HANDS-ON ACTIVITIES

IDENTITY & ACCESS MANAGEMENT

FEBRUARY, 2018

1. AN INTRODUCTION TO OAUTH FLOWS

Incorporate the right SSO components in an overall solution and set of flows.

USE CASE

Universal Containers has set up a photo-sharing web site as the resource server and a printing service which, as a client application, has both a desktop client as well as a web-based application client through a third-party hosted web site. Access between the client applications and the resource server should be seamless. To ensure security is adhered to, Universal Containers also wants to provide read-only access to some subset of photos for only a limited amount of time, after which the authorization becomes invalid. In addition, a customer has logged a support ticket that he has lost his mobile device and is concerned that any stranger who picks up the device can access the application, as the session is valid. Universal Containers administrators have been tasked to resolve this issue for the customer.

For the purpose of this exercise, we will assume that the resource server is hosted on Salesforce and the client requests are initiated from the Axiom OAuth tester.

DETAILED REQUIREMENTS

1. Create and configure an OAuth Web Server Flow.

2. Configure the OAuth Scope settings to set the timeout value.

3. Create and configure an OAuth User Agent Flow.

4. Revoke Access to the OAuth Access Token.

PREREQUISITE SETUP STEPS

1. Create a user record in the Salesforce org to be used for the user login for the exercise.

CONSIDERATIONS

1. What needs to be configured on the Salesforce org in order for the external client application to integrate with Salesforce APIs? Is this configuration for outbound/inbound/bi-directional scenario?

2. How do we ensure access is only given for a limited period of time for the application?

3. How do we allow Universal Containers to revoke access if they decide they no longer wish the client to have access?

4. Do we need to pass and store user credentials on the client application to achieve the seamless access requirement?

5. Can we use SAML only for single-sign-on scenarios for the desktop client application in Salesforce? If not, why?

CONSIDERATIONS SOLUTIONS:

1. What needs to be configured on the Salesforce org in order for the external client application to integrate with Salesforce APIs? Is this configuration for outbound/inbound/bi-directional scenario?
    a. A connected app needs to be configured in the Salesforce org.
    b. A connected app is only for an inbound scenario from an external application into Salesforce.

2. How do we ensure access is only given for a limited period of time for the application?
    a. Configure the Refresh Token Policy setting under the OAuth Policy for the Connected App.

3. How do we allow Universal Containers to revoke access if they decide they no longer wish the client to have access?
    a. Use the **Revoke** action on the user row in the Connected Apps Usage page to revoke access to the valid token. Any third party with the device will now need to authenticate again in order to get access with a new access/refresh token.

4. Do we need to pass and store user credentials on the client application to achieve the seamless access requirement?
    a. No. Both the OAuth Web Server Flow and User Server Flow provide access to the application through the use and exchange of tokens. No user credentials are stored on the client application.

5. Can we use SAML only for single-sign-on scenarios for the desktop client application in Salesforce? If not, why?
    a. No. SAML is a browser-based protocol and requires a browser to facilitate the exchange of SAML tokens in order for the authentication to take place.

For OAuth-based desktop or mobile applications, the authentication function is separated from the authorization function. SAML is used for the authentication function, which is achieved by the use of embedded browsers for desktop or mobile applications, and the authorization function is fulfilled using the OAuth protocol.

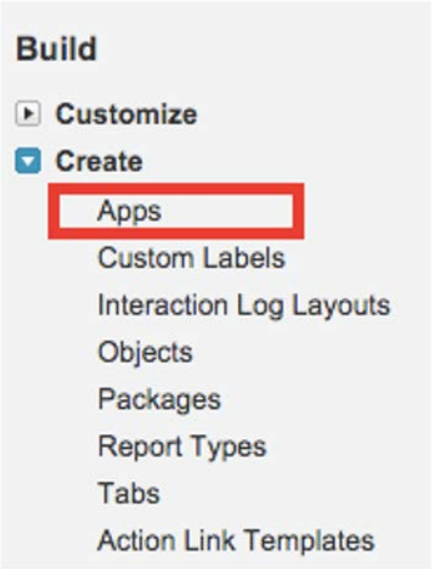
When layered together with SAML, the OAuth protocol is simply treated like any other bookmark or deep-link. In short, there is no additional development or deployment required to enable single-sign-on for desktop and mobile apps.

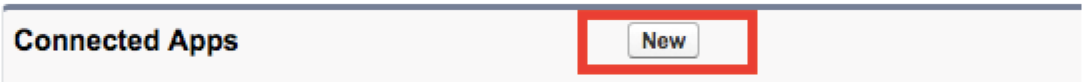
The SAML protocol in this case does not do the authorization function required for the scenario; hence it cannot be used alone to achieve single-sign-on flow for desktop/mobile applications.

SOLUTION DESCRIPTION

1. CREATE AND CONFIGURE AN OAUTH WEB SERVER FLOW.

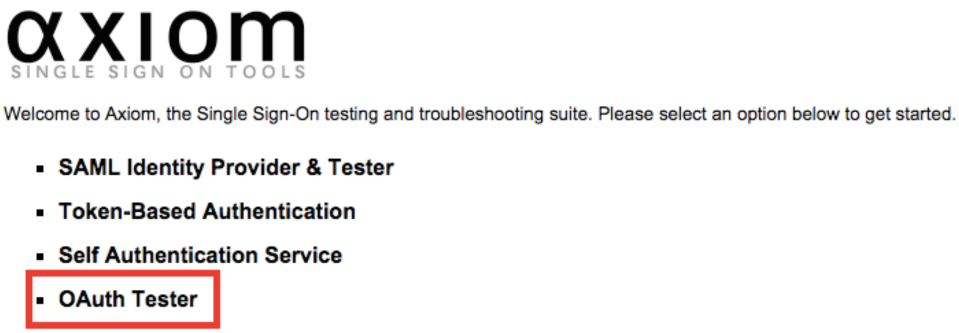
a. In your Salesforce org, create a Connected App in: **Setup | Create | Apps**.





b. Under the Connected Apps Section, click the **New** button and fill in the required fields for the Connected App (Connected App Name, API Name, Contact Email). We will return to this screen in Step e.

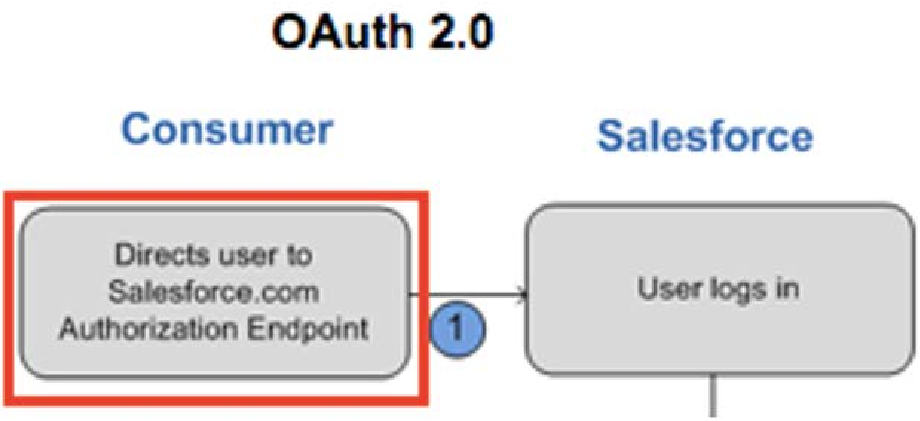
c. In a separate browser, go to http://axiomsso.herokuapp.com and click the **OAuth Tester**.


axiom
SINGLE SIGN ON TOOLS

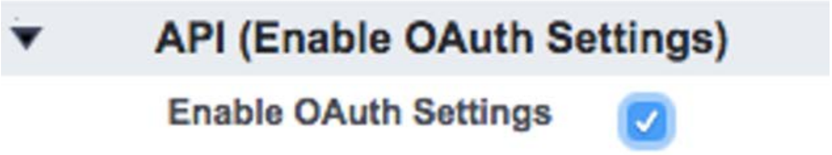
Welcome to Axiom, the Single Sign-On testing and troubleshooting suite. Please select an option below to get started.

SAML Identity Provider & Tester
Token-Based Authentication
Self Authentication Service
OAuth Tester

d. Click OAuth 2.0, Step 1: Directs user to Saleforce.com Authorization Endpoint.



e. Return to the browser screen in Step c and select **Enable OAuth Settings** under the section API (Enable OAuth Settings). Once this setting is checked, more fields will be made available for configuration.



▼ API (Enable OAuth Settings)

Enable OAuth Settings ☑

f. Set the callback URL to the Axiom-provided Redirect URI: https://axiomsso.herokuapp.com/OAuth2HandleAuthCode.action

g. In the Selected OAuth Scopes setting, add **Access and manage your data (api)** and **Perform requests on your behalf at any time (refresh_token, offline_access)**.

h. Leave the other settings as default and save the Connected App setting by clicking the **Save** button.

i. Click **Continue** on the next screen, which has the message "Allow from 2-10 minutes for your changes to take effect on the server before using the connected app."

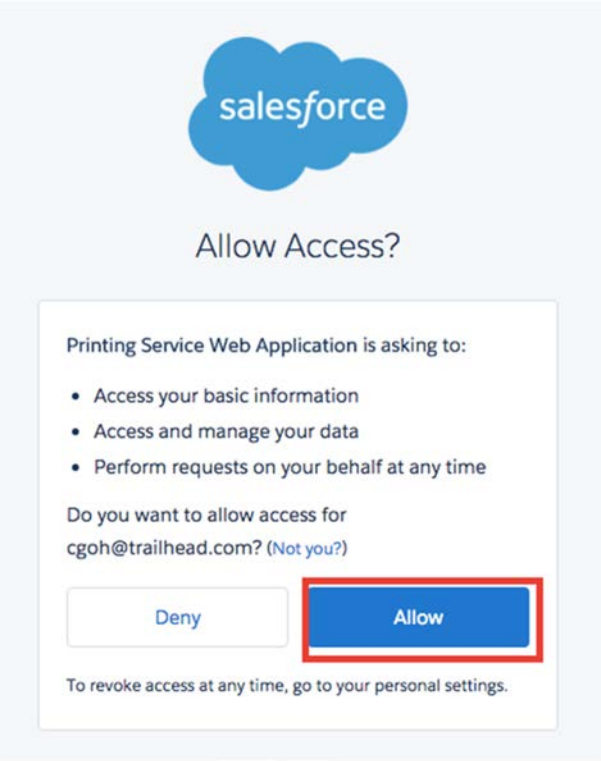
j. Your connected app is now created. In the same screen, copy the Consumer Key under the API (Enable OAuth Settings) section.

k. Navigate to the browser screen in Step d (Axiom) and paste in the Consumer Key copied from Salesforce. Click the button Request Authorization Code.

l. You should be redirected to login and authorize or just authorize if you are already logged in with a session. Use the credentials of the user you created for this build exercise if you have not logged in.

NOTE:

If this is not working, wait 2-10 minutes and try again. This is due to some delay in the connected app setup/propagation.



Allow Access?

Printing Service Web Application is asking to:

- Access your basic information
- Access and manage your data
- Perform requests on your behalf at any time

Do you want to allow access for cgoh@trailhead.com? (Not you?)

To revoke access at any time, go to your personal settings.

An example of the authorization screen is as follows. Click Allow to allow the authorization.

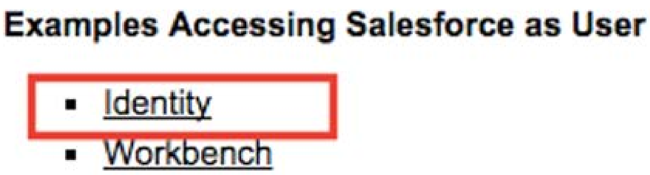
m. The Axiom application now displays the Authorization Code which you can use to request an Access Token.

n. Navigate to the browser screen in Step j (Salesforce). Under the API (Enable OAuth Settings) section, click the **Click to reveal** link beside the Consumer Secret field to reveal the Consumer Secret number combination for the Connected App and copy the Consumer Secret.

o. Paste in the Consumer Secret from the Connected App in the Consumer Secret: field back in the Axiom screen and click the **Request Access Token** button.

p. You are now granted an Access Token and you have a valid session.

q. Verify your session by clicking the **Identity** URL. This invokes the RESTful identity service and returns a XML response showing your identity details.



Examples Accessing Salesforce as User

- Identity
- Workbench

r. Go to your user record: **Setup | Users** and check the Login History section. There should be a successful login entry with the Login Type as Remote Access 2.0 and application with the name of your Connected App.

s. Also, check Connected Apps Usage: **Setup | Connected Apps OAuth Usage**. The User Count for your configured Connected App should be 1.

2. CONFIGURE THE OAUTH SCOPE SETTINGS TO SET THE TIMEOUT VALUE.

a. Navigate to the Connected Apps Usage page in Salesforce: **Setup | Connected Apps OAuth Usage**.

b. Click **View App Info** for your configured Connected App.

c. Click the **Edit** button.

d. Review the settings under OAuth Policies and Session Policies. These two settings are key to control the behavior of the length of access for the application.

e. Set the Timeout Value under Session Policies. This value sets the expiration of the access tokens for the connected app's session.

f. Set the Refresh Token Policy under OAuth policies. Refresh Token Policy specifies the validity period for a refresh token. Refresh tokens are used by the OAuth-enabled connected app to obtain new sessions without requiring the user to provide their credentials. The connected app simply exchanges the refresh token for a new session. Using refresh token policies, administrators control how long a refresh token is used.

g. Click the **Save** button to commit the configuration changes or the **Cancel** button to discard the changes.

3. CREATE AND CONFIGURE A OAUTH USER AGENT FLOW.

a. Construct an OAuth Authorize URL in the format below to simulate the User Agent flow and test by simply pasting into your browser. When successful, you should be asked to login, authorize, and should then see the session Id in a URL fragment.

   https://login.salesforce.com/services/oauth2/authorize?response_type=token&client_id=CLIENT_ID&redirect_uri=CALLBACK_URL

   The client_id will be the consumer key of your configured Connected App and redirect_uri will be the Callback URL setting defined in the configured Connected App.

NOTE:

For the CALLBACK_URL, use the axiom callback URL with URL encoding.

For example:

https%3A%2F%2Faxiomsso.herokuapp.com%2FOAuth2HandleAuthCode.action

An example of a sample URL will look like the below. Note that for your instance, the client_id will be different.:

https://login.salesforce.com/services/oauth2/authorize?response_type=token&client_id=3MVG9ZL0ppGP5UrD55QeA7GU.b23GP7BRAU8KTZgwKxpV43VM2S5Eids7CTZp2Yr1_3fS4W9hiVVUp1XjGBtS&redirect_uri=https%3A%2F%2Faxiomsso.herokuapp.com%2FOAuth2HandleAuthCode.action

b. If an authorization screen pops up, click **Allow** and the Axiom page similar to the image below will be displayed.

c. Ignore the page, but inspect the URL in the browser (cut and paste the URL into notepad). The URL has a hash fragment with the access token, which is the session ID. You can use this access token to access your Salesforce data.

d. Go to your user record: **Setup | Users** and check the Login History section. There should be a successful login entry with the Login Type as Remote Access 2.0 and application with the name of your Connected App.

e. Also, check Connected Apps Usage: **Setup | Connected Apps OAuth Usage**. There should be a user count increase that verifies that the connected app has initiated a successful login into the Salesforce org.

4. REVOKE ACCESS TO THE OAUTH ACCESS TOKEN.

a. Navigate to the Connected Apps Usage page in Salesforce: **Setup | Connected Apps OAuth Usage**.

b. Click the **User Count** link for your configured Connected App.

c. Click the **Revoke** link under the Action column for the user record, which you want to revoke access.

d. Revoking access would mean the user would need to request for a new Access Token in order to access the application.

2. HUB AND SPOKE SSO FOR MULIT-SALESFORCE ORGS

Articulate the Various SSO Concepts And Components.

USE CASE

Universal Containers is a leading logistics supplier in the industry with presence in both America and Europe, with HQ based in America. Currently, the organization uses two separate Salesforce orgs for the America and Europe business. User credentials currently are not replicated across the two orgs. To facilitate collaboration between the two regions, Universal Containers has started on a project initiative to provide seamless access across the two orgs. As part of this build exercise, you can assume that user credentials can be replicated across one org, so as to provide a central repository for credentials.

DETAILED REQUIREMENTS

1. Configure a My Domain for each org.
2. Deploy each My Domain to your users.
3. Enable the Identity Provider in your Identity Provider org.
4. Download the Self-Signed Certificate from the IDP org.
5. Configure SAML in your first Service Provider org.
6. Tell your Identity Provider about this Service Provider org.
7. Test your configuration.

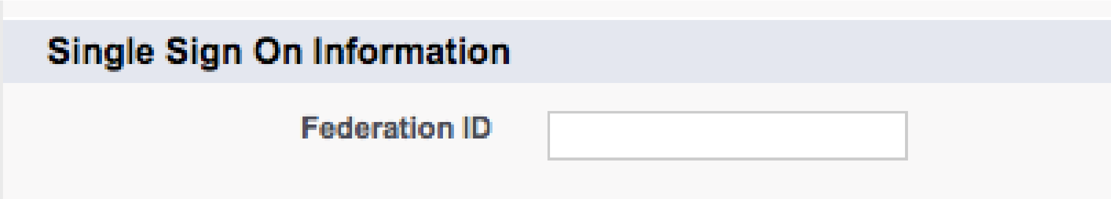
PREREQUISITE SETUP STEPS

1. Subscribe to 2 developer orgs at https://developer.salesforce.com/signup if you do not have an available Salesforce org that can be used for the purpose of this build exercise.

2. Determine which org will play the role of the Identity Provider ("IdP"). This will be the Salesforce org used to authenticate the user and log him/her into the Service Providers ("SP"). The other org will be used as the Service Provider.

3. Create a master list of all users and create the corresponding users in both the Identity Provider org and the Service Provider Org. For example, there should be both a Bob Jones user in the IdP org and the SP org.

4. Provide each user with a Federation ID (unique identifier for the user across all orgs). The Federation ID will be used to link the user record between the user record in the SP org to the user record in the IdP org. For Example: A "Bob Jones" user in IdP org will have the same Federation ID as the corresponding "Bob Jones" user in the SP Org. In short, each user will have a unique username, but a common Federation ID. The Federation ID can be edited in the user record and can be any string value. A common value to use to ensure it is unique between the orgs will be to use the email address of the user.



Single Sign On Information

Federation ID	

CONSIDERATIONS

1. Why is the My Domain configuration required for an SP Initiated SAML Flow?

2. How do we simulate an IdP initiated flow in our configuration setup?

3. How do we simulate an SP initiated flow in our configuration setup?

4. What is the role of the browser in the SAML flow?

5. How do we ensure that the browser knows how to redirect to the original requested resource?

CONSIDERATIONS SOLUTIONS:

1. Why is the My Domain configuration required for a SP Initiated SAML Flow?
    a. By configuring a My Domain, admins enable the Force.com platform to recognize the org for which unauthenticated requests are intended, and perform customized behavior, such as redirecting to their Identity Provider through the retrieval of the Single Sign-On settings configuration of the Service Provider org. This would not have been possible if the login was done through a URL at login.salesforce.com as the application will only know which org you are trying to log in to only after the authentication happens. This is because login.salesforce.com is a generic login URL for all Salesforce orgs.

2. How do we simulate an IdP-initiated flow in our configuration setup?
    a. From the IdP Salesforce org, click on a link to a resource that is hosted in the SP Provider Salesforce org. The user should not be prompted for any login to the SP Provider org.

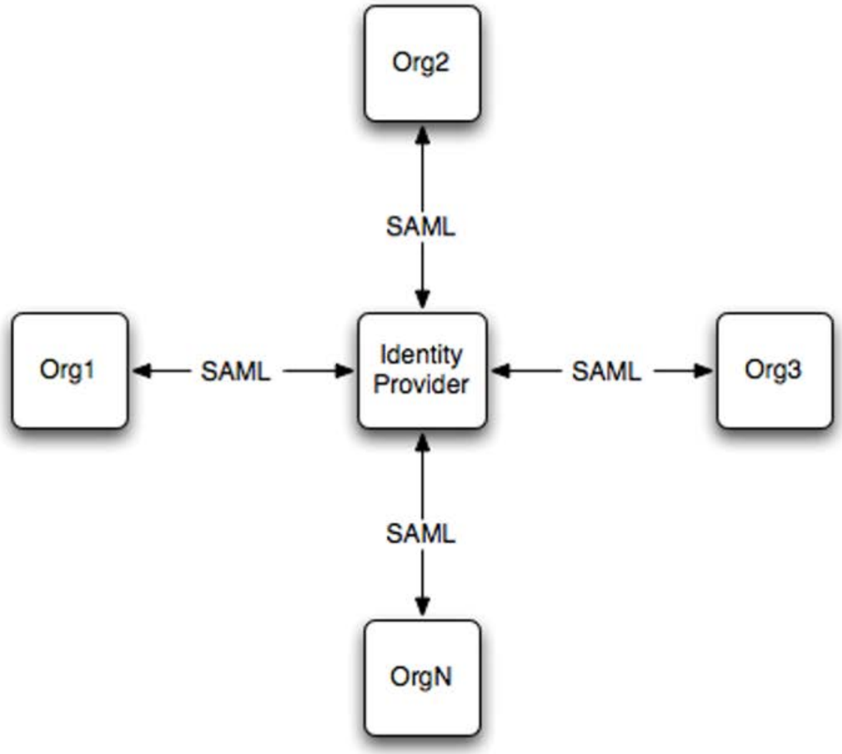
3. How do we simulate an SP initiated flow in our configuration setup?
    a. Access the SP Salesforce org through its MyDomain URL. If there is no valid authentication session, the browser will redirect the user to the IdP to enter their credentials for authentication.

4. What is the role of the browser in the flow?
    a. The browser is used to facilitate the exchange of SAML tokens between the SP and the IdP.

5. How do we ensure that the browser knows how to redirect to the original requested resource?
    a. This is achieved through the use of the RelayState parameter, which is passed via the browser as part of the SAML token exchange.

SOLUTION DESCRIPTION

The approach for implementing Single Sign-On across multiple orgs can most easily be conceptualized as a "hub-and-spoke" model.

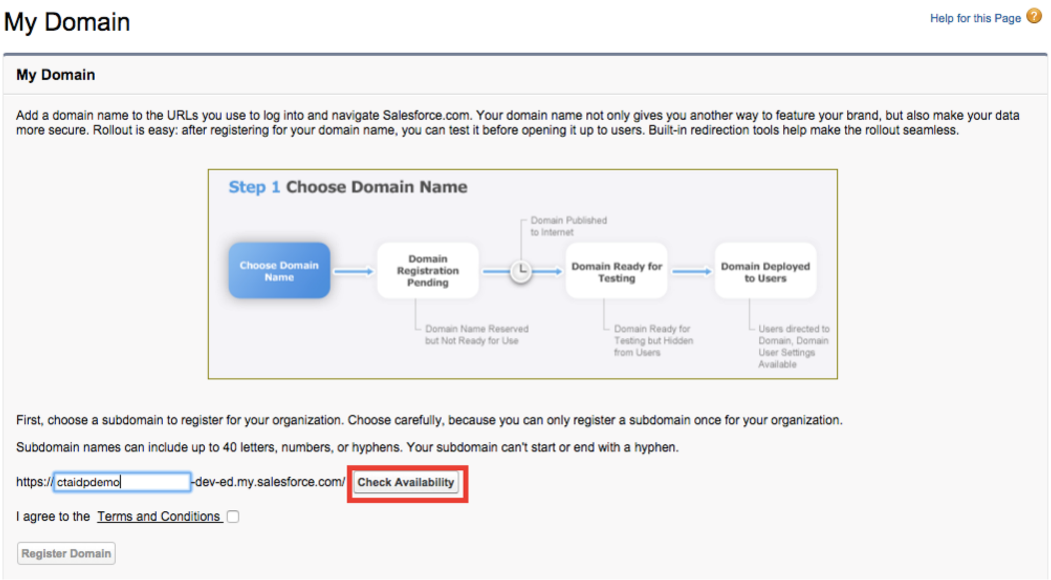


In the center of this architecture, we have an Identity Provider, or IdP. This plays the role of the centralized authentication "hub," and is responsible for validating a user's credentials and asserting the user's Identity to the "spokes," the orgs.

Using this technique, the user may easily move from org to org; as authentication in a new org is required, the process of redirecting to the Identity Provider is repeated. Since the Identity Provider can maintain a centralized session for the user, each time this occurs, it may not be necessary for the user to re-authenticate. The result is a seamless single sign-on experience for the user, resulting in faster and simpler access to all their resources.

1. CONFIGURE A MY DOMAIN FOR EACH ORG.

a. Log in as an Administrator to each of your orgs, and browse to Setup | Domain Management | My Domain.Choose a custom host name. Click on the button Check Availability.
b. If the hostname is not available, choose another host name.



My Domain

Help for this Page

My Domain

Add a domain name to the URLs you use to log into and navigate Salesforce.com. Your domain name not only gives you another way to feature your brand, but also make your data more secure. Rollout is easy: after registering for your domain name, you can test it before opening it up to users. Built-in redirection tools help make the rollout seamless.

Step 1 Choose Domain Name
Choose Domain Name
Domain Registration Pending
Domain Published to Internet
Domain Ready for Testing
Domain Deployed to Users
Domain Name Reserved but Not Ready for Use
Domain Ready for Testing but Hidden from Users
Users directed to Domain, Domain User Settings Available

First, choose a subdomain to register for your organization. Choose carefully, because you can only register a subdomain once for your organization.

Subdomain names can include up to 40 letters, numbers, or hyphens. Your subdomain can't start or end with a hyphen.

https:// ctaidpdemo -dev-ed.my.salesforce.com/ Check Availability

I agree to the Terms and Conditions □

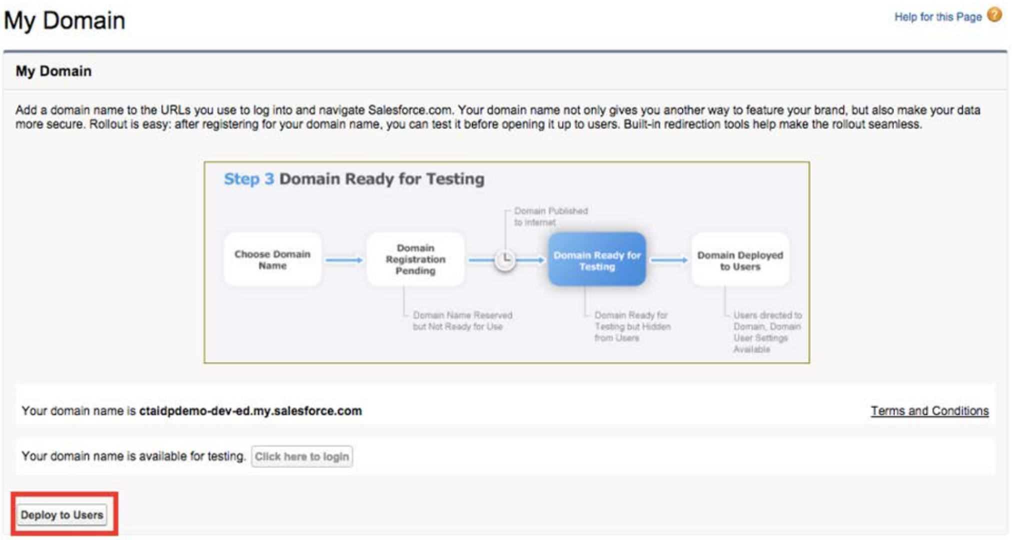
Register Domain

c. Check the box next to I agree to the Terms and Conditions and click the button Register Domain.
d. While salesforce.com is working to reduce latency of this process, it may take up to 24 hours to complete. You'll receive an email when your change is complete.
e. Repeat steps b through d for the other org.

2. DEPLOY EACH MY DOMAIN TO YOUR USERS.

a. Log in as an Administrator to each of your orgs via the MyDomain URL as received in the email from step d above, and browse to **Setup | Domain Management | My Domain**.

b. Click the **Deploy To Users** button to deploy the domain to all users



My Domain

My Domain

Add a domain name to the URLs you use to log into and navigate Salesforce.com. Your domain name not only gives you another way to feature your brand, but also make your data more secure. Rollout is easy: after registering for your domain name, you can test it before opening it up to users. Built-in redirection tools help make the rollout seamless.

Step 3 Domain Ready for Testing

Choose Domain Name
Domain Registration Pending
Domain Name Reserved but Not Ready for Use
Domain Published to Internet
Domain Ready for Testing
Domain Ready for Testing but Hidden from Users
Domain Deployed to Users
Users directed to Domain, Domain User Settings Available

Your domain name is **ctaidpdemo-dev-ed.my.salesforce.com**

Your domain name is available for testing.

c. Repeat step b for the other org.

3. ENABLE THE IDENTITY PROVIDER IN YOUR IDENTITY PROVIDER ORG.

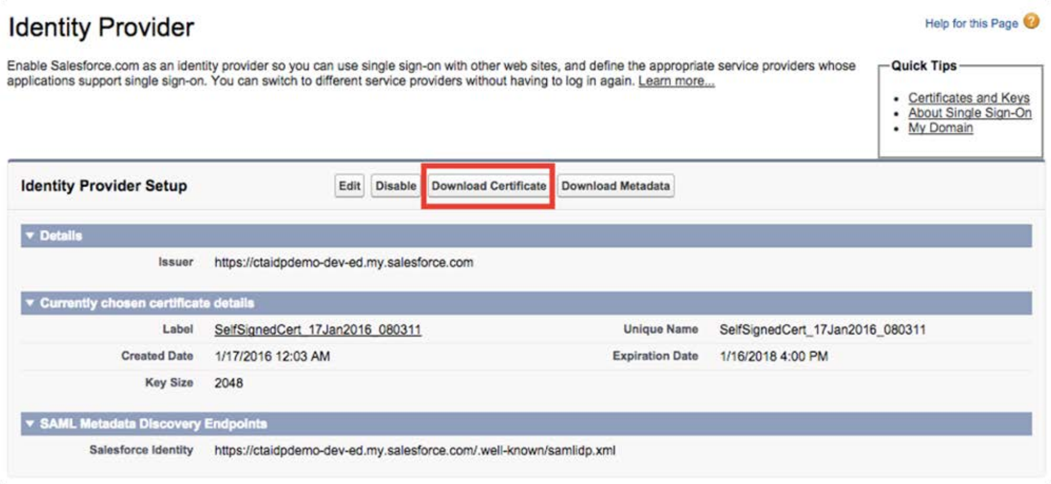
a. Log in as an Administrator to the org identified as the IdP org to be used for this exercise and browse to Setup | Security Controls | Identity Provider.

b. Click the Enable Identity Provider button if the org has not been enabled as an Identity Provider.

c. Make note of the Issuer field, which is the MyDomain URL for this Org to be used as an IdP for the build exercise. We will use this field to configure the Single Sign-On Settings in the SP org.

4. DOWNLOAD THE SELF-SIGNED CERTIFICATE FROM THE IDP ORG.

a. Click the **Download Certificate** button in the Identity Provider setup screen to download the certificate file to your desktop.



Identity Provider

Enable Salesforce.com as an identity provider so you can use single sign-on with other web sites, and define the appropriate service providers whose applications support single sign-on. You can switch to different service providers without having to log in again. Learn more...

Quick Tips

- Certificates and Keys
- About Single Sign-On
- My Domain

Identity Provider Setup

▼ Details

Issuer	https://ctaidpdemo-dev-ed.my.salesforce.com		

▼ Currently chosen certificate details

Label	SelfSignedCert_17Jan2016_080311	Unique Name	SelfSignedCert_17Jan2016_080311
Created Date	1/17/2016 12:03 AM	Expiration Date	1/16/2018 4:00 PM
Key Size	2048		

▼ SAML Metadata Discovery Endpoints

Salesforce Identity	https://ctaidpdemo-dev-ed.my.salesforce.com/.well-known/samlidp.xml

b. This certificate corresponds to a private key only available to your Identity Provider. You will need this file in subsequent steps, as it will be used by each of your Service Provider orgs to establish trust to the Identity Provider.

5. CONFIGURE SAML IN YOUR FIRST SERVICE PROVIDER ORG.

a. Log in as an Administrator to the first SP org. Browse to **Setup |Security Controls | Single Sign-On Settings**.

b. Click **Edit** and check the **SAML enabled** checkbox and click the **Save** button.

c. Click the **New** button under the SAML Single Sign-On Settings section.

d. Fill in the Name and API Name fields. This can be any string value.

e. Enter your Identity Provider's Issuer that you made note of in Step 3c into the Issuer field.

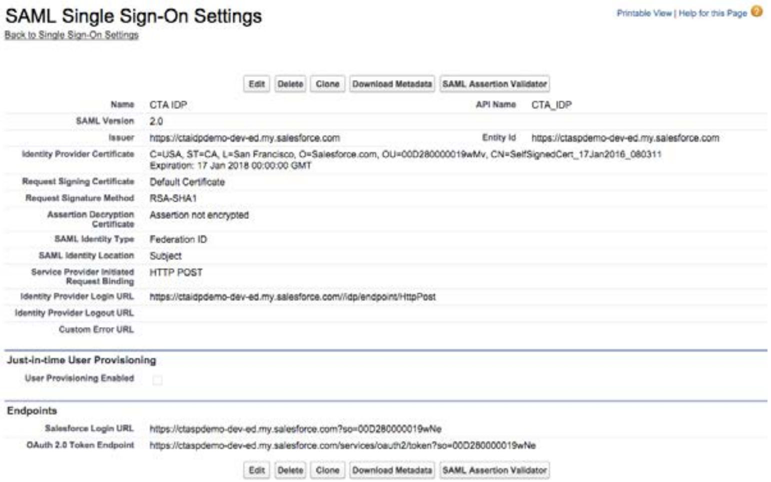
f. Click the **Choose File** button for the Identity Provider Certificate, and select the Identity Provider certificate that you downloaded to the desktop in Step 4a.

g. Change the **Entity Id** to the My Domain URL for this SP org.

h. Change SAML Identity Type to **Assertion contains Federation ID from the User object**.

i. Enter the My Domain URL of your Identity Provider + "/idp/endpoint/HttpPost" into the Identity Provider Login URL.

j. Click the **Save** button. If the save is successful, you should see a screen similar to the below image:


Printable View | Help for this Page
Back to Single Sign-On Settings
Edit
Delete
Clone
Download Metadata
SAML Assertion Validator

SAML Single Sign-On Settings

Name	CTA IDP	API Name	CTA_IDP
SAML Version	2.0		
Issuer	https://ctaidpdemo-dev-ed.my.salesforce.com	Entity Id	https://ctaspdemo-dev-ed.my.salesforce.com
Identity Provider Certificate	C=USA, ST=CA, L=San Francisco, O=Salesforce.com, OU=00D280000019wMv, CN=SelfSignedCert_17Jan2016_080311 Expiration: 17 Jan 2018 00:00:00 GMT		
Request Signing Certificate	Default Certificate		
Request Signature Method	RSA-SHA1		
Assertion Decryption Certificate	Assertion not encrypted		
SAML Identity Type	Federation ID		
SAML Identity Location	Subject		
Service Provider Initiated Request Binding	HTTP POST		
Identity Provider Login URL	https://ctaidpdemo-dev-ed.my.salesforce.com//idp/endpoint/HttpPost		
Identity Provider Logout URL			
Custom Error URL			

Just-in-time User Provisioning

User Provisioning Enabled	☐

Endpoints

Salesforce Login URL	https://ctaspdemo-dev-ed.my.salesforce.com?so=00D280000019wNe
OAuth 2.0 Token Endpoint	https://ctaspdemo-dev-ed.my.salesforce.com/services/oauth2/token?so=00D280000019wNe

k. Take note of the field value for the Salesforce Login URL field.

l. Browse to **Setup | Domain Management | My Domain**.

m. Click the **Edit** button under the Authentication Configuration section.

n. Under the Authentication Service field, deselect **Login Page** and select the service that corresponds the name of the Single Sign-On settings configured in step j.

6. TELL YOUR IDENTITY PROVIDER ABOUT THIS SERVICE PROVIDER ORG.

a. Log in as an administrator to the org identified as the IdP org to be used for this exercise and browse to Setup | Security Controls | Identity Provider.

b. Click Service Providers are now created via Connected Apps. Click here, under the list of Service Providers.

c. Enter values for the Connected App Name, API Name, and Contact Email fields for your Connected App.

d. Select Enable SAML under Web App Settings.

e. Enter the My Domain URL of the SP Org as the Entity ID.

f. Enter the ACS URL. The value of this is the Salesforce.com Login URL you made note of in Step 5k.

g. Select Federation ID for Subject Type field.

h. Click the Save button. The Connected App details are displayed if the save is successful.

i. Click the Manage button.

j. Click the Manage Profiles button. Assign this SSO configuration to any profiles of your choosing. The test user's profile, which you are using to test this configuration, should be assigned here.

7. TEST YOUR CONFIGURATION.

a. Log out of both orgs.

b. Type the My Domain URL of the IdP org into your browser and log in as the test user.

c. Now, type the URL of the SP org into your browser.

d. You will immediately be redirected to the IdP org. Since you're already authenticated, you'll get redirected back to your SP org, and you should be logged in without being prompted for an additional login.

e. Check the Login History on the SP org: **Setup | Manage Users | Login History**. There should be a successful Login record of Login Type SAML-Initiated SSO.

3. SOCIAL SIGN-ON

Describe How to Incorporate Social Sign-On Into A Solution.

USE CASE

Universal Containers has recently embarked upon a social journey and, as part of this initiative, they would like to enable Social Sign On using Facebook for Customer Communities.

As part of a Customer Communities implementation, Universal Containers would like their customers to have ability to log in using Facebook.

As part of Customer Communities implementation, Universal Containers would like their customers to have ability to login using Facebook.

DETAILED REQUIREMENTS

- Ability for Customer Community user to login using Facebook.
- Ability to capture user data attributes from social profiles.
- Ability to link existing Customer Community user to their Facebook account.

PREREQUISITE SETUP STEPS

- Set Facebook as Authentication Provider in Salesforce.
- Create a Facebook Application on the Facebook Developer website.
- Modify the Registration Handler class to handle user provisioning and user data received from Facebook.
- (Optional) Create a Community and enable Facebook as login option – Community User.

- (Optional) Create My Domain and enable Facebook as login option – Internal UserCreation of “Community” & “My Domain” not covered in this exercise.

NOTE:

ℹ Creation of Community and My Domain not covered in this exercise

CONSIDERATIONS

- Ability for Customer Community users to use their Facebook credentials to log in to Customer Community.

- Are there exiting users who need to be enabled for social sign on?

- Which data attributes needs to be captured from the user social profile?

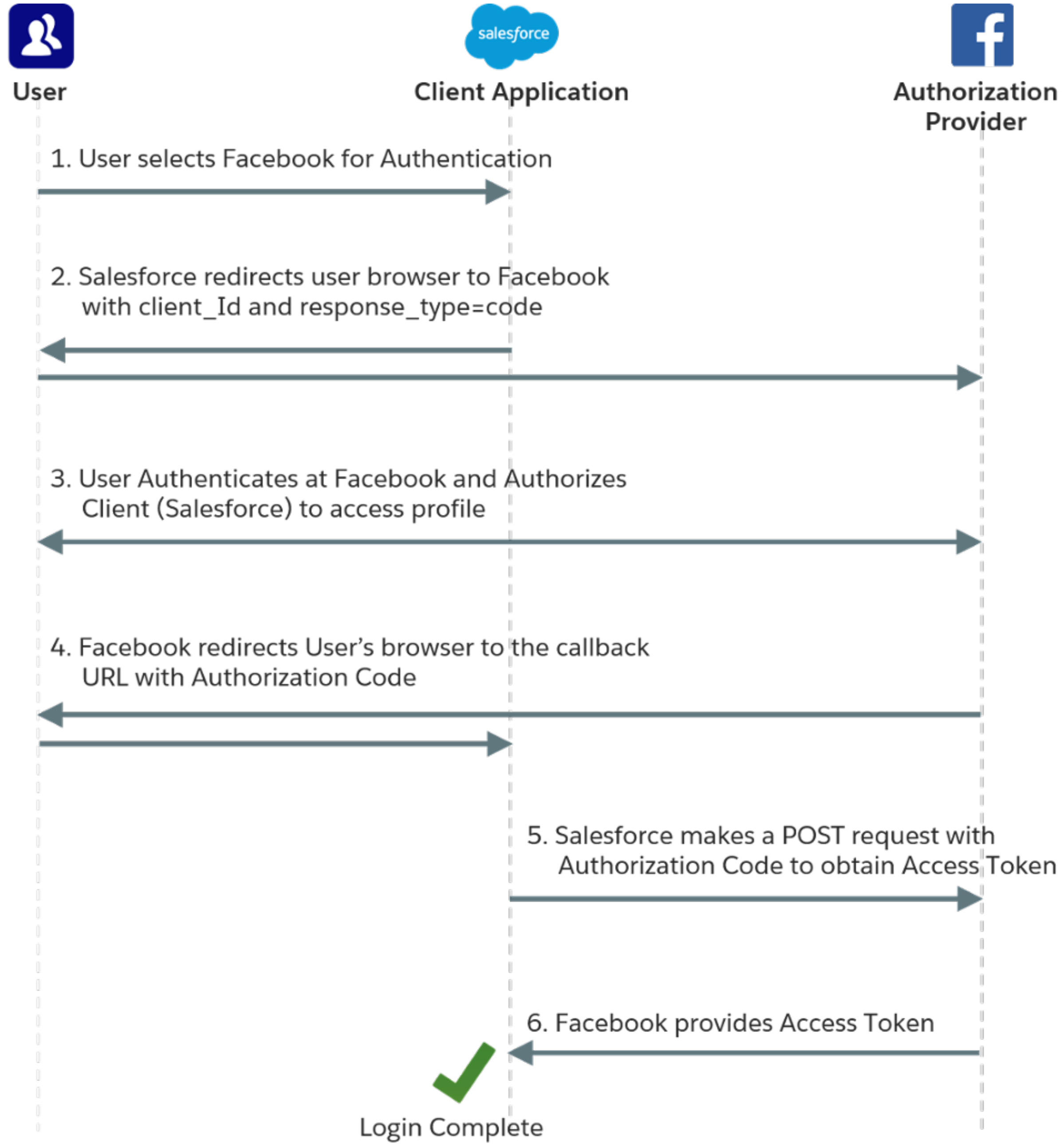
BEST SOLUTION OVERVIEW

Social Sign-On for Salesforce supports Facebook, Janrain, Salesforce, Open ID Connect, Microsoft Access Control Service, LinkedIn, Twitter, Google, and GitHub.

Facebook authentication can be set up for both Salesforce users and Community users. As part of this exercise, we will:

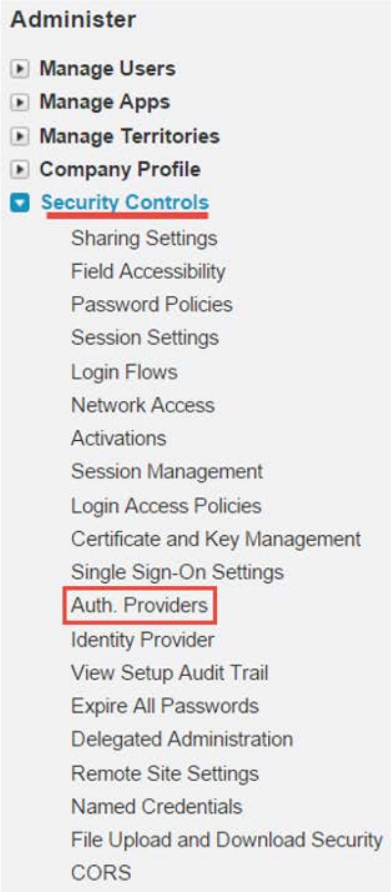
- Create a Facebook application that can be used to authenticate to Salesforce or a Salesforce Community.
- Setup Facebook as Auth. Provider in Salesforce.
- Modify the Auto-Generated Registration Handler class.
- Test the setup using the **Test-Only Initialization URL**.
- Link existing users via the **Existing User Linking URL**.
- Log in using the **Single Sign-On Initialization URL**.
  - Optionally, add a Facebook Login on Salesforce Login or Community Login page. (To add Facebook as a login option on the Salesforce login page, create My Domain. To add Facebook as login option on the Community login page, create a Community.

AUTHENTICATION FLOW


User
salesforce
Client Application
Authorization Provider
1. User selects Facebook for Authentication
2. Salesforce redirects user browser to Facebook with client_Id and response_type=code
3. User Authenticates at Facebook and Authorizes Client (Salesforce) to access profile
4. Facebook redirects User's browser to the callback URL with Authorization Code
5. Salesforce makes a POST request with Authorization Code to obtain Access Token
6. Facebook provides Access Token
Login Complete

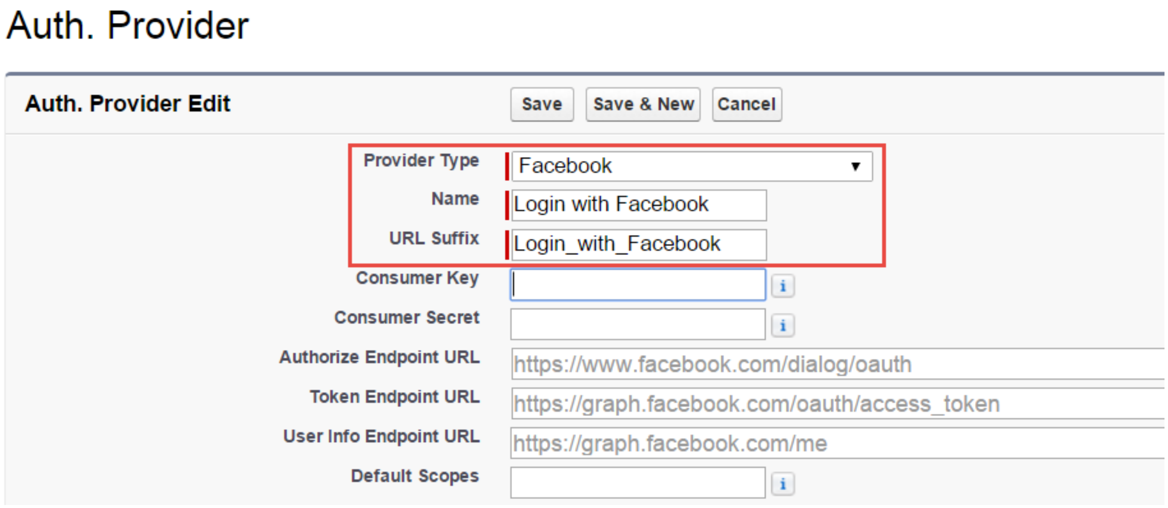
DETAILED SETUP INSTRUCTIONS:

1. Navigate to Setup | Administer |Security Controls | Auth. Providers and click the New button.



2. Pick Facebook from the Provider Type picklist. **Facebook**

3. Enter the Name and URL Suffix in the Auth. Provider Edit screen.

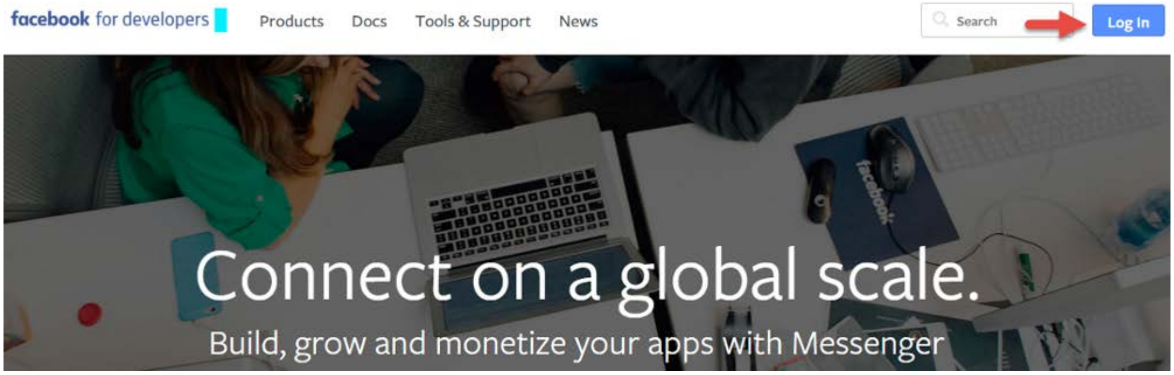


Auth. Provider

Auth. Provider Edit

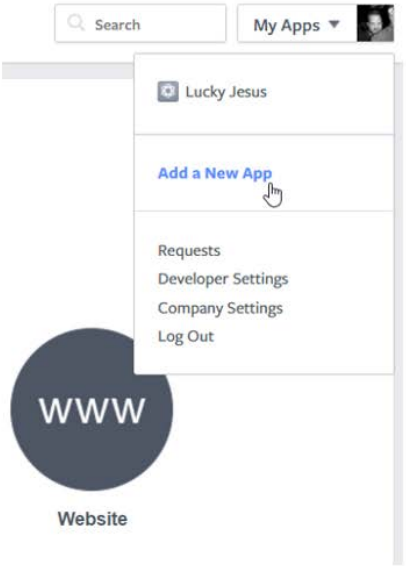
Provider Type	Facebook
Name	Login with Facebook
URL Suffix	Login_with_Facebook
Consumer Key	
Consumer Secret	
Authorize Endpoint URL	https://www.facebook.com/dialog/oauth
Token Endpoint URL	https://graph.facebook.com/oauth/access_token
User Info Endpoint URL	https://graph.facebook.com/me
Default Scopes	

4. In another browser, go to http://developers.facebook.com and log in as a Facebook user.

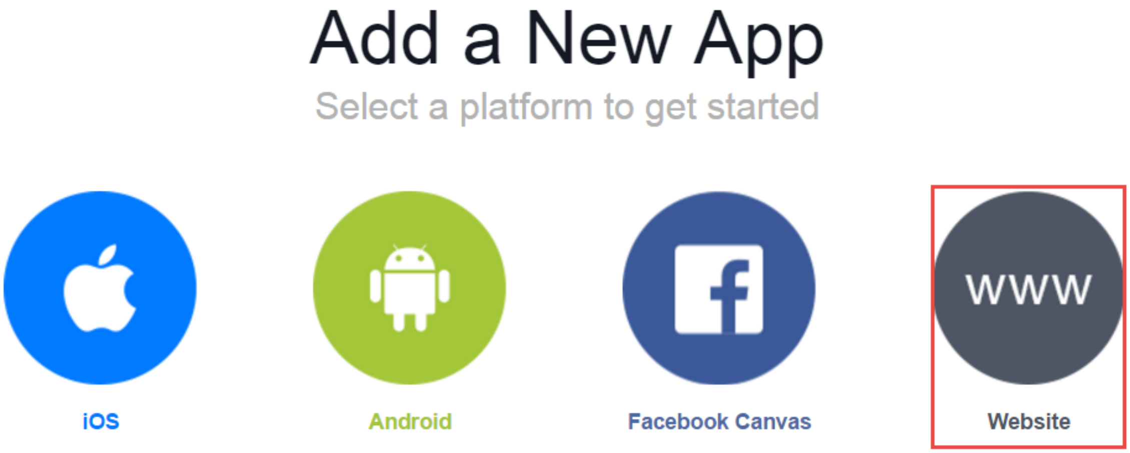


facebook for developers
Products
Docs
Tools & Support
News
Search
Log In
Connect on a global scale.
Build, grow and monetize your apps with Messenger

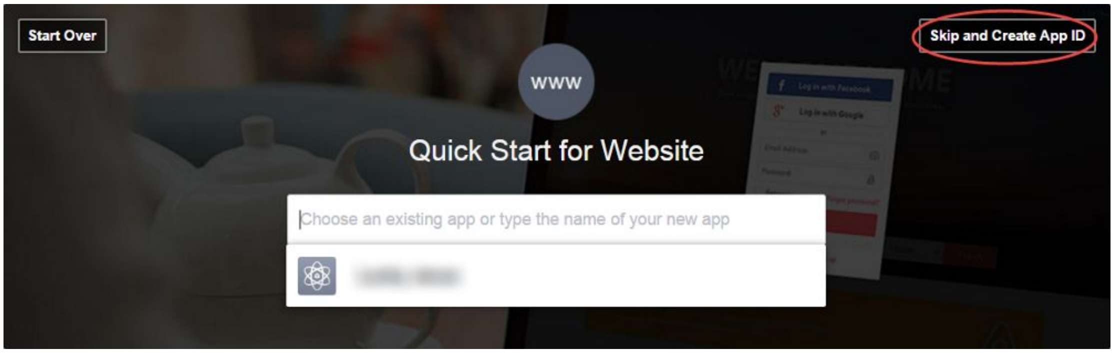
5. If you have not already registered, click the **Register** button. Click the toggle to read **Yes** on the Facebook Privacy Policy. On this next screen, click **My Apps** and select **Add a New App**.



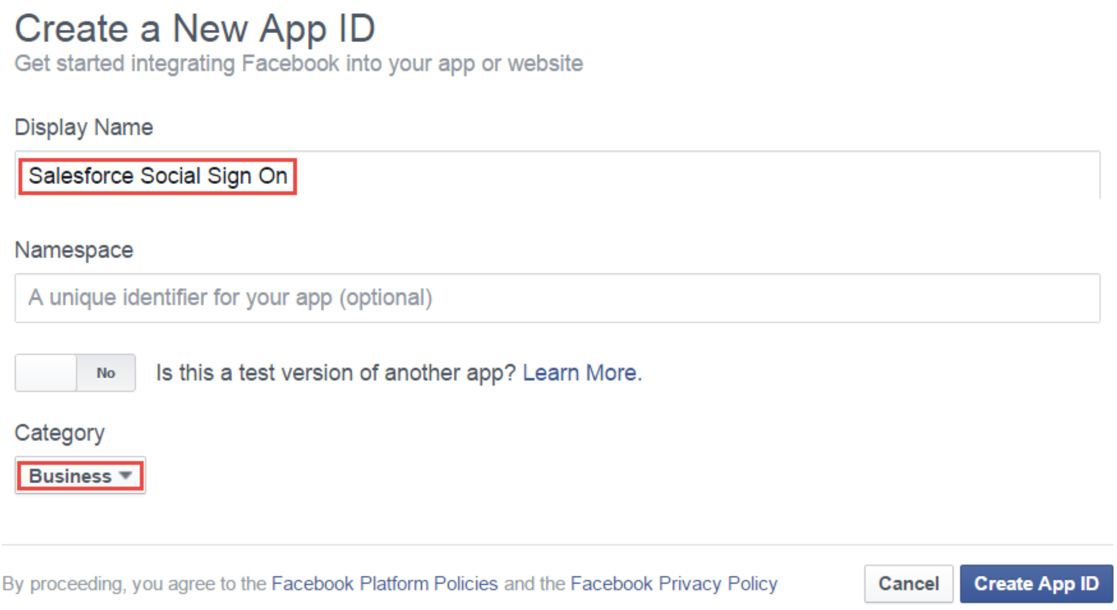
Next, select **Website** as the Platform.



6. If you have created Facebook apps previously, Facebook shows you your most recent app. Click **Skip and Create App ID**.



7. Enter the App name in the Display Name text box on the Create a New App ID dialog.



Create a New App ID

Get started integrating Facebook into your app or website

Display Name

Salesforce Social Sign On

Namespace

A unique identifier for your app (optional)

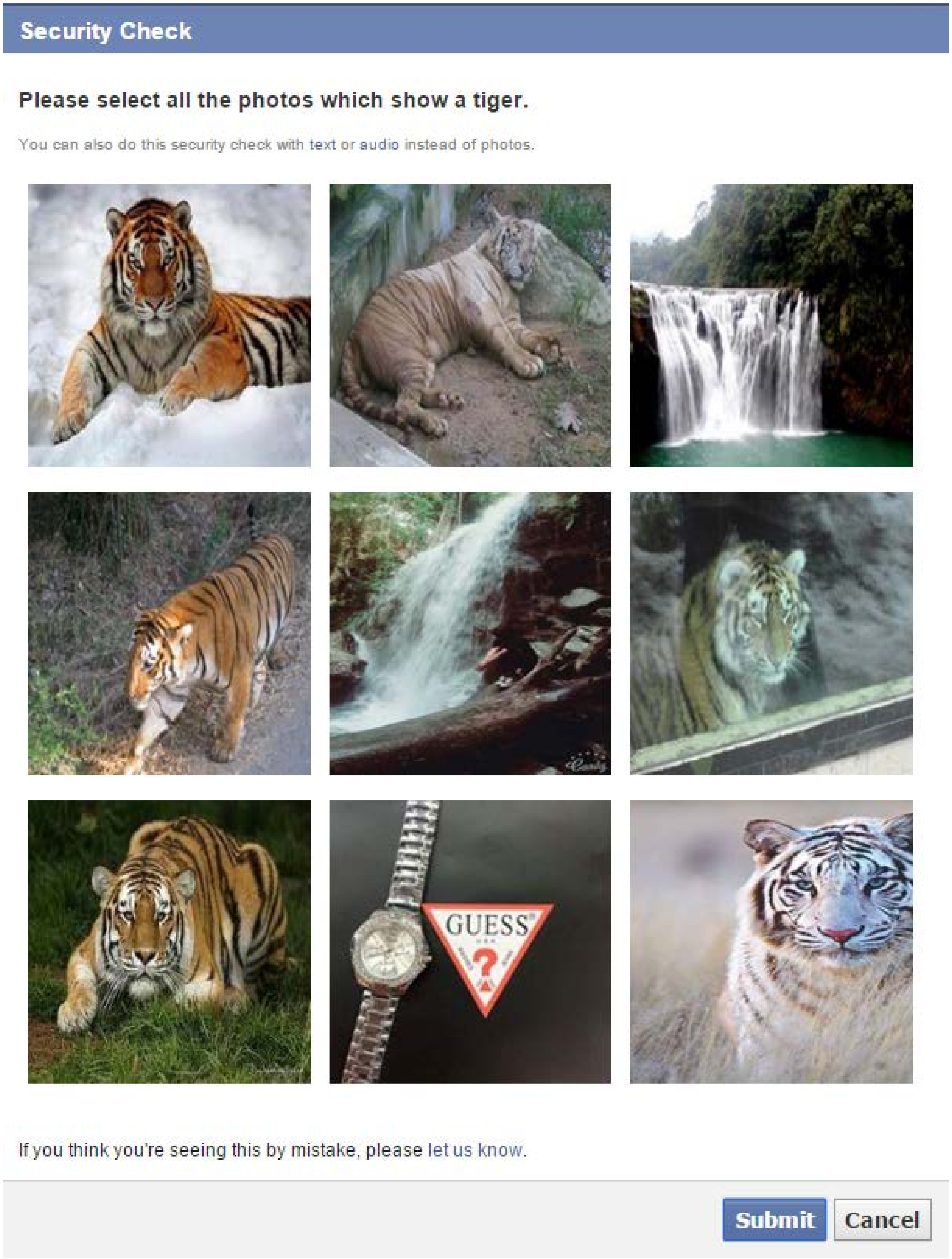
No Is this a test version of another app? Learn More.

Category

Business ▾

By proceeding, you agree to the Facebook Platform Policies and the Facebook Privacy Policy

Click **Create App ID**. A pop-up window will then ask you a challenge question to prove you’re not a robot. (You're not, are you?)



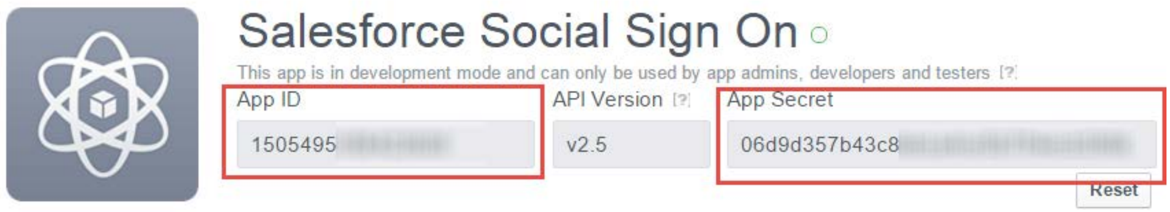
Security Check

Please select all the photos which show a tiger.

You can also do this security check with text or audio instead of photos.

If you think you're seeing this by mistake, please let us know.

You will then be redirected to a Dashboard. Copy the **App ID** and **App Secret**.

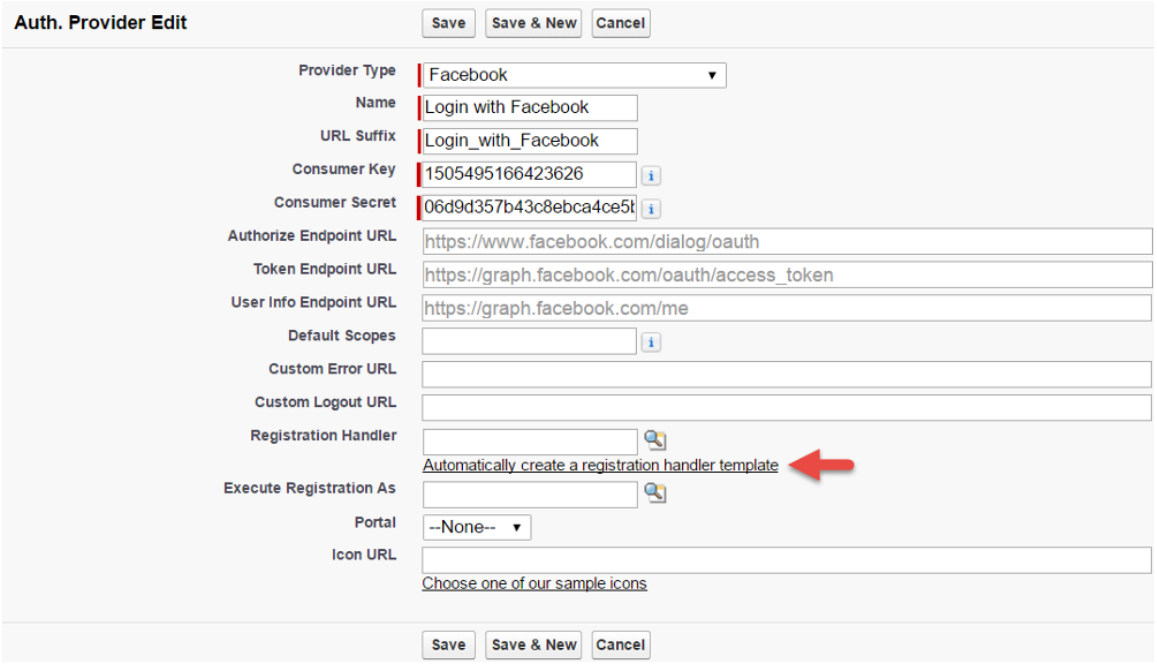


Salesforce Social Sign On ○

This app is in development mode and can only be used by app admins, developers and testers [?]

App ID	1505495[redacted]
API Version [?]	v2.5
App Secret	06d9d357b43c8[redacted]

8. Go back to the Auth. Provider Setup window and fill in the consumer key (Facebook App ID) and the consumer secret (Facebook App Secret.)
9. Choose the Apex class for Registration Handler: either existing class or click the **Automatically create dummy registration handler** link to automatically create a Registration Handler class.



Auth. Provider Edit

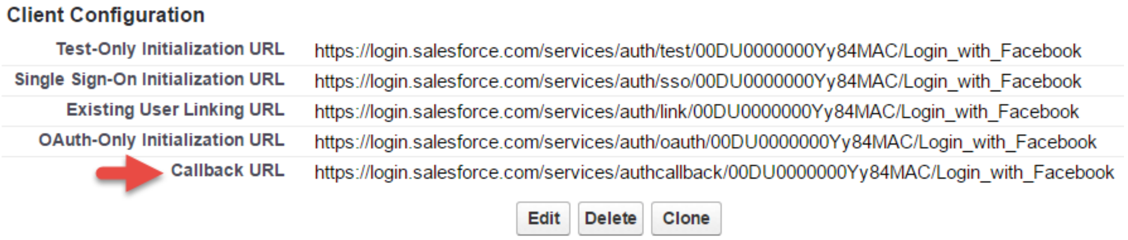
Provider Type	Facebook
Name	Login with Facebook
URL Suffix	Login_with_Facebook
Consumer Key	1505495166423626
Consumer Secret	06d9d357b43c8ebca4ce5t
Authorize Endpoint URL	https://www.facebook.com/dialog/oauth
Token Endpoint URL	https://graph.facebook.com/oauth/access_token
User Info Endpoint URL	https://graph.facebook.com/me
Default Scopes	
Custom Error URL	
Custom Logout URL	
Registration Handler	Automatically create a registration handler template
Execute Registration As	
Portal	--None--
Icon URL	Choose one of our sample icons

NOTE:

The Auto-generated class has a lot of hard-coded values and must be modified. See Appendix 1 for the Auto-generated class and sample modified code.

Also choose the user that will execute the registration (must have "Manage Users" permission.) Next, update the User Info Endpoint URL: https://graph.facebook.com/me?fields=picture,name,first_name,last_name,id,email,friends

10. Click **Save**. Salesforce will provide following URLs:



Client Configuration

Test-Only Initialization URL	https://login.salesforce.com/services/auth/test/00DU0000000Yy84MAC/Login_with_Facebook
Single Sign-On Initialization URL	https://login.salesforce.com/services/auth/sso/00DU0000000Yy84MAC/Login_with_Facebook
Existing User Linking URL	https://login.salesforce.com/services/auth/link/00DU0000000Yy84MAC/Login_with_Facebook
OAuth-Only Initialization URL	https://login.salesforce.com/services/auth/oauth/00DU0000000Yy84MAC/Login_with_Facebook
Callback URL	https://login.salesforce.com/services/authcallback/00DU0000000Yy84MAC/Login_with_Facebook

Test-Only Initialization URL: Administrators use this URL to ensure the third-party provider is set up correctly. The administrator opens this URL in a browser, signs in to the third party, and is redirected back to Salesforce with a map of attributes.

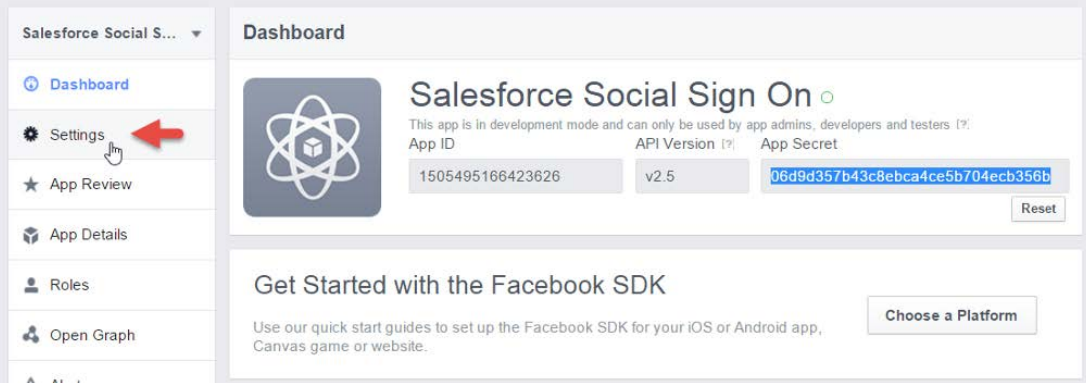
Single Sign-On Initialization URL: Use this URL to perform single sign-on into Salesforce from a third party (using third-party credentials). The end user opens this URL in a browser, and signs in to the third party. This then either creates a new user for them, or updates an existing user, and then signs them into Salesforce as that user.

Existing User Linking URL: Use this URL to link existing Salesforce users to a third-party account. The end user opens this URL in a browser, signs in to the third party, signs in to Salesforce, and approves the link.

Oauth-Only Initialization URL: Use this URL to obtain OAuth access tokens for a third party. Users must authenticate with Salesforce for the third-party service to get a token; this flow does not provide for future single sign-on functionality.

Callback URL: Use the callback URL for the endpoint that the authentication provider calls back to for configuration. The authentication provider has to redirect to the Callback URL with information for each of the above client configuration URLs.

11. Copy the Callback URL. Navigate back to the Facebook App page and click **Settings**.



Dashboard

Salesforce Social Sign On

This app is in development mode and can only be used by app admins, developers and testers

App ID: 1505495166423626

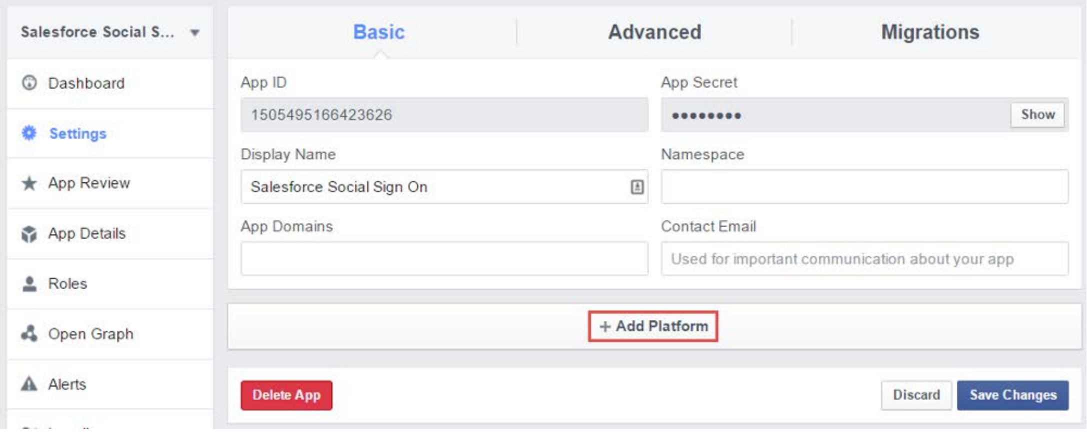
API Version: v2.5

App Secret: 06d9d357b43c8ebca4ce5b704ecb356b

Get Started with the Facebook SDK

Use our quick start guides to set up the Facebook SDK for your iOS or Android app, Canvas game or website.

Click **Add Platform** and choose **Website**.



App ID: 1505495166423626

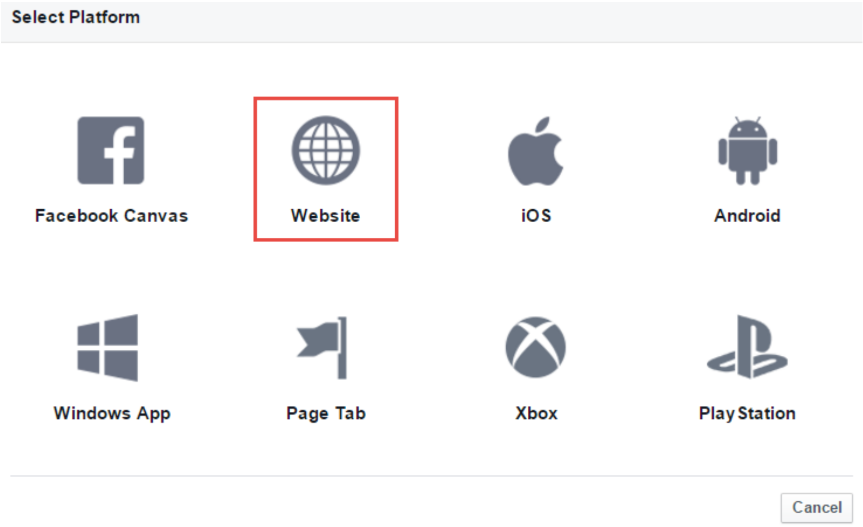
App Secret: ••••••••

Display Name: Salesforce Social Sign On

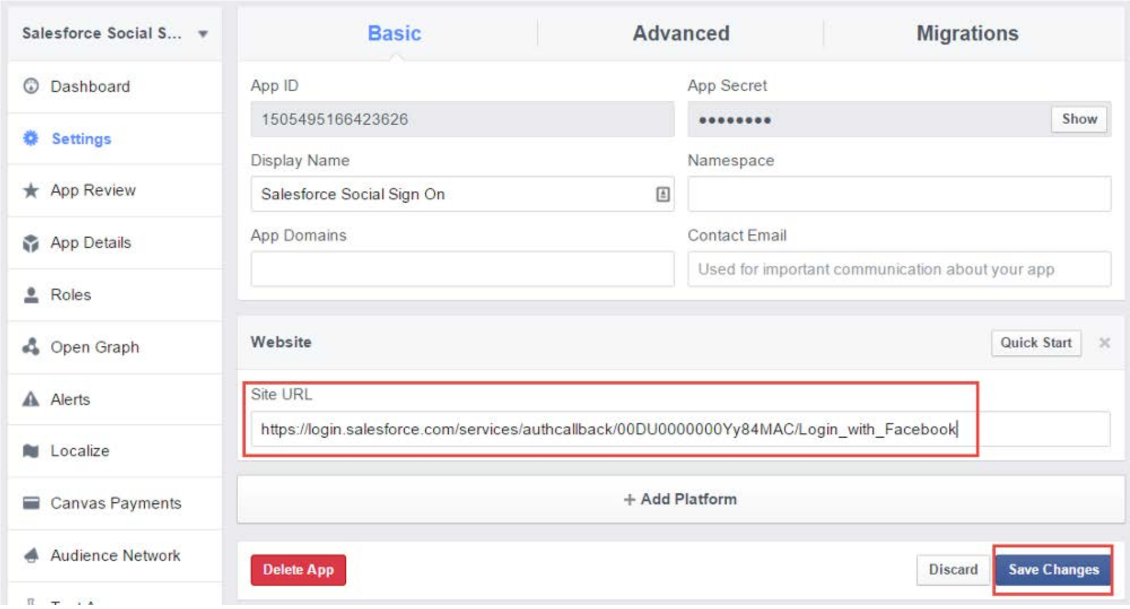
Namespace:

App Domains:

Contact Email: Used for important communication about your app


Select Platform
Facebook Canvas
Website
iOS
Android
Windows App
Page Tab
Xbox
Play Station
Cancel

Paste the Callback URL in the Site URL text box. Click **Save Changes**.



App ID: 1505495166423626

App Secret: •••••••••

Display Name: Salesforce Social Sign On

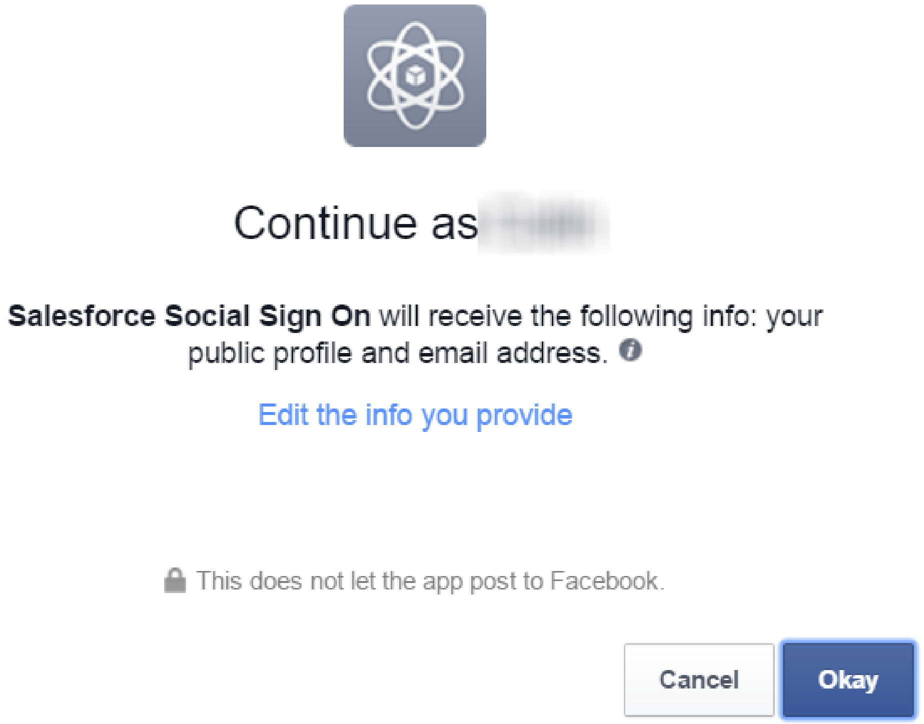
Namespace:

App Domains:

Contact Email: Used for important communication about your app

Site URL: https://login.salesforce.com/services/authcallback/00DU0000000Yy84MAC/Login_with_Facebook

12. Use the Test Only Initialization URL to test a response from Salesforce via Facebook.
    a. Make sure you are logged out of Facebook.
    b. Go the Test-Only Initialization URL.
    c. You will be prompted to log in to Facebook.
    d. Facebook will prompt you to allow the app you just created.



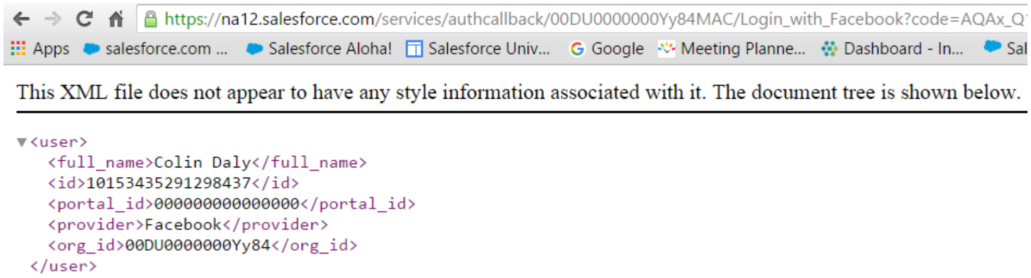
Continue as [redacted]

Salesforce Social Sign On will receive the following info: your public profile and email address.

Edit the info you provide

This does not let the app post to Facebook.

The response shows the data received by Salesforce from Facebook (in an Auth.UserData object.)



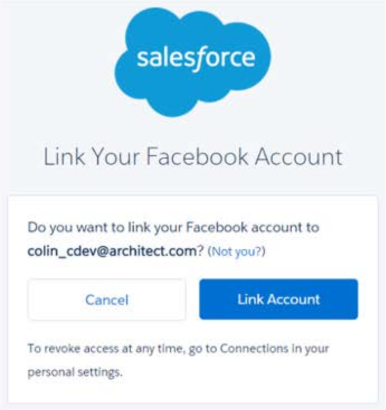
https://na12.salesforce.com/services/authcallback/00DU0000000Yy84MAC/Login_with_Facebook?code=AQAx_Q

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<user>
   <full_name>Colin Daly</full_name>
   <id>10153435291298437</id>
   <portal_id>000000000000000</portal_id>
   <provider>Facebook</provider>
   <org_id>00DU0000000Yy84</org_id>
 </user>
```

13. Link an existing SFDC account to a Facebook account. Use the Existing User Linking URL to allow a user to link his/her Facebook account to an existing Salesforce account. This establishes a Third Party Account Link on the user record in Salesforce and allows for SSO using third-party credentials.

    If you are not presently logged into Facebook, it will prompt you to do so. If you are not currently logged into Salesforce, it will direct you to log in as the Salesforce user that you want to link to Salesforce. Salesforce will prompt you to allow link to the third-party account.

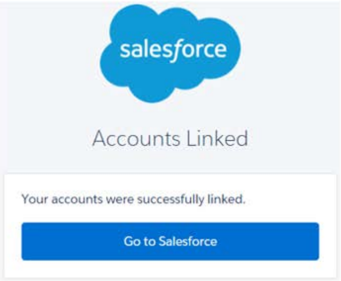


Link Your Facebook Account

Do you want to link your Facebook account to colin_cdev@architect.com? (Not you?)

To revoke access at any time, go to Connections in your personal settings.

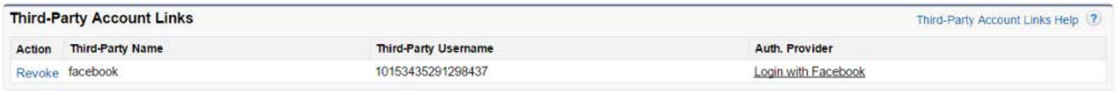
Click **Link Account** and the accounts are successfully linked.



Accounts Linked

Your accounts were successfully linked.

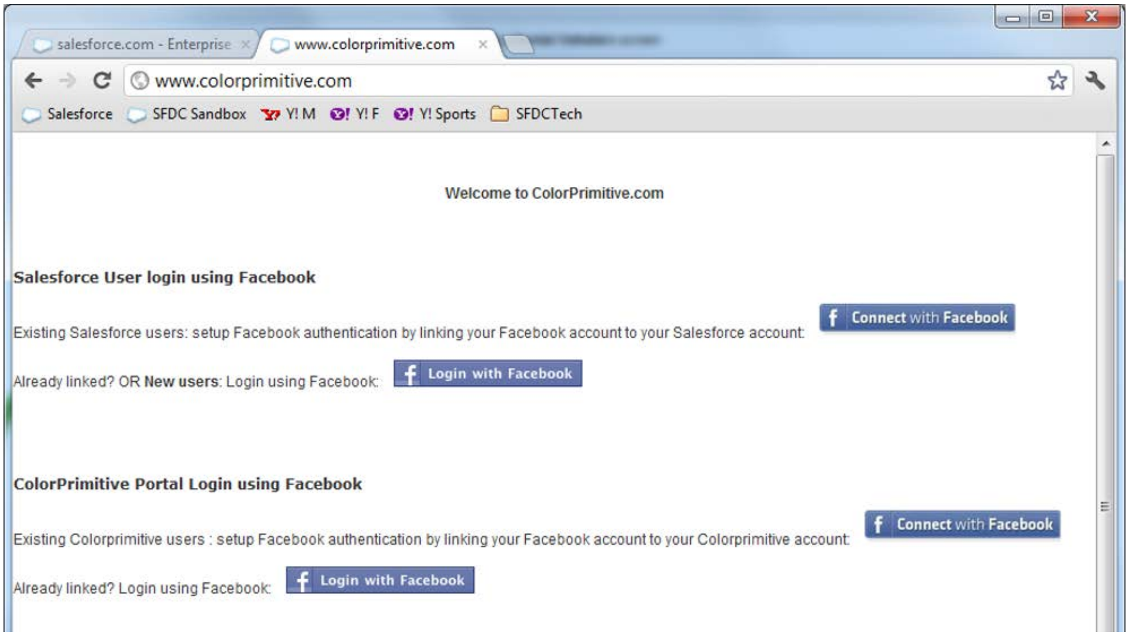
To view the third-party link in Salesforce, go to **Administer | Manage Users | Users** and click on the **user record**. Scroll down to the Third-Party Account Links section. It displays the third party and username and has ability to revoke the access (or remove.)


Third-Party Account Links

Action	Third-Party Name	Third-Party Username	Auth. Provider
Revoke	facebook	10153435291298437	Login with Facebook

14. Use the Single Sign-On Initialization URL to sign in using an already linked Salesforce account, or to provision a new user in Salesforce (either regular or Community). The provisioning is done by the Registration Handler class. (Refer to Appendix 1).
    a. It will prompt you to log in to Facebook (if not already logged in.)
    b. It then logs into SFDC automatically and redirects to the SFDC home page.

15. Sample public website page with URLs for linking and SSO:



Welcome to ColorPrimitive.com

Salesforce User login using Facebook

Existing Salesforce users: setup Facebook authentication by linking your Facebook account to your Salesforce account: **Connect with Facebook**

Already linked? OR **New users**: Login using Facebook: **Login with Facebook**

ColorPrimitive Portal Login using Facebook

Existing Colorprimitive users : setup Facebook authentication by linking your Facebook account to your Colorprimitive account: **Connect with Facebook**

Already linked? Login using Facebook: **Login with Facebook**

16. Optionally, add a Facebook login option on the Salesforce login page or Community Login Page.

For the Community Login Page:

- Go to **App Setup | Customize | Communities | All Communities.**
- Select the Community by clicking **Manage**. This will navigate to the Community Manage page.
- Go to **Administration | Login & Registration**.
- In the Login section, under External users can login or register with: mark Login with Facebook as **True**.
- Save the setting.
- Go to the Community Login Page and select **Login with Facebook.**

For the Salesforce Login Page:

- Go to Setup | Domain Management | My Domain.
- In Authentication Configuration section, under Authentication Service mark Login With Facebook as True.
- Save the setting.
- Go to the My Domain URL and select **Login with Facebook**.

APPENDIX 1

SAMPLE AUTH REGISTRATION HANDLER CLASS (WITH MODIFICATIONS)

```
//Registration Handler for FB Authentication for SFDC users

global class AutocreatedRegHandler1454925297642 implements Auth.RegistrationHandler{
    global boolean canCreateUser(Auth.UserData data) {

        //TODO: Check whether we want to allow creation of a user with this data
        return true;
    }

    global User createUser(Id portalId, Auth.UserData data){

        system.debug('createUser');
        system.debug( data.attributeMap );

        if(!canCreateUser(data)) {
            //Returning null or throwing an exception fails the SSO flow
            return null;
        }

        if(data.attributeMap.containsKey('sfdc_networkid')) {

            //We have a community id, so create a user with community access
            //TODO: Get an actual account

            Account a = [SELECT Id FROM account WHERE name='Acme'];
            Contact c = new Contact();

            c.accountId = a.Id;
            c.email = data.email;
            c.firstName = data.firstName;
            c.lastName = data.lastName;

            insert(c);

            //TODO: Customize the username and profile. Also check that the username doesn't already exist and
            //possibly ensure there are enough org licenses to create a user. Must be 80 characters or less.

            User u = new User();
            Profile p = [SELECT Id FROM profile WHERE name='Customer Community User'];

            u.username = data.firstname +'.'+ data.lastname + '@acmecorp.com';
            u.email = data.email;
            u.lastName = data.lastname;
            u.firstName = data.firstname;

            String alias = data.firstname+'.'+data.lastname;

            //Alias must be 8 characters or less
            if(alias.length() > 8) {
                alias = alias.substring(0, 8);
            }

            u.alias = alias;
            u.languagelocalekey = UserInfo.getLocale();
            u.localesidkey = UserInfo.getLocale();
            u.emailEncodingKey = 'UTF-8';
            u.timeZoneSidKey = 'America/Los_Angeles';
            u.profileId = p.Id;
            u.contactId = c.Id;

            return u;

        }else{

            //This is not a community, so create a regular standard user
            User u = new User();
            Profile p = [SELECT Id FROM profile WHERE name='Standard User'];

            //TODO: Customize the username. Also check that the username doesn't already exist and
            //possibly ensure there are enough org licenses to create a user. Must be 80 characters
            //or less.

            u.username = data.lastName+'.'+data.firstName + '@demoorg.com';
            u.email = data.email;
            u.lastName = data.lastName;
            u.firstName = data.firstName;

            String alias = data.lastName+'.'+data.firstName;

            //Alias must be 8 characters or less

            if(alias.length() > 8) {
                alias = alias.substring(0, 8);
            }

            u.alias = alias;
            u.languagelocalekey = UserInfo.getLocale();
            u.localesidkey = UserInfo.getLocale();
            u.emailEncodingKey = 'UTF-8';
            u.timeZoneSidKey = 'America/Los_Angeles';
            u.profileId = p.Id;

            return u;
        }

    }

    global void updateUser(Id userId, Id portalId, Auth.UserData data){

        User u = new User(id=userId);

        //TODO: Customize the username. Must be 80 characters or less.
        //u.username = data.username + '@myorg.com';

        u.email = data.email;
        u.lastName = data.lastName;
        u.firstName = data.firstName;

        //String alias = data.username;
        //Alias must be 8 characters or less
        //if(alias.length() > 8) {
            //alias = alias.substring(0, 8);
        //}

        //u.alias = alias;

        update(u);
    }
}
```

ADDITIONAL FILES

ANNOTATIONS

```
<?xml version='1.0' encoding='UTF-8'?>
<remarks xmlns="http://www.standardnine.com/s9ml" data-uuid="f7c7701f81d94e638509f883adb5b9b7">
    <remark key="key name" data-uuid="a13309c28a0b47cca5144ef4e5ab4d3e">
        <title>Remark Title</title>
        <text>Text of remark.</text>
    </remark>
</remarks>

```

CONFIG

```
<?xml version='1.0' encoding='UTF-8'?>
<config xmlns="http://www.standardnine.com/s9ml" data-uuid="5539325e2bde4e218ae5be4c4db1c871">
   <!-- <outlines>
      <section level="0">
        <excludes>
          <exclude paths="front_matter/*.s9ml"/>
          <exclude paths="chapter*/ch01_section_1.html"/>
        </excludes>
      </section>
      <section level="1" xpath="//xhtml:section/xhtml:h2">
        <excludes>
          <exclude paths="frontmatter/*.html"/>
          <exclude paths="chapter*/*guide*.html"/>
        </excludes>
      </section>
      <section level="2" xpath="//xhtml:section/xhtml:h3">
        <excludes>
          <exclude paths="frontmatter/*.html"/>
          <exclude paths="chapter*/*guide*.html"/>
        </excludes>
      </section>
    </outlines> -->
    <outlines>
        <section level="1" xpath=".//xhtml:body//xhtml:h2" />
        <section level="2" xpath=".//xhtml:body//xhtml:h3" />
    </outlines>
    <images>
        <profile name="jpg_90">
            <format>jpg</format>
            <quality>90</quality>
        </profile>
        <apply profile="jpg_90">
            <alpha>false</alpha>
            <glob>*.png</glob>
        </apply>
    </images>
    <exports> <!--
       <profile type="epub"/>
       <profile type="epub_with_video"/> -->
       <profile type="pdf"/>
      <!-- <profile type="ibooks"/>
       <profile type="kindle"/>
       <profile type="scorm12"/> -->
   </exports>
</config>
```

ADDITIONAL ACTIVITIES

1. ESTABLISH A FEDERATION ID

For this single sign-on implementation, we'll set a user attribute that links the user between their Salesforce organization and an external application.

- From Setup click Manage Users | Users.
- Click Edit next to your current user.
- In the Single Sign On Information section, enter the Federation ID: admin@universalcontainers.com.
- For this example, we arbitrarily made up a Federation ID. The Federation ID is a unique username for each user that can be shared across multiple applications. Sometimes this is the employee ID for that user. The important part of the Federation ID is that it is not duplicated for more than one user within a single Salesforce organization (you can have the same Federation ID for the same user in more than one Salesforce organization).
- Click Save.

2. SET UP YOUR IDENTITY PROVIDER

You'll use Axiom, a single sign-on testing app hosted on Heroku, to go through the steps of setting up an identity provider. Get an identity provider certificate from the Axiom app and set it up in your Salesforce organization.

- In a new browser window, go to Axiom.
- Click **SAML Identity Provider & Tester**.
- Click **Download the Identity Provider Certificate**. The certificate validates signatures, and you need to upload it to your Salesforce organization. Remember where you save it.
- In your Salesforce organization, from **Setup**, click **Security Controls | Single Sign-On Setting**s.
- Click Edit.
- Select SAML Enabled.
- Click Save.
- In SAML Single Sign-On Settings, click New.
- Enter the following values:

17. Name: Axiom Test App
18. Issuer: http://axiomsso.herokuapp.com
19. Identity Provider Certificate: Choose the file you downloaded in step 3.
20. Request Signing Certificate: Leave as the Default Certificate.
21. SAML Identity Type: Select **Assertion contains the Federation ID from the User object**.
22. SAML Identity Location: Select **Identity is in the NameIdentifier element of the Subject statement**.
23. Service Provider Initiated Request Binding: Select **HTTP Redirect**.
24. Entity Id: Enter your My Domain name including "https," such as https://universalcontainers.my.salesforce.com
25. Click **Save** and leave the browser page open.

3. GENERATE SAML

- To continue this tutorial, please click here

4. TRAILHEAD MODULES

- User Authentication
- Identity Basics