# **Connected Apps**

A connected app is a framework that enables an external application to integrate with Salesforce using APIs and standard protocols, such as SAML, OAuth, and OpenID Connect. Connected apps use these protocols to authenticate, authorize, and provide single sign-on (SSO) for external apps. The external apps that are integrated with Salesforce can run on the customer success platform, other platforms, devices, or SaaS subscriptions. For example, when you log in to your Salesforce mobile app and see your data from your Salesforce org, you’re using a connected app.

By capturing metadata about an external app, a connected app tells Salesforce which authentication protocol—SAML, OAuth, and OpenID Connect—the external app uses, and where the external app runs. Salesforce can then grant the external app access to its data, and attach policies that define access restrictions, such as when the app’s access expires. Salesforce can also audit connected app usage.

Connected apps are available in your Salesforce orgs and Experience Cloud sites. By default, your org and your Experience Cloud users can access connected apps. But you can restrict user access to a connected app with permissions and policies.

# **Connected App Use Cases**

There are four main use cases for which your org can implement connected apps. You can use a connected app to integrate external applications with the Salesforce API, such as a web-based app that pulls in order status data from your Salesforce org. You can also use connected apps to integrate service providers with your Salesforce org, and to set security policies to control what data a third-party app can access from your org. And you can configure a connected app to provide authorization for external API gateways, such as API gateways hosted on MuleSoft’s Anypoint Platform.

## **1. Access Data with API Integration**

You can use a connected app to request access to Salesforce data on behalf of an external application. For a connected app to request access, it must be integrated with the Salesforce API using the OAuth 2.0 protocol. OAuth 2.0 is an open protocol that authorizes secure data sharing between applications through the exchange of tokens. Developers and independent software vendors (ISVs) use OAuth authorization flows to integrate their app with the Salesforce API. These authorization flows enable a user to work in one app but see the data from another.

## **2.Integrate Service Providers with Your Salesforce Org**

When Salesforce acts as your identity provider, you can use a connected app to integrate your service provider with your Salesforce org. Use one of these methods to configure a connected app for a service provider.

You can use a connected app with SAML 2.0 to integrate a service provider with your Salesforce org. Salesforce supports SAML single sign-on (SSO) when the service provider or the identity provider initiates the flow.

For example, you build a custom Your Benefits web app that implements SAML 2.0 for user authentication. You want your users to be able to log in to this app with their Salesforce credentials. To set up this SSO flow, configure the Your Benefits web app as a connected app. Define your Salesforce org as the SAML identity provider for the connected app. Your users can now log in to the Your Benefits web app with their Salesforce credentials.

## **3.Manage Access to Third-Party Apps**

Admins can set security policies to control what data a third-party app can access from your org. Admins can also define who can use the third-party app.

For example, you install a third-party app that allows your org’s users to make travel reservations. By selecting the option “Admin approved users are pre-authorized” for the connected app, you can assign specific user profiles to the app. Only the users with this user profile can access the app. You can also set a refresh token policy to revoke the travel reservation app’s access to your Salesforce data after a set amount of time.

In addition to setting security policies to manage third-party apps, you can uninstall, and—when necessary—block these apps from the Salesforce org.

## **4.Provide Authorization for External API Gateways**

Salesforce can act as an independent OAuth authorization server to protect resources hosted on an external API gateway. Using OpenID Connect dynamic client registration, resource servers can dynamically create client apps as connected apps in Salesforce. Salesforce can then authorize these connected apps to access protected resources hosted by the third-party service.

For example, Salesforce can act as the OAuth authorization server for API gateways that are hosted on MuleSoft's Anypoint Platform. MuleSoft’s Anypoint Platform, which is the resource server, can dynamically create client apps as connected apps. These connected apps can send a request to Salesforce asking for access to data protected by the API gateways. Salesforce can then authorize the connected apps, granting them access to the data protected by the API gateways.