

Diving In

Philosophically, I can describe HTTP web services in 12 words: exchanging data with remote servers using nothing but the operations of HTTP. If you want to get data from the server, use HTTP `GET`. If you want to send new data to the server, use HTTP `POST`. Some more advanced HTTP web service apis also allow creating, modifying, and deleting data, using HTTP `PUT` and HTTP `DELETE`. That's it. No registries, no envelopes, no wrappers, no tunneling. The “verbs” built into the HTTP protocol (`GET`, `POST`, `PUT`, and `DELETE`) map directly to application-level operations for retrieving, creating, modifying, and deleting data.

The main advantage of this approach is simplicity, and its simplicity has proven popular. Data — usually `XML` or `JSON` — can be built and stored statically, or generated dynamically by a server-side script, and all major programming languages (including Python, of course!) include an HTTP library for downloading it. Debugging is also easier; because each resource in an HTTP web service has a unique address (in the form of a URL), you can load it in your web browser and immediately see the raw data.

Examples of HTTP web services:

- [Google Data APIs](#) allow you to interact with a wide variety of Google services, including [Blogger](#) and [YouTube](#).
- [Flickr Services](#) allow you to upload and download photos from [Flickr](#).
- [Twitter API](#) allows you to publish status updates on [Twitter](#).
- [...and many more](#)

Python 3 comes with two different libraries for interacting with HTTP web services:

- [http.client](#) is a low-level library that implements [RFC 2616](#), the HTTP protocol.

- `urllib.request` is an abstraction layer built on top of `http.client`. It provides a standard API for accessing both HTTP and FTP servers, automatically follows http redirects, and handles some common forms of HTTP authentication.

So which one should you use? Neither of them. Instead, you should use `httplib2`, an open source third-party library that implements HTTP more fully than `http.client` but provides a better abstraction than `urllib.request`.

To understand why `httplib2` is the right choice, you first need to understand HTTP.