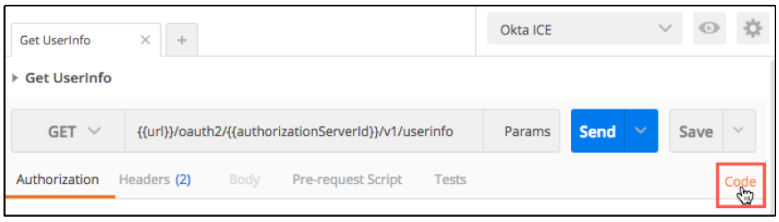
Objective	In this lab, you access resources that will help developers implementing applications that leverage the Okta API AM.
Scenario	Now that you implemented and tested the configuration in Okta, its time to prepare the development team. For this, you decided to provide your Postman collections, along with few code snippets, samples, and documentation references.
Duration	10 minutes

Generate Code from Postman

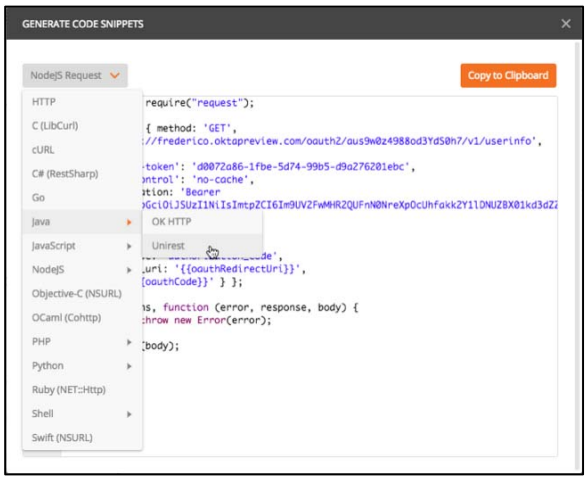
Tips:

- The Postman's Generate Code Snippets feature can create code snippets for the REST API calls you explored in popular programming languages, using values defined for your environment.
- The code snippets can be leveraged by developers to learn how to develop the REST API calls. System Administrators can also use snippets generated for CLI utilities like cURL and wget to test the REST APIs without using Postman.

1. Return to the Get UserInfo collection in Postman and then click **Code**.



2. Explore the options available in the Generate Code Snippets pop-up.

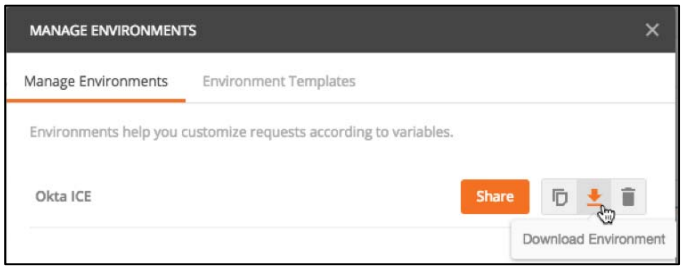


3. Close the Generate Code Snippets pop-up.

Export Postman Collections and Environments

Tip: Postman allows you to export environment variables and collections that can be used by developers as reference and to test the Okta API AM authentication.

1. To export the environment in Postman, click the **gear icon** (top-right corner) and then click **Manage Environments**.
2. Click the **Download Environment** (arrow icon) next to Okta ICE:



3. Save the environment file in a folder of your preference.

   Tip: This file contains the keys and values you set for the Okta ICE environment.
4. Close the Manage Environments pop-up.
5. To export the collection in Postman, expand the Collection menu for **OAuth 2.0 (Okta API)**, and then click **Export**.
6. Save the collection file in a folder of your preference.

   Tip: Although Okta provides a public collection for the APIs in developers.okta.com, you can customize your collection – by removing or adding additional requests – and export it in Postman.

Bookmark Assets for Developers

Tip: Okta provides important assets that you can use to guide your developers to implement OAuth and OIDC applications.

1. Launch your browser.
2. Access and bookmark the following pages:

Page	Description
www.jsonwebtoken.io	Web utility for decoding JWT tokens.
www.jwtinspector.io	Google Chrome plugin for decoding JWT tokens. You can use this option in case you don't want to past your JWT token in an external website.
www.oauth.com	Friendly documentation about the

	OAuth standard.
developer.okta.com/documentation	Documentation index for Okta's developers.
www.github.com/okta	Okta's official page on GitHub, where you developers can find code samples and other resources.

Explore a Code Sample

Tips:
Okta provides complete code samples in the most popular programming languages. These samples can be used by your developers as a sample implementation or as a boilerplate to develop applications integrated with Okta.
The Okta GitHub page is constantly updated with new samples in different programming languages. Also, each sample is constantly updated to incorporate new features provided by Okta SSO and API AM.

1. Access Okta's official page on GitHub (http://www.github.com/okta.)
2. Under Repositories, search for `samples`.
3. Click **samples-python-flask**.
4. Scroll down and explore the README.md contents.
   The `README.md` file provides general information about the sample and how you can deploy and configure the same in your environment and Okta org.
5. Optionally, explore the other files under the repository.

   Notes:
   - The source code is spread throughout the repository folders and files per the sample architecture and programming languages.
   - The repository contains additional files that document the application. As an example, the LICENSE file contains the sample code licensing (most of the times, Apache 2.0.)