

Creating a New API or Provider

Adding a new provider is quite an important change for jclouds. The first thing you'll want to do is review [How to Contribute](#). We need to make sure it will work properly and we'll be able to properly test it so don't be discouraged by the reviews. A good example of a JIRA issue for tracking the development of a new API and providers is [Implement OpenStack Marconi and Rackspace Cloud Queues](#).

Having said this, a number of general considerations to take into account when starting a new API or provider:

1. New providers are added in labs ([labs](#), [aws](#), [openstack](#)) first, until they are stable. For more information on how APIs/providers are considered for promotion out of labs, see [How Does an API/Provider Get Out of Labs?](#)
2. We use Guava as our core library. Try to use its utilities before adding new dependencies. In general, do not add dependencies before consulting our dev@ list.
3. We require both live and mock tests. "mock" tests verify that the api generates the expected request according to the method annotations, and that the response is properly parsed. We use MockWebServer to do that, and you can take a look at the MockTests classes of an existing api.
4. Often the best thing to do is to copy the entire directory structure of one of the existing apis or providers and delete everything that you don't need but keep some of the files around in each directory to serve as an example of how to do things. You can rename those classes and start your work from there.

See [the API/Provider Writing Practices](#) for more detailed information on code style and other guidelines.

Which API or provider should you choose to use as an example for your work? It depends! Please email us on our dev@ list to describe the work your doing and we will recommend one. The earlier you get engaged with the jclouds community, the easier the whole process will be.