

# Web Services, SUDS, and REST

## Web Services Overview

Web services are software solutions that allow for interacting with machines residing on a network. In short, web services are nothing more than web pages for machines. They provide a standard way for a third party to request and receive data from a piece of hardware on the network without having to know anything about how that machine works.

## Protocols

There are two common approaches to Web Services in Ignition: making a HTTP (HyperText Transfer Protocol) method call, or making a SOAP (Simple Object Access Protocol) call.

- HTTP methods, such as GET and POST, are called using the built-in system functions, such as `system.net.httpGet()`. More information on HTTP calls can be found on the [HTTP Methods](#) page.
- SOAP is XML based, and typically requires a third party Python library. There are [many third party Python libraries that utilize SOAP](#), however, documenting them goes outside the scope of this manual (as do all third party libraries).

The approach you choose typically depends on the server you're trying to make calls to: more specifically, the protocol(s) it supports.

### What About the SUDS Library?

The SUDS library, a library that used to come included with the Python Standard Library, offered SOAP based functionality. However, SUDS development has been halted, and is no longer included in the standard library.

In the interest of posterity, the legacy SUDS documentation has been condensed and can be found on the [SUDS - Library Overview](#) page. Note that the legacy documentation should be considered deprecated.

### On this page

...

- [Web Services Overview](#)
- [Protocols](#)
- [Can Ignition Make RESTful Calls?](#)
- [Common Web Services Workflow](#)

## Can Ignition Make RESTful Calls?

Yes it can! However, it is important to understand that REST is an architecture, **NOT** a protocol. Instead, REST utilizes and describes how a protocol should be used. Thus, a RESTful architecture could use both of the protocols mentioned on this page, although HTTP is far more common.

## Common Web Services Workflow

While all Web Services follow the same standards, they all do different things. They wouldn't be worth anything if you didn't get the information you need, or if they contained a lot of excess data. If you are unfamiliar with a particular Web Service, there are a few things that you can do to figure out what data is available and how to get it.

1. Identify a Web Service that you will be using. Usually the Web Service has an API somewhere documenting how requests should be made.
2. Write a script to pull some information from the Web Service. If using HTTP, this could mean starting with a GET call, where as SOAP would involve retrieving the WSDL (Web Services Description Language). In both cases, you may need to find a way to authenticate against the server (usually with some user credentials or an auth token, the API for the service would have more details).
3. Once you have the results from the GET/WSDL, identify the information or functions you want to use.
4. Write a script to use that function and return your values.
5. Parse the results and use them. This can be for display, saving to a database, or anything else you need.

Note: Web Services sometimes take a lot of time to return results, especially the first time they are called. If you put your Web Services script in a button, the client will freeze until the call is complete (this is because the event handlers are run on the GUI thread). It's a good idea to use `system.util.invokeAsynchronous()` or add a waiting image to your screen to let the user know Ignition is working as expected.

[Related Topics ...](#)

- WebDev Module

In This Section ...