**Jenkins** [**Remote access with**](https://wiki.jenkins-ci.org/display/JENKINS/Remote+access+API) **REST API**

Jenkins provides machine-consumable remote access API to its functionalities. Currently it comes in three flavors:

1. XML
2. JSON with JSONP support
3. Python

Remote access API is offered in a REST-like style. That is, there is no single entry point for all features, and instead they are available under the ".../api/" URL where "..." portion is the data that it acts on.

## Remote API can be used to do things like these:

1. retrieve information from Jenkins for programmatic consumption.
2. trigger a new build
3. create/copy jobs

## To access the api

<http://localhost:8080/job/myjob1/api>

http://localhost:8080/job/myjob/149/api/json

<http://localhost:8080/job/myjob1/api>/json?tree=duration

Open Jenkins on your webbrowser, and notice that on the right bottom there is REST API link. Click it. This link appears on every single page of Jenkins, and points you to an API output for the exact page you are browsing.

## Submitting jobs

A build can be started just by POSTing to

http://server/job/myjob/buildWithParameters?PARAMETER=Value

**Jobs without parameters**

You merely need to perform an HTTP POST on ﻿JENKINS\_URL/job/JOBNAME/build?token=TOKEN where TOKEN is set up in the job configuration.

**Jobs with parameters**

Simple example - sending "String Parameters":

curl -X POST JENKINS\_URL/job/JOB\_NAME/build \

--data token=TOKEN \

--data-urlencode json='{"parameter": [{"name":"id", "value":"123"}, {"name":"verbosity", "value":"high"}]}'

Another example - sending a "File Parameter":

curl -X POST JENKINS\_URL/job/JOB\_NAME/build \

--user USER:PASSWORD \

--form file0=@PATH\_TO\_FILE \

--form json='{"parameter": [{"name":"FILE\_LOCATION\_AS\_SET\_IN\_JENKINS", "file":"file0"}]}'

E.g.curl -X POST http://JENKINS\_URL/job/JOB\_NAME/build --form file0=@/home/user/Desktop/sample.xml --form json='{"parameter": [{"name":"harness/Task.xml", "file":"file0"}]}'

Please note, in this example, the symbol '@' is important to mention. Also, the path to the file is absolute path.

## Remote API and security

When your Jenkins is secured, you can use HTTP BASIC authentication to authenticate remote API requests. See [Authenticating scripted clients](https://wiki.jenkins-ci.org/display/JENKINS/Authenticating+scripted+clients) for more details.

In order to make this command work, you need to configure your Jenkins job to take a file parameter and 'name' in this command corresponds to 'file location' field in the Jenkins job configuration.

In above example, 'harness' is just a name of folder that may not exist in your workspace, you can write "name":"Task.xml" and it will place the Task.xml at root of your workspace.

Remember that name in this command should match with File location in file parameter in job configuration.

**Jenkins Java REST Client**