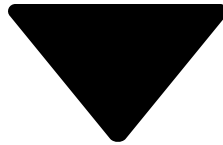


1. [Apex Integration Services](#)



2. [Apex Integration Overview](#)

Apex Integration Overview



Note

Attention, Trailblazer!

Salesforce has two different desktop user interfaces: Lightning Experience and Salesforce Classic. This module is designed for **Salesforce Classic**.

You can learn about switching between interfaces, enabling Lightning Experience, and more in the [Lightning Experience Basics](#) module here on Trailhead.

Learning Objectives

After completing this module, you will be able to:

- Describe the differences between web service and HTTP callouts.
- Authorize an external site with remote site settings.

Make Callouts to External Services from Apex

An Apex callout enables you to tightly integrate your Apex code with an external service. The callout makes a call to an external web service or sends an HTTP request from Apex code, and then receives the response.

Apex callouts come in two flavors.

- Web service callouts to SOAP web services use XML, and typically require a WSDL document for code generation.
- HTTP callouts to services typically use REST with JSON.

These two types of callouts are similar in terms of sending a request to a service and receiving a response. But while WSDL-based callouts apply to SOAP Web services, HTTP callouts can be used with any HTTP service, either SOAP or REST.

So you are probably asking yourself right now, “Which one should I use?” Whenever possible, use an HTTP service. These services are typically easier to interact with, require much less code, and utilize easily readable JSON. All the “cool kids” have been switching to REST services over the last couple of years, but that’s not to say that SOAP Web services are bad. They’ve been around forever (in Internet years) and are commonly used for enterprise applications. They are not going away anytime soon. You’ll probably use SOAP mostly when integrating with legacy applications or for transactions that require a formal exchange format or stateful operations. In this module we’ll touch on SOAP, but will spend most of our time on REST.

Authorize Endpoint Addresses

We love security at Salesforce! So, any time you make a callout to an external site we want to make sure that it is authorized. We can’t have code calling out willy-nilly to any endpoint without prior approval. Before you start working with callouts, update the list of approved sites for your org on the Remote Site Settings page.

We’ll be using the following endpoints in this module, so go ahead and add them now. If you forget to add an endpoint, trust me, you’ll get a reminder when you try to run your code. We’ll be making calls to the following sites.

- <https://th-apex-http-callout.herokuapp.com>
- <https://th-apex-soap-service.herokuapp.com>

Authorize both of these endpoint URLs by following these steps.

1. From Setup, enter `Remote Site Settings` in the Quick Find box, then click **Remote Site Settings**.
2. Click **New Remote Site**.
3. For the remote site name, enter `animals_http`.
4. For the remote site URL, enter `https://th-apex-http-callout.herokuapp.com`. This URL authorizes all subfolders for the endpoint, like `https://th-apex-http-callout.herokuapp.com/path1` and `https://th-apex-http-callout.herokuapp.com/path2`.
5. For the description, enter `Trailhead animal service: HTTP`.
6. Click **Save & New**.
7. For the second remote site name, enter `animals_soap`.
8. For the remote site URL, enter `https://th-apex-soap-service.herokuapp.com`.
9. For the description, enter `Trailhead animal service: SOAP`.
10. Click **Save**.

Resources

- [REST vs SOAP: When Is REST Better?](#)

Quiz Complete!

+100 points



Apex Integration Services

100%

Progress: 100%

[View more modules](#)