



Salesforce External Identity Implementation Guide

Salesforce, Winter '17



CONTENTS

SALESFORCE IDENTITY FOR EXTERNAL USERS	1
WHAT IS SALESFORCE IDENTITY FOR EXTERNAL USERS?	2
HOW CAN I USE SALESFORCE IDENTITY FOR EXTERNAL USERS?	3
Acquire and Engage New Users	3
Deliver a Consistent Experience with a Single Identity Across All Channels	3
Secure and Manage Your Customer and Partner Ecosystems	4
Integrate and Customize to Your Business Needs	4
EXTERNAL IDENTITY LICENSES	5
EXTERNAL IDENTITY AND COMMUNITIES	6
PREPARE YOUR ORG	7
Create a Developer Org	7
Set Up a My Domain	7
Control Authorization with Custom Profiles and Roles	8
Create an Account for CRM Integration	9
CREATE A BRANDED LOGIN PAGE	10
Set Up Your Community	10
Customize Your Community	11
Create a Branded Login Experience	13
Activate Your Community	14
ENABLE SELF-REGISTRATION	16
Create a Basic Branded Self-Registration Page	16
Add Fields to Collect Additional Information	16
Add a Password Field to Enable Login Directly During Registration	18
ENABLE SELF-REGISTRATION FOR B2C USERS (OPTIONAL)	20
Enable Person Accounts	20
Configure Self-Registration for Person Accounts	20
ACCEPT IDENTITY FROM AN EXISTING IDENTITY PROVIDER	22
Social Sign-On	22
Create an Auth. Provider	23
Customize Your Registration Handler	23

Enable Your Auth. Provider in Your Community 23

ACCEPT USER IDENTITY WITH SAML AND JUST-IN-TIME PROVISIONING 25

SET UP SSO AND ACCESS FOR YOUR WEB APP 26

Create a Connected App for Your Web App 26

Create a Sample Service Provider on Heroku 27

Configure Salesforce Identity to Provide Identity for Your App 27

Authorize Your Web Application 27

Configure Your App to Trust Salesforce Identity 28

Personalize Your App with Custom Attributes 28

More About Single Sign-On for Your Web App 29

PROVIDE SSO AND ACCESS FOR MOBILE APPS 30

Create a Connected App for Your Mobile App 30

Install the Salesforce Mobile SDK 31

Create a Mobile App 31

Configure the Mobile App to Point to Your Community 31

More About Single Sign-On for Your Mobile App 32

EXTERNAL IDENTITY ON GITHUB AND SUCCESS COMMUNITY 33

INDEX 34

SALESFORCE IDENTITY FOR EXTERNAL USERS

This implementation guide describes how to set up Salesforce Identity for external users. Refer to this guide when you want to set up external identity on your own Salesforce org.

But first, to get a thorough understanding of external identity, there's nothing better than learning by doing. So take advantage of these Trailhead modules.

Identity Basics

Get an overview of the features in Salesforce Identity and see how external identity fits into its feature set. Familiarize yourself with key identity terms like single sign-on, social sign-on, identity providers (IdP), and service providers (SPs). Get familiar with the identity protocols, Security Assertion Markup Language (SAML), OAuth 2.0, and OpenID Connect.

External Identity for Customers

Walk through the entire process of setting up external identity on a trial Developer org. The steps in this guide are similar to the steps you follow in the Trailhead module.

WHAT IS SALESFORCE IDENTITY FOR EXTERNAL USERS?

Salesforce Identity is an Identity and Access Management (IAM) service that connects users to your applications, services, and devices. It provides a centralized point of management for your admins and provides a single, trusted identity for your end users. Traditionally, IAM services have focused on employee-facing use cases. Today, companies are using identity as a way to better connect with their customers and partners. We call this external identity.

When used for external identities, Salesforce Identity transforms CRM contacts into real digital identities that can self-register, log in, update their profile, and securely access web and mobile applications with a single identity. Plus, it's customized to your specific business process and brand using the power of the Salesforce App Cloud.

By delivering identity services directly from the same platform you use for sales, service, and marketing, you can recognize users across all your digital channels and create a consistent experience for customers and partners across all lines of business. The information and insight gathered converges with your existing CRM data and processes, thus building a single view of all your relationships.

Using Salesforce Identity, you build deeper, richer relationships with customers and partners by creating and maintaining a single identity for interaction across all channels.

HOW CAN I USE SALESFORCE IDENTITY FOR EXTERNAL USERS?

Salesforce Identity for external users offers a broad set of capabilities for connecting with your customers and partners, as well as extensive customization and integration options. Here are some common use cases and features.

[Acquire and Engage New Users](#)

Your business is growing, and you need to onboard customers and partners quickly. Salesforce Identity can help enable and scale your customer and partner acquisition processes with self-registration, social sign-on, and CRM integration.

[Deliver a Consistent Experience with a Single Identity Across All Channels](#)

Salesforce Identity lets you engage with your users everywhere. Get a single, 360-degree view of your users while delivering a consistent, streamlined end-user experience for your brand.

[Secure and Manage Your Customer and Partner Ecosystems](#)

By centralizing management of your users, Salesforce Identity makes life easy for your admins. They have a single place to manage Identity users and create reports and dashboards on their access.

[Integrate and Customize to Your Business Needs](#)

Salesforce Identity is integrated into the Salesforce App Cloud and is fully customizable, extensible, and scalable for any business.

Acquire and Engage New Users

Your business is growing, and you need to onboard customers and partners quickly. Salesforce Identity can help enable and scale your customer and partner acquisition processes with self-registration, social sign-on, and CRM integration.

Self-Registration

External users can create user accounts quickly and easily with fully branded and customizable registration processes.

Social Sign-On

Customers and prospects can bring their own identity from social networks and public providers, such as Facebook, Google, Amazon, and PayPal.

CRM and Back-Office Integration

You can easily integrate your customers with your Salesforce org. When you run registration on your customer platform, identity data is no longer stuck in an IT system. Enrich your CRM data, create leads, link to your back-office customer records, and drive approval processes by implementing an external identity solution.

Deliver a Consistent Experience with a Single Identity Across All Channels

Salesforce Identity lets you engage with your users everywhere. Get a single, 360-degree view of your users while delivering a consistent, streamlined end-user experience for your brand.

Single Sign-On (SSO)

Save your users' time by letting them log in once to seamlessly access your apps. Uses secure industry standards like SAML, OpenID, and OAuth.

Mobile Identity

Deliver mobile apps to your customers with automatic SSO, authorization, and mobile-specific policies. Salesforce gives you a robust, open-source mobile SDK to easily create your mobile apps.

Cloud Directory Services

Adapt your business with customizable fields, automatable workflows, batch processing, and delegated administration through cloud Cloud directory services.

Secure and Manage Your Customer and Partner Ecosystems

By centralizing management of your users, Salesforce Identity makes life easy for your admins. They have a single place to manage Identity users and create reports and dashboards on their access.

Authorization and Policy Management

Deliver the right experience to your users at the right time and for the right reasons. Built-in access management, authorization, and robust policies make it easy for you to effective identity management.

Multifactor Authentication

Add an extra layer of security when logging in or accessing critical resources using secure, mobile two-factor authentication.

Provisioning and Unprovisioning Apps

Provide access and personalization to your apps with a customizable push-provisioning engine for just-in-time provisioning and single sign-on.

Reporting and Dashboards

Gain visibility into usage, adoption, and security with drag-and-drop customizable reports and dashboards.

Integrate and Customize to Your Business Needs

Salesforce Identity is integrated into the Salesforce App Cloud and is fully customizable, extensible, and scalable for any business.

Fully Branded

Extend your company's brand securely with drag-and-drop branding for login, self-registration, and federation services.

Workflows and Business Processes

Scale your administration and integration efforts with visually designed workflow processes.

Open APIs and Open Standards

Take advantage of the full suite of development tools that Salesforce Identity offers. It provides APIs for everything you need and supports major open identity standards, including SAML, OAuth 2.0, OpenID Connect, and SCIM.

EXTERNAL IDENTITY LICENSES

The External Identity license enables you to easily deliver identity services, such as single sign-on to your customers and partners. The license is optimized for customer success. By tracking monthly active users, you pay only for users that engage with your services.

External Identity is a standalone license and purchased in blocks of monthly active users. It seamlessly coexists with Community licenses. It's also included for free with all paid community user licenses in Enterprise, Performance, and Unlimited Editions. Each Developer Edition org includes five External Identity user licenses.

With an External Identity license, you can access Contact objects, Account objects, Asset objects, and two custom objects so that you can deliver powerful self-service applications. The license includes extra data storage and API requests, but that might not be adequate for your use case. Make sure that your org has sufficient resources before rolling out your external identity system. For more information, contact your Salesforce representative.

The following licenses are also available.

Identity Only

Enables use cases similar to External Identity for your internal employees.

Identity Connect

An on-premise component that synchronizes users with Microsoft Active Directory (AD). While not commonly used in external scenarios, occasionally companies store their external users in AD.

Customer Community Plus or Partner Community

For customers who want to implement delegated administration. You can use these licenses for the org's Salesforce admins. They pair well with the External Identity license.

EXTERNAL IDENTITY AND COMMUNITIES

Communities are branded spaces for employees, customers, and partners to connect. You can customize and create communities to deliver specific business applications and services, including identity services.

Don't confuse community user licenses with underlying community capabilities. While Salesforce provides community licenses for use cases like customer self-service, there is no direct correlation between a community and community licenses.

Salesforce External Identity uses communities for its deployment. Community deployment accommodates for a unique brand and configuration, so you can act as both a service provider and identity provider for all your applications without your customers realizing that the service runs on Salesforce. Similar to My Domain, communities allow for unique DNS domains, either in the format of `https://customername.force.com` or custom SSL domains.

For more information, see [Salesforce Communities Overview](#) in the Salesforce Help.

PREPARE YOUR ORG

To begin your walkthrough of a typical external identity implementation, you create a developer org. Because Salesforce Identity integrates with the customer and partner business processes that you run on Salesforce, you perform a few basic administrative tasks to set up a typical deployment.

For a refresher on the entire process for creating an external identity community, watch the [How to Get Started Basic Developer Edition Org for Identity](#) video. Then follow the tasks in this section.

[Create a Developer Org](#)

Developer orgs have all the features and licenses you need to get started with Salesforce Identity. You can use existing orgs, trial orgs, and sandboxes for external identity, but Developer orgs are a great way to get started.

[Set Up a My Domain](#)

My Domain allows admins to define their own DNS subdomain name and associate it with their Salesforce org. A subdomain name gives you more control over the authentication options for your company and also highlights your company name.

[Control Authorization with Custom Profiles and Roles](#)

One important facet of identity and access management is the ability to control who has access to what. To get started, you set up two basic ways to control authorization: profiles and roles.

[Create an Account for CRM Integration](#)

One of the great things about Salesforce Identity for external users is that it's already integrated with your customer success platform. By driving identity on the same platform that you use for managing your customers and partners, you simplify your integration requirements while providing your users a better experience.

Create a Developer Org

Developer orgs have all the features and licenses you need to get started with Salesforce Identity. You can use existing orgs, trial orgs, and sandboxes for external identity, but Developer orgs are a great way to get started.

1. Go to <https://developer.salesforce.com/signup>.
2. Enter your contact information.
3. Choose a unique username.
4. Submit the form and wait for your welcome email.
5. In the welcome email, click the link and set your password.

That's it—you now have your own Developer org.

Set Up a My Domain

My Domain allows admins to define their own DNS subdomain name and associate it with their Salesforce org. A subdomain name gives you more control over the authentication options for your company and also highlights your company name.

The Salesforce SAML identity provider requires that the org be configured with a subdomain. Fortunately, setting up a subdomain is easy with My Domain.

1. In your developer org, from Setup, enter *My Domain* in the **Quick Find** box, then select **My Domain**.
2. Enter the subdomain name you want to use within the Salesforce URL. For example, a company called Universal Containers uses the subdomain *universalcontainers*. Then the company's login URL is `https://universalcontainers.my.salesforce.com/`.
3. Click **Check Availability**. If the name is already taken, choose a different one.
4. Click **Register Domain**. Salesforce updates its domain registries with your new subdomain. When it's done, you receive an email message with a subject like, "Your Developer Edition domain ready for testing." It takes just a few minutes.
5. After you receive the email, click the link to go to your subdomain. You're automatically signed in to the domain.
6. On the My Domain page, click **Deploy to Users** and you're done!

For more instructions on how you can use My Domain for internal identity use cases, see the [Salesforce Identity Implementation Guide](#).

Control Authorization with Custom Profiles and Roles

One important facet of identity and access management is the ability to control who has access to what. To get started, you set up two basic ways to control authorization: profiles and roles.

Profiles define how users access objects and data and what they can do in Salesforce and with your connected apps. You can use a default profile, but it's a best practice to clone and customize the default profiles to meet your own organizational requirements.

1. In your developer org, from Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles**.
 - a. Click **Clone** next to **External Identity User**.
 - b. Enter a name for the profile. Let's call it *Customers*.
 - c. Click **Save**.
2. Customize your profile.
 - a. Click **Edit**.
 - b. Search for API Enabled, and then select the checkbox next to this permission.
 - c. Click **Save**.

3. Create a role structure.

Beyond the permissions offered by profiles and permission sets, Salesforce lets you specify sharing settings to determine the access that users have to your Salesforce org's data. A user role hierarchy is one option for controlling sharing. You can combine it with sharing settings to segment data visibility.

- a. From Setup, enter *Role* in the **Quick Find** box, then select **Roles**.
 - b. From the dropdown list, select **Product-based Sample**, then select **Set Up Roles**.
 - c. Under CEO, click **Add Role**.
 - d. For the role label, enter *Customer Manager*.
 - e. Click **Save**.
4. Add the external identity role to your user.
 - a. From Setup, enter *Users* in the **Quick Find** box, then select **Users**.
 - b. Find your user and click **Edit**.
 - c. From the Role picklist, select **Customer Manager**.

- d. Click **Save**.

You've now learned the basics of authorization. For more information about configuring authorization, check out the [Data Security](#) module. For more information about creating users and securing access, see the [User Management](#) module.

Create an Account for CRM Integration

One of the great things about Salesforce Identity for external users is that it's already integrated with your customer success platform. By driving identity on the same platform that you use for managing your customers and partners, you simplify your integration requirements while providing your users a better experience.

To simplify integration, external users are tied into the CRM data model within Salesforce. So when your users register or update their profile, you get a consistent view of the customer within your Sales and Service processes.

1. In your developer org, switch to the Sales application.
2. Click the **Accounts** tab.
3. Create an account called Customers.
4. Click **Save**.

You've now completed all the prerequisites for creating an external identity community. To learn more, check out the [Accounts and Contacts](#) Trailhead module.

CREATE A BRANDED LOGIN PAGE

Salesforce Identity for external users makes it simple to deploy a custom identity experience. This custom experience is important. You want your users to experience your brand consistently, whether they're visiting your website for the first time or signing in to your community. The first thing you set up is a community to support your deployment and configure a branded login page.

To learn how, watch the [How to Set Up a Community for Identity and Deploy a Branded Login Page](#) video. Then walk through the steps in this section.

[Set Up Your Community](#)

To enable Salesforce Communities for your org, you provide a community domain name, much like the domain name you created when setting up your org. The community domain collects all your communities under one URL. Typically, your community domain name is your company name.

[Customize Your Community](#)

Let's go through the basics of configuring your community for External Identity use cases. First, use profiles to lock down your data so that external users see only what you want them to see. Then use the Community Manager to customize your users' login experience. This process is where you add your brand to your external identity community.

[Create a Branded Login Experience](#)

While that page is nice, it doesn't feel like it's optimized for a customer-facing brand, does it? Let's start using Community Builder to see how quickly we can customize the user's branding experience.

[Activate Your Community](#)

To complete your external identity community setup, you must activate it.

Set Up Your Community

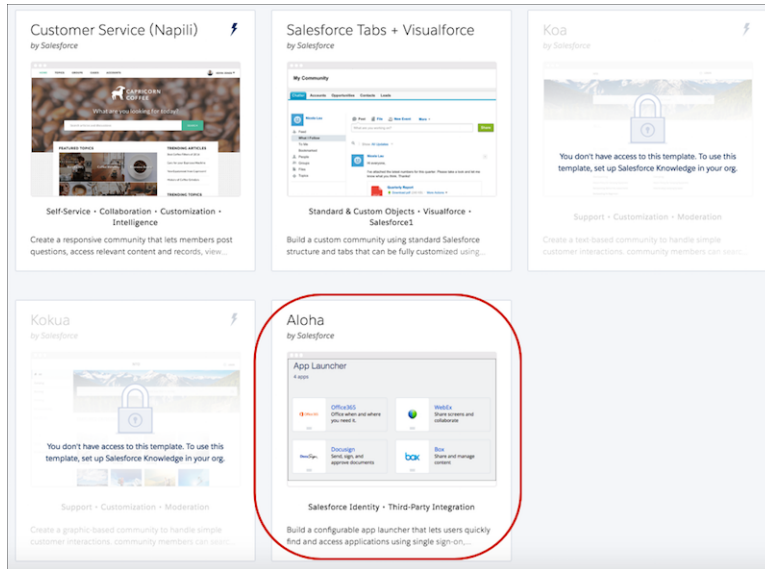
To enable Salesforce Communities for your org, you provide a community domain name, much like the domain name you created when setting up your org. The community domain collects all your communities under one URL. Typically, your community domain name is your company name.

To get started, create a community to host your external identities.

1. In your developer org, from Setup, enter *Communities* in the `Quick Find` box, then select **Communities Settings**.
2. Select **Enable communities**.
3. Enter a memorable domain name. Keep in mind that customers and partners interact with this domain name. Also, after you choose this name, you can't change it. However, for complete control over branding later, you can add a [custom SSL domain](#).
4. Select **Check Availability**.
5. Click **Save**, and then click **OK**.

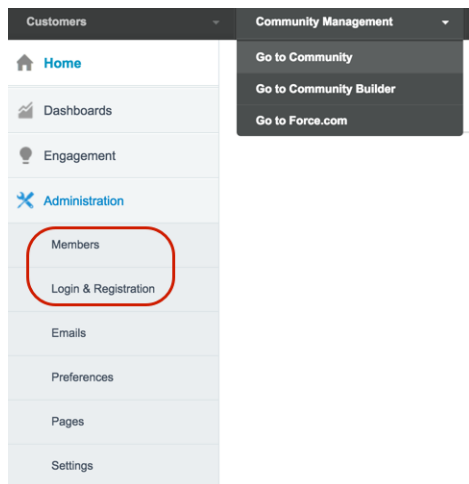
The basic Communities feature is enabled. Next, let's create your external identity community.

6. Click **New Community**.
You're shown a series of community templates.



7. Choose the Aloha template. The Aloha template is designed with external identity in mind. For more information, see the [Getting Started with the Aloha Community Template for Salesforce Identity Guide](#).
8. Give your community a name. Let's call it Customers.
9. Click **Create Community**.

Your new community is now available! Click **Manage & Moderate** to open Community Management. This is where you manage your community and create its login page.



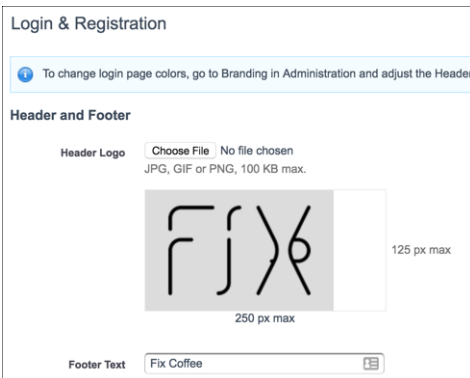
Customize Your Community

Let's go through the basics of configuring your community for External Identity use cases. First, use profiles to lock down your data so that external users see only what you want them to see. Then use the Community Manager to customize your users' login experience. This process is where you add your brand to your external identity community.

You can use your own images for branding or use ours. Download and unzip this file, which contains a sample logo and backgrounds:
<https://www.salesforceidentity.info/ExternalIdentityAssets.zip>.

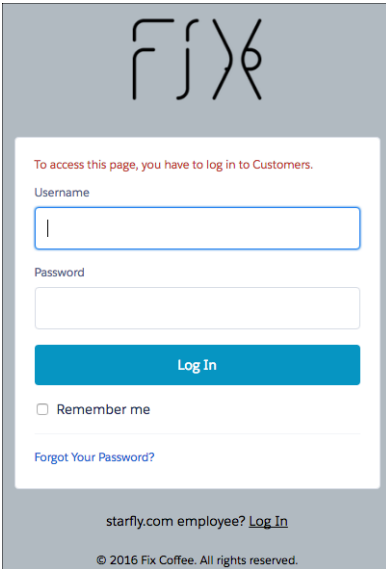
1. Authorize external identity users access to your community.
 - a. From Community Management, select **Administration**, then select **Members**.
You can now control what people can access when they log in to your community.
 - b. In the search list, enter *Portal*.
 - c. In the list of Available Profiles, locate the **Customers** profile and then click **Add** to add it to Selected Profiles.
 - d. Click **Save**.
Salesforce updates membership for your community and sends an email when it's done. Now External Identity users with the Customers profile are authorized to self-register and use single sign-on to access your community.
2. Customize your login page.
 - a. From the Community Manager, select **Login & Registration**.
 - b. Next to Header Logo, click **Choose File**.
 - c. Select a logo. For example, select *fix-logo.png* from the sample files you downloaded.
 - d. Brand the footer text. Let's call it Fix Coffee.
 - e. Click **Save**.

If you used *fix-logo.png* for your logo, the Login & Registration page looks like this.



3. Let's see how your changes appear on a login page.
 - a. Copy the URL of your community—it's in the browser window address bar.
 - b. Open a new browser or incognito window and paste the URL in the address bar.

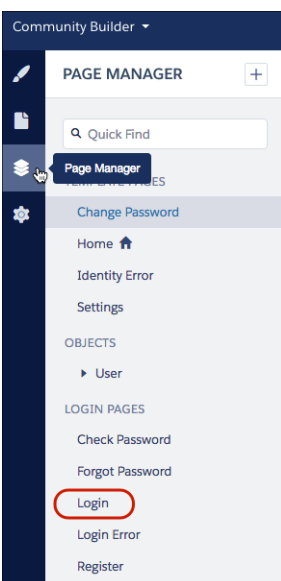
Your page looks something like this.



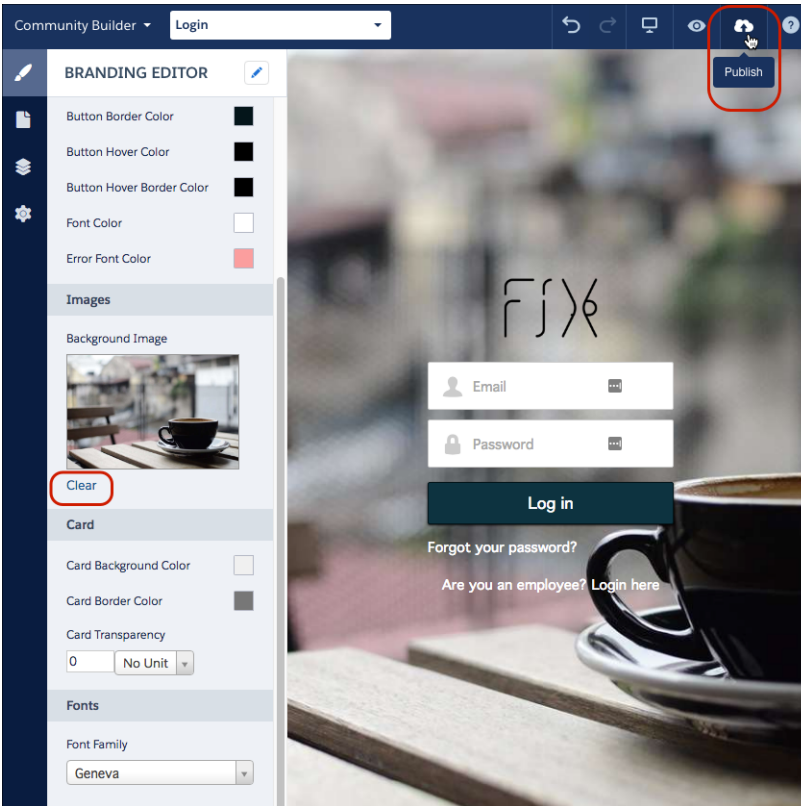
Create a Branded Login Experience

While that page is nice, it doesn't feel like it's optimized for a customer-facing brand, does it? Let's start using Community Builder to see how quickly we can customize the user's branding experience.

1. From Community Management, click your name in the top-right and then click **Setup**.
2. From Setup, enter *All Communities* in the **Quick Find** box, then select **All Communities** and click the **Manage** link next to your community.
3. From Community Management, click **Go to Community Builder**.
4. Under Page Manager, select **Login**, click **Edit**, then the **Branding Editor** icon (🎨). You can now customize your login page brand.



- Under background image, select **Clear** and upload a new background image. If you don't have a background image, upload `cupontablesmall.png` from the sample files you downloaded.
- Tweak your button colors and other options, and make any other changes to brand your login page. Here's an example of what your page can look like after you make some changes.



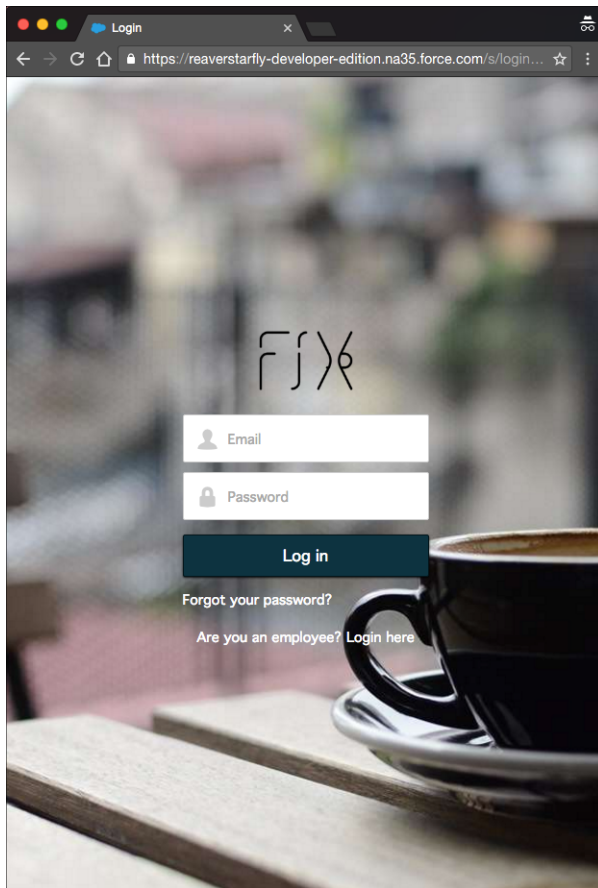
- When you're done branding your page, click **Publish**.
- From Community Builder, select **Go to Community Management**, select **Administration**, and then select **Login & Registration**.
- Under Login, for Page, select **Community Builder** and then select **login** from the page picker.
- Click **Save**.

Activate Your Community

To complete your external identity community setup, you must activate it.

- From Setup enter *All Communities* in the Quick Find box, then select **All Communities** and click the manage link next to your community.
- Select **Administration** and then select **Settings**.
- Click **Activate Community**, then click **OK**.
Salesforce sends you an email when the community is activated.
- Verify that your community is activated.
 - From Community Management, select **Administration**, then select **Settings**, and copy the URL of your community.

- b. Open a new browser or incognito window and paste the URL into the address bar. The browser displays the login page for your community, which looks a lot like the login page you just created with Community Builder.



You've completed the basics of branding with Community Builder! As a side benefit, you've also branded the Forgot Password and Self Registration pages. Speaking of self-registration, that's your next challenge.

ENABLE SELF-REGISTRATION

You've deployed a branded login experience, but you don't have any users. Let's invite your users to log in using self-registration. To learn how, you can watch the [Enabling Self-Registration in a Community](#) video. Then follow the steps in this section.

[Create a Basic Branded Self-Registration Page](#)

Without even knowing it, you've built a branded self-registration page. The Community Builder template and branding you did when you customized your community took care of it automatically. To activate your self-registration page, follow these steps.

[Add Fields to Collect Additional Information](#)

When users register, you often want to ask them for more than just basic information. You can easily add fields to the registration page.

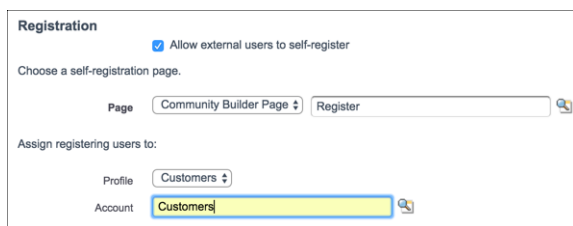
[Add a Password Field to Enable Login Directly During Registration](#)

You can add a password field to your self-registration form to require users to supply a password when they register. Because you're already in Community Builder, it's simple to add the field.

Create a Basic Branded Self-Registration Page

Without even knowing it, you've built a branded self-registration page. The Community Builder template and branding you did when you customized your community took care of it automatically. To activate your self-registration page, follow these steps.

1. In Community Management, select **Login & Registration**.
2. Select **Allow external users to self-register**.
3. For page, select **Community Builder Page** and then select **Register**.
4. For Profile, select **Customers**. This setting automatically gives new users your External Identity user profile.
5. For Account, select **Customers**. You already created this account as part of preparing your org. When you're done with the page, it looks like this.



The screenshot shows the 'Registration' configuration interface. At the top, there is a checkbox labeled 'Allow external users to self-register' which is checked. Below this, the instruction 'Choose a self-registration page.' is followed by a 'Page' dropdown menu set to 'Community Builder Page' and a search box containing 'Register'. Underneath, the instruction 'Assign registering users to:' is followed by a 'Profile' dropdown menu set to 'Customers' and an 'Account' dropdown menu also set to 'Customers'.

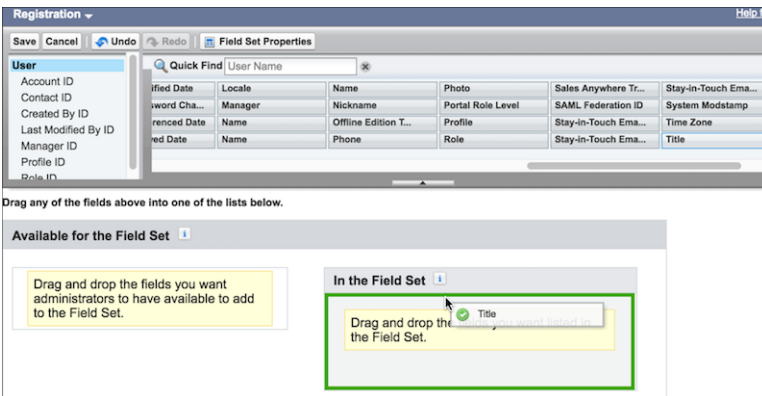
6. Click **Save**.
7. Return to the browser and reload the login page for your community. Click **Not a Member?** to register for the community.

Add Fields to Collect Additional Information

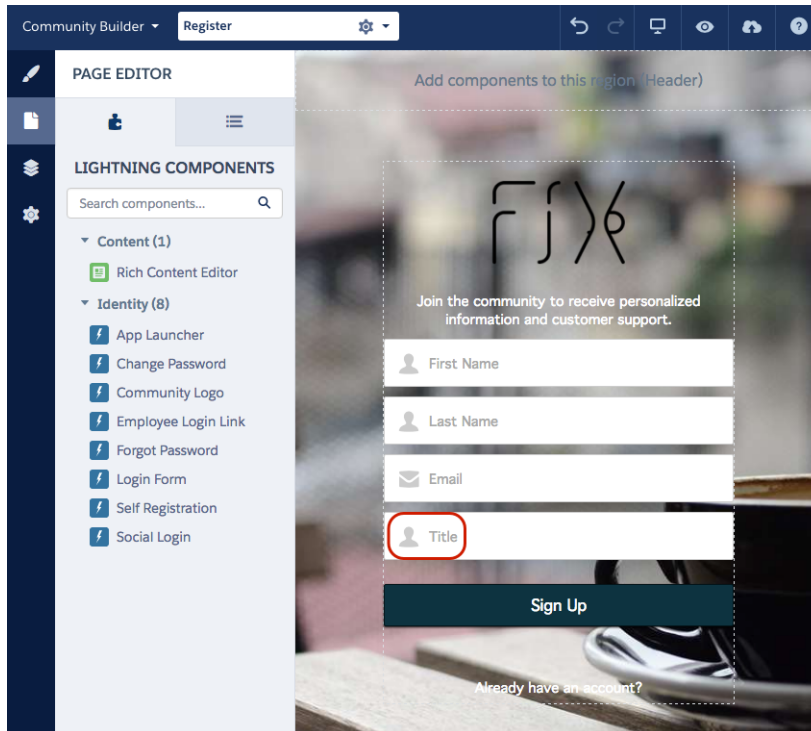
When users register, you often want to ask them for more than just basic information. You can easily add fields to the registration page.

Tailoring your registration page involves navigating to a few different areas in the app. First, watch the [Enabling Self-Registration in a Community](#) video. Then follow these steps.

1. In Setup, enter *Users* in the **Quick Find** box, then under **Customize Users**, select **Field Sets**.
2. Click **New** to create a field set. Name it *Registration*.
3. Under **Where is this used**, enter *reg field set*.
4. Drag **Title** into the field set, and click **Save**.



5. Go back to Community Builder. From Setup, enter *All Communities* in the **Quick Find** box, then click **Builder** next to your community.
6. From the dropdown menu at the top, select **Register**, and then select the **Page Editor**.
7. Select the Page Layout icon (☰), and then select **Self Registration**.
8. On the right, scroll to **Extra Fields Field Set Name** and enter *Registration*. The form reloads and displays your title field.

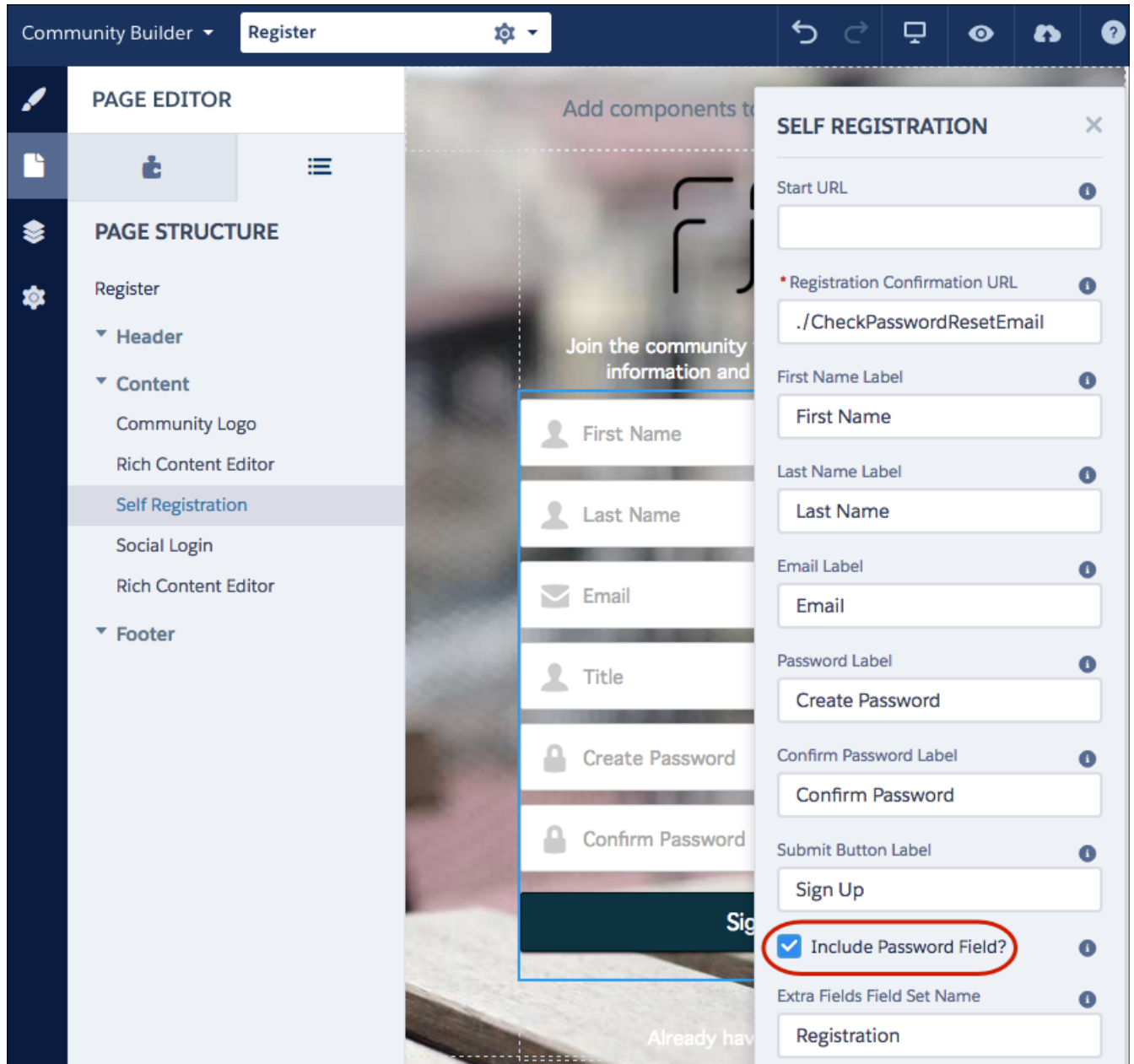


9. Click **Publish**.

Add a Password Field to Enable Login Directly During Registration

You can add a password field to your self-registration form to require users to supply a password when they register. Because you're already in Community Builder, it's simple to add the field.

1. From the Community Builder property manager, select **Include Password Field**.
The self-registration form reloads and displays the password field.



2. Click **Publish**.
3. After you receive the confirmation email, go back to your browser and check that your self-registration form includes the password field.

ENABLE SELF-REGISTRATION FOR B2C USERS (OPTIONAL)

Previously, you enabled self-registration for users in a simple business-to-business (B2B) data model. Each contact was associated with a default account called Customers. You can modify this process to support multiple accounts or even support a business-to-consumer (B2C) data model.

Salesforce supports a B2C model through person accounts. The best way to get started with person accounts is to review the [Setting Up Person Accounts Implementation Guide](#).

You can also watch the [Setting up Person Accounts and Enabling Them for Self-Registration in a Community](#) video. It walks you through setting up and enabling person accounts for self-registration.

[Enable Person Accounts](#)

Before you can use person accounts, you must enable them. It's easy to enable them, but it involves filing a case with Salesforce. Let's walk through it.

[Configure Self-Registration for Person Accounts](#)

To use person accounts instead of business accounts for self-registration, follow these steps.

Enable Person Accounts

Before you can use person accounts, you must enable them. It's easy to enable them, but it involves filing a case with Salesforce. Let's walk through it.

1. Create a record type for business accounts.

You create a record type for business account. You don't create a record type for person account because Salesforce creates it when it enables person accounts.

- a. From Setup, enter *Accounts* in the *Quick Find* box, then select **Record Types**.
- b. Click **New**.
- c. For Record Type Label, enter *Business Account*, then click **Next**.
- d. For page layout, select **Account Layout** as the layout to apply to all profiles.
- e. Click **Save**.

2. Verify your sharing settings.

- a. From Setup, enter *Security Controls* in the *Quick Find* box, then select **Sharing Settings**.
- b. Make sure that Contact is set to **Controlled by Parent**.

3. Contact Salesforce Customer Support to enable person accounts.

Configure Self-Registration for Person Accounts

To use person accounts instead of business accounts for self-registration, follow these steps.

1. Assign record types to your community's security profile by updating your community's public-access settings. This step ensures that the security profile that controls anonymous access in your community has access to account record types.

- a. From Setup, enter *All Communities* in the **Quick Find** box, then select **All Communities** and click **Manage** next to the Customers community.
 - b. Select **Administration**, then **Pages**, then select **Go to Force.com**.
 - c. Click **Public Access Settings**.
 - d. Under Record Type Settings, click **Edit** next to Accounts.
 - e. Select business and person record types and add them to Selected Record Types.
 - f. Click **Save**.
2. Update the self-registration setting on your login configuration page to use person accounts.
 - a. From Setup, enter *All Communities* in the **Quick Find** box, then select **All Communities** and click **Manage** next to the Customers community.
 - b. Select **Administration**, and then select **Login & Registration**.
 - c. Scroll down to Registration and make sure that the default Account field is empty. By removing the default, new users are created as person accounts.
 - d. Click **Save**.

You're done. New users that register with your branded self-registration form are now B2C-style users using person accounts.

ACCEPT IDENTITY FROM AN EXISTING IDENTITY PROVIDER

While self-registration is a great way to get started, often users exist in your back-office systems or with social providers, such as Facebook, LinkedIn, or Twitter. Salesforce Identity lets you use these existing sources with single sign-on (SSO) and just-in-time (JIT) provisioning. SSO and JIT provisioning let you create and update user accounts on the fly.

The following methods are available for SSO into Salesforce and communities.

Social sign-on

Salesforce users can authenticate and log in from different social identity providers, such as a Twitter and Facebook. They can also log in through open federation standards like OpenID Connect.

Federated authentication

Use Security Assertion Markup Language (SAML) to send authentication and authorization data between affiliated but unrelated web services.

Delegated authentication

Integrate Salesforce with various legacy authentication technologies.

Both federated authentication and social sign-on let you accept identities from existing identity providers and create users or link to and update existing users. Social sign-on is a common and effective way to engage your customers without having them create accounts.

[Social Sign-On](#)

Salesforce Identity supports a variety of public authentication providers, such as LinkedIn, Google, Facebook, Twitter, and open-standard OpenID Connect through the auth. providers framework. Using these providers, you can accept identity and link to existing Salesforce users. You can also create and update users on the fly using identity information asserted by the provider.

Social Sign-On

Salesforce Identity supports a variety of public authentication providers, such as LinkedIn, Google, Facebook, Twitter, and open-standard OpenID Connect through the auth. providers framework. Using these providers, you can accept identity and link to existing Salesforce users. You can also create and update users on the fly using identity information asserted by the provider.

The video [Setting Up Social Sign-On](#) walks you through setting up social sign-on. To get started, the following steps help you set up social sign-on with Facebook. The process is similar for all providers, so if you don't use Facebook, you can easily substitute another provider.

[Create an Auth. Provider](#)

You choose which Auth. providers can access your Salesforce org from Setup. With a few clicks, you can add the option to log in with one or more social accounts. Here's how to set up Facebook as an Auth. provider.

[Customize Your Registration Handler](#)

The registration handler is an Apex class that handles the heavy lifting of creating users, updating users, and linking to existing users, accounts, and contacts. You can also integrate more business processes, such as creating opportunities or calling out to back-office customer systems.

[Enable Your Auth. Provider in Your Community](#)

You created an Auth. provider for Facebook and customized it with a registration handler. Now instruct the login page in your community to display Facebook as an option on your external identity community's login page.

Create an Auth. Provider

You choose which Auth. providers can access your Salesforce org from Setup. With a few clicks, you can add the option to log in with one or more social accounts. Here's how to set up Facebook as an Auth. provider.

1. In your developer org, from Setup, enter *Auth. Providers* in the *Quick Find* box, then select **Auth. Providers**.
2. Click **New** and select **Facebook** for the provider type.
3. Name the Auth. provider *Facebook* and enter the URL suffix.
4. For this exercise, leave the Consumer Key, Consumer Secret, User Info Endpoint URL, and Default Scopes fields empty. When you leave these fields empty, Salesforce Identity uses a default application when interacting with Facebook. You can't customize the brand that users see nor the scope of access you request from the provider. In a real deployment, you register an application with the provider and configure your own consumer key (*client_id*) and consumer secret (*client_secret*).
5. For Registration Handler, select **Automatically create a registration handler template**.
6. For Execute Registration As, choose your admin user as the registration handler. Heads up: This step is essential and often gets overlooked.
7. For Icon URL, select **Choose one of our sample icons**.
8. In the new window, find a Facebook icon that you want to use, click it, and copy the URL.
9. Close the window and paste the URL as your Icon URL.
10. Click **Save**.

Customize Your Registration Handler

The registration handler is an Apex class that handles the heavy lifting of creating users, updating users, and linking to existing users, accounts, and contacts. You can also integrate more business processes, such as creating opportunities or calling out to back-office customer systems.

You can edit the generated registration handler. Or to get started, use one of our open-source samples.

1. In another browser window, open the registration handler, <https://github.com/salesforceidentity/IdentityTrail-Module3/blob/master/SimpleFacebookRegistrationHandler.cls>. This class creates an account and contact, and it also creates an opportunity during user creation.
2. Click **raw** and copy the code.
3. Return to your Auth. provider and click **AutoGeneratedRegHandler**.
4. Click **Edit**.
5. Select all the code and paste it over the old code.
6. Click **Save**.

You now have a fully functional Auth. provider that's ready for social sign-on with Facebook.

Enable Your Auth. Provider in Your Community

You created an Auth. provider for Facebook and customized it with a registration handler. Now instruct the login page in your community to display Facebook as an option on your external identity community's login page.

1. From Setup, enter *All Communities* in the *Quick Find* box, then select **All Communities** and click **Manage** next to your community.

2. Select **Administration**, then select **Login & Registration** and confirm that Facebook shows up in the checkbox under Login.
3. Click **Save**.
4. Test your changes by going to your community in a new browser or incognito window.
5. Reload the login page.
6. Click the Facebook logo.
7. Log in with your Facebook account.

You are immediately granted access to the community. If you return to the browser where you are administering Salesforce, go to Accounts and drill into the Customer account. You find that you show up as a contact. When you view your contact, you see that you have an opportunity associated with the contact. The registration handler created the opportunity.

For more information on configuring social sign-on for various providers, see [Social Sign-On](#) in the Salesforce Technical Library. You can find more sample Apex classes that implement the RegistrationHandler interface on the GitHub repository, <https://github.com/salesforceidentity>.

ACCEPT USER IDENTITY WITH SAML AND JUST-IN-TIME PROVISIONING

Salesforce Identity lets you bring your own identity from standards-based systems using SAML. You can integrate with existing SAML identity providers, letting users access your community based on your own authentication systems.

We assume that you're already familiar with SAML authentication protocols and you know how to work with your identity provider to configure SSO for your company. For more information on setting up SSO, watch the see [Setting up Single Sign-On \(Salesforce Classic\)](#) video.

With just-in-time (JIT) provisioning, you can use a SAML assertion to create or update users on the fly as part of the SSO process. How? You can either pass a Salesforce-defined set of attributes in your SAML assertion or have Salesforce Identity adapt to existing third-party schemas using Apex provisioning handlers. For more information, see [Just-in-Time Provisioning for Communities in Salesforce Help](#).

SET UP SSO AND ACCESS FOR YOUR WEB APP

So far we've focused on establishing user identity in Salesforce Identity with self-registration, social sign-on, and branded login services. After you're managing users successfully, the next step is to use identity services like SAML, OpenID Connect, or OAuth engines to provide identity to other applications.

In this section, you set up single sign-on with a sample application using SAML. For an overview, watch the [How to Set Up Single Sign-on With a Sample Application Using SAML](#) video.

[Create a Connected App for Your Web App](#)

A connected app is an application that integrates with Salesforce Identity using APIs and identity services. Connected apps use standard identity protocols like SAML, OAuth, and OpenID Connect to authenticate, provide single sign-on, and provide tokens for use with Salesforce APIs.

[Create a Sample Service Provider on Heroku](#)

To implement single sign-on, you need an app that speaks SAML. We've prepared a free sample that gets you up and running quickly.

[Configure Salesforce Identity to Provide Identity for Your App](#)

Teach Salesforce Identity about the SAML configuration of your new app.

[Authorize Your Web Application](#)

The Salesforce Identity SAML identity provider understands your app via the connected app, but your users aren't authorized to access it. You still have to configure authorization.

[Configure Your App to Trust Salesforce Identity](#)

Even though you've described your sample app to Salesforce Identity, your app doesn't yet trust Salesforce to act as an identity provider. You must configure the app to accept SAML messages. This process is known as SAML metadata exchange.

[Personalize Your App with Custom Attributes](#)

You might notice that your app displays attributes of the user's identity. These attributes are shared through standard SAML attribute assertions, which is useful when you want to personalize the app by providing more information about the user.

[More About Single Sign-On for Your Web App](#)

You've learned the basics of acting as an identity provider for your web app. For more information, use the following resources.

Create a Connected App for Your Web App

A connected app is an application that integrates with Salesforce Identity using APIs and identity services. Connected apps use standard identity protocols like SAML, OAuth, and OpenID Connect to authenticate, provide single sign-on, and provide tokens for use with Salesforce APIs.

Let's create a SAML-based connected app that users can see and administrators can manage.

1. From Setup, enter *Apps* in the *Quick Find* box, then select **Apps** and scroll to **Connected Apps**.
2. Click **New**.
3. Give your app a name. Let's call it My SSO App.
4. For Contact Email, enter your email address.
5. Select **Choose one of our sample logos**.

6. In the new window, select a logo you like and copy the URL.
7. Close the window, and paste the URL in the Logo URL field in your Connected App window.
8. Click **Enable SAML**.

You now have the basics of a connected app in place, but you need to connect the app to something. Let's set up another app so you can establish trust between the app and Salesforce Identity for SSO.

Create a Sample Service Provider on Heroku

To implement single sign-on, you need an app that speaks SAML. We've prepared a free sample that gets you up and running quickly.

The sample app runs on Heroku. Heroku is a Salesforce App Cloud offering that provides platform as a service in a wide variety of languages. It also offers an amazing developer experience. As you see, deploying a new app can be as simple as clicking a button. If you don't have a Heroku account, sign up for free at [Heroku](https://heroku.com).

After you have a Heroku account, go to <https://toolbelt.heroku.com> and install the Heroku toolbelt. Then follow these steps.

1. In a new browser window, go to <https://github.com/salesforceidentity/heroku-identity-java>.
2. Click **Deploy to Heroku**. A new page in the Heroku Dashboard displays that clones the sample for you.
3. On the dashboard, you can optionally name the app.
4. Click **Deploy for Free**.
5. Heroku copies the app that you'll control. When the copy is complete, click **View**.
6. Click **Login**.
Because you haven't configured the app to trust Salesforce as an identity provider, you see instructions about how to set it up.

Configure Salesforce Identity to Provide Identity for Your App

Teach Salesforce Identity about the SAML configuration of your new app.

1. Copy the Start URL value on your apps page. (It's the same as the Entity ID and ACS URL for this particular app.)
2. Return to your connected app window.
3. Paste the Start URL value into the Start URL, Entity ID, and ACS URL fields.
4. Click **Save**.

You've now configured a connected app with metadata for your sample SAML service provider.

Authorize Your Web Application

The Salesforce Identity SAML identity provider understands your app via the connected app, but your users aren't authorized to access it. You still have to configure authorization.

1. In the connected app window, click **Manage**.
2. Scroll to the Profiles section and click **Manage Profiles**.
3. Choose your Customers and System Administrator profiles.
4. Click **Save**.

Anyone with the Customers or System Administrator profile can use SAML to access the app.

Configure Your App to Trust Salesforce Identity

Even though you've described your sample app to Salesforce Identity, your app doesn't yet trust Salesforce to act as an identity provider. You must configure the app to accept SAML messages. This process is known as SAML metadata exchange.

Just as you've provided metadata about the app to Salesforce, you have to provide metadata about Salesforce to the app. In practice, this process varies from app to app, but the fundamentals remain the same. You provide the app the unique name of the identity provider, the URLs where it runs, and a certificate to use to validate single sign-on messages from the identity provider.

Salesforce Identity exposes standard SAML metadata documents that can be downloaded or accessed via a URL. The sample app you deployed accepts metadata either way. Let's take the easy route and use the URL.

1. Access the SAML metadata through a URL.
 - a. Scroll to the SAML Login Information section and expand the section for your community.
 - b. Copy the Metadata Discovery Endpoint value.
 - c. On a command line, use Heroku toolbelt to update the configuration of the app as follows: `heroku config: set --app your_app_name SAML_METADATA=your metadata url`

You've now configured your sample service provider to trust your Salesforce Identity IDP.

2. Let's test it!
 - a. Return to your sample app and reload the page.
 - b. You're now automatically signed in as your administrator using SAML.
3. Test the configuration with a user.
 - a. In a new browser or incognito window, load your application.
 - b. Click **Login**.
 - c. Click the Facebook icon. If required, log in with Facebook.

You're automatically returned to your application via SSO.

Personalize Your App with Custom Attributes

You might notice that your app displays attributes of the user's identity. These attributes are shared through standard SAML attribute assertions, which is useful when you want to personalize the app by providing more information about the user.

Connected apps let you extend this information through custom attributes. Using custom attributes, you can enrich the data sent to your app declaratively, choosing from attributes of users, their profiles, and their Salesforce org. When the app interacts with Salesforce over SAML or OpenID Connect, these attributes are shared in a standardized way.

To enrich the attributes sent, follow these steps.

1. Go to the Connected Apps page for your app.
2. Scroll to Custom Attributes and click **New**.
3. Set the Attribute key to Profile.
4. Click **Insert Field**.
5. Click **\$Profile** and find Name.

6. Click **Insert**.
7. Return to your sample service provider and log out.
8. Click **Login** to get single sign-on, including your new attribute.

Custom attributes are flexible, and you can use the Salesforce formula language to combine or transform attributes for your particular use case. For example, you can create a custom attribute called "IsOver18" with a formula like this.

```
IF(( $User.Birthday__c - TODAY() + 6574 ) >= 0, 'false', 'true' )
```

At runtime, the attribute logic looks at a custom date field on the user object, calculates whether the user is over 18, and discloses true or false. This attribute allows you to assert that the user meets a business policy without disclosing the actual birthday to the target application.

For more information on using formulas, review the [Using Formula Fields](#) Trailhead module. You can also construct custom attributes using Apex.

More About Single Sign-On for Your Web App

You've learned the basics of acting as an identity provider for your web app. For more information, use the following resources.

[Setting up Single Sign-On \(Salesforce Classic\) video](#)

[Salesforce SSO How Tos](#) in Salesforce Technical Library

PROVIDE SSO AND ACCESS FOR MOBILE APPS

Salesforce Identity isn't limited to web applications. You can also use it to provide identity for mobile apps. Without much added work, you can use the Salesforce Mobile SDK to create mobile apps that integrate with everything you've set up.

You interface Salesforce Identity with mobile apps using the OAuth protocol. OAuth is an open standard used for authorization that provides applications secure, delegated access to services on behalf of a user without sharing the user's credentials. Fortunately, you don't need to know much about OAuth to use it. Salesforce Identity and the Salesforce Mobile SDK work together.

To walk through creating a mobile app, check out the video [How to Create a Sample Mobile App and Take Advantage of Salesforce Identity](#). Then follow these steps to create a sample mobile app. To get started, we create a connected app that supports OAuth. The process is similar to the SAML work you just completed.

[Create a Connected App for Your Mobile App](#)

The connected app integrates your mobile app with Salesforce Identity. This example assumes that you're using macOS and iOS. The steps for Android are similar, but they're not covered here.

[Install the Salesforce Mobile SDK](#)

Salesforce Mobile SDK is an open-source suite of familiar technologies (including a REST API and OAuth 2.0). You use the SDK to rapidly build HTML5, native, and hybrid mobile apps that connect to the Salesforce platform.

[Create a Mobile App](#)

Let's use the Salesforce Mobile SDK to jump-start our app.

[Configure the Mobile App to Point to Your Community](#)

Let's teach the mobile app about your community to finish the identity configuration.

[More About Single Sign-On for Your Mobile App](#)

You've learned the basics of acting as an identity provider for mobile apps. For more information, several Trailhead modules can guide you.

Create a Connected App for Your Mobile App

The connected app integrates your mobile app with Salesforce Identity. This example assumes that you're using macOS and iOS. The steps for Android are similar, but they're not covered here.

1. In your developer org, from Setup, enter `Apps` in the `Quick Find` box and then select **Apps**.
2. Scroll to Connected Apps and click **New**.
3. Enter a name for your app. Let's call it My Mobile App.
4. For Contact Email, enter your email address.
5. Click **Enable OAuth Settings**.
6. Enter a callback URL. Use `mymobileapp://callback`.
7. Select the `id`, `openid`, `api`, `refresh_token`, `web`, and `visualforce` scopes.
8. Click **Save**.
9. Click **Continue**.

Install the Salesforce Mobile SDK

Salesforce Mobile SDK is an open-source suite of familiar technologies (including a REST API and OAuth 2.0). You use the SDK to rapidly build HTML5, native, and hybrid mobile apps that connect to the Salesforce platform.

If you don't already have the Salesforce Mobile SDK, follow the installation instructions in the [Salesforce Mobile SDK Development Guide](#) to download it.

Create a Mobile App

Let's use the Salesforce Mobile SDK to jump-start our app.

1. At a command line, change to a directory where you want to create your app assets.
2. Run `forceios create`.
3. For application type, enter `native`.
4. For application name, enter `MyMobileApp`.
5. Press Enter to create the app in the current directory.
6. For package name, enter `com.yourcompany`.
7. For organization name, enter `YourCompany`.
8. Return to the Connected App page in your Developer org and copy the consumer key.
9. Paste the key in the forceios utility as the value for Connected App ID.
10. Return to the Connected App page in your Developer org and copy the callback URL.
11. Paste the URL in the forceios utility as the value for the Callback URI.
12. Press Enter.

The Mobile SDK creates a mobile app project for you.

Configure the Mobile App to Point to Your Community

Let's teach the mobile app about your community to finish the identity configuration.

1. At a command line, change to the app's directory by running `cd MyMobileApp`.
2. Open your app in XCode by running `open MyMobileApp.xcodeproj`.
3. In your Developer org, copy your community URL, omitting `https://`. If you don't recall the URL, from Setup, enter `Communities` and then select **All Communities**.
4. Return to XCode.
5. In the file browser, expand `MyMobileApp > Supporting Files`.
6. Click `MyMobileApp-info.plist`.
7. Select the `SFDCOAuthLoginHost` key value and replace `login.salesforce.com` with your community URL (once again, without `https://`).

Now all you have to do is build your app. Click the triangle-shaped button to build your app and watch it connect to your community. You can now log in to your app, even using social sign-on if you want. With Salesforce Identity, you focus on building your app rather than spending resources integrating identity.

More About Single Sign-On for Your Mobile App

You've learned the basics of acting as an identity provider for mobile apps. For more information, several Trailhead modules can guide you.

- [Salesforce Identity How-To](#) video series
- [Mobile Basics](#) Trailhead module
- [Native iOS](#) Trailhead module
- [Native Android](#) Trailhead module
- [HTML5 & Hybrid](#) Trailhead module

EXTERNAL IDENTITY ON GITHUB AND SUCCESS COMMUNITY

We've covered the basics of setting up Salesforce Identity for external users. What you've learned in this guide provides the basis for customizing Salesforce Identity to your specific business goals.

However, you can do much more with Salesforce Identity. For more information on customizing Salesforce Identity for your business, check out our advanced samples on the Salesforce Identity GitHub account.

If you don't see an answer to your question or a solution to your problem, post to the Salesforce Identity group in the Success Community. The Salesforce Identity Team loves to hear from customers.

INDEX

[\\$Profile](#) 28

A

[account](#) 3, 9
[activate community](#) 14
[Aloha template](#) 10
[Auth. provider](#) 23
[Auth. providers](#) 23

B

[B2C](#) 20
[business accounts](#) 20
[business-to-consumer](#) 20

C

[clone external identity profile](#) 8
[cloud directory services](#) 3
[communities](#) 6
[Community Builder](#) 13
[Community Management menu](#) 10
[connected app](#) 26, 30
[create developer org](#) 7
[CRM](#) 2–3, 9
[custom attributes](#) 28
[custom login page](#) 10

D

[dashboards](#) 4
[definition](#) 2
[delegated authentication](#) 22
[developer org](#) 7
[domain name](#) 7

E

[Entity ID](#) 27

F

[Facebook](#) 22–23, 28
[federated authentication](#) 22
[field sets](#) 16
[forceios utility](#) 31

G

[GitHub](#) 23, 33

H

[Heroku](#) 27

I

[IAM services](#) 1
[Identity Basics](#) 1
[Identity Connect](#) 5
[Identity only license](#) 5

J

[JIT](#) 22, 25
[just-in-time provisioning](#) 4, 22, 25

L

[licenses](#) 5
[login page](#) 13
[logo](#) 26

M

[membership to community](#) 11
[Metadata Discovery Endpoint](#) 28
[mobile identity](#) 3, 30
[multi-factor authentication](#) 4
[My Domain](#) 7

N

[Not a member link](#) 16

O

[OAuth](#) 30

P

[password on login page](#) 18
[person accounts](#) 20
[profiles](#) 2, 8–9, 11, 20, 27–28

R

[register domain](#) 7
[registration handler](#) 23
[roles](#) 8

S

[SAML](#) 27
[SAML-based connected app](#) 26
[sandboxes](#) 7, 18

Index

self-registration [2–3](#), [16](#), [20](#)
SFDCOAuthLoginHost [31](#)
single sign-on [3](#), [23](#)
start URL [27](#)
Success Community [33](#)

T

title field [16](#)
Trailhead [1](#)

U

use cases [4](#), [6](#), [20](#), [22](#)

W

workflows [3–4](#)

X

XCode [31](#)