Jenkins REST API is a web-based API that allows you to interact with Jenkins programmatically using HTTP requests. The REST API provides a way to retrieve and manipulate various types of data in Jenkins, such as jobs, builds, nodes, plugins, and more.

Using the REST API, you can perform a variety of tasks, including:

- Triggering builds: You can use the REST API to start a build of a specific job in Jenkins or to trigger a parameterized build with specific parameters.
- Retrieving information: You can retrieve information about jobs, builds, nodes, and other objects in Jenkins using the REST API. This information can include details such as the status of a build, the console output of a build, or the configuration of a job.
- Managing plugins: You can use the REST API to install, update, or remove plugins in Jenkins.
- Modifying Jenkins configurations: You can update Jenkins configurations using the REST API. For example, you can modify job configurations, add or remove nodes, or modify global settings.

Jenkins provides a REST API that allows you to interact with the Jenkins server programmatically. The REST API can be used to retrieve information about Jenkins jobs, nodes, views, plugins, and more, as well as trigger builds and configure Jenkins.

To use the Jenkins REST API, you need to send HTTP requests to the Jenkins server. The most common HTTP methods used with the Jenkins REST API are GET, POST, PUT, and DELETE. The response to a request is usually in JSON or XML format.

Here are some examples of how to use the Jenkins REST API:

- To get a list of all jobs in Jenkins, you can send a GET request to the following URL:

http://<jenkins-url>/api/json

This will return a JSON object containing information about all the jobs in Jenkins.

- To trigger a build of a specific job, you can send a POST request to the following URL:

http://<jenkins-url>/job/<job-name>/build

Replace **<job-name>** with the name of the job you want to trigger a build for. You can also include additional parameters in the request body to pass to the job.

- To retrieve information about a specific job, you can send a GET request to the following URL:

http://<jenkins-url>/job/<job-name>/api/json

This will return a JSON object containing information about the job, including its name, status, last build number, and more.

- To retrieve information about a specific build of a job, you can send a GET request to the following URL:

http://<jenkins-url>/job/<job-name>/<build-number>/api/json

Replace **<build-number>** with the build number you want to retrieve information for. This will return a JSON object containing information about the build, including its status, duration, console output, and more.

These are just a few examples of what you can do with the Jenkins REST API. For more information on how to use the REST API, including how to authenticate and use the API in different programming languages, you can refer to the official Jenkins documentation.

To trigger a build of a specific job in Jenkins using the REST API, you can send a POST request to the following URL

http://<jenkins-url>/job/<job-name>/build

To retrieve a list of all the installed plugins, you can send a GET request to the following URL:

http://<jenkins-url>/pluginManager/api/json?depth=1

1. To install a new plugin, you can send a POST request to the following URL:

http://<jenkins-url>/pluginManager/installNecessaryPlugins

In the body of the POST request, you should include a **form-data** field named **json** with the following JSON data:

{ "installPlugins": "<plugin-name>:<version>" }

Replace **<plugin-name>** with the name of the plugin you want to install and **<version>** with the version of the plugin you want to install. If you want to install multiple plugins at once, you can separate them with commas, like this:

{ "installPlugins": "<plugin1-name>:<version1>,<plugin2-name>:<version2>" }

2. To update an existing plugin, you can send a POST request to the following URL:

http://<jenkins-url>/pluginManager/api/json/postPluginManager/batchInstall

In the body of the POST request, you should include a **form-data** field named **json** with the following JSON data:

{ "plugins": [{ "name": "<plugin-name>", "source": { "update": true } }] }

Replace **<plugin-name>** with the name of the plugin you want to update. If the plugin has updates available, Jenkins will download and install the latest version of the plugin.

3. To remove an existing plugin, you can send a POST request to the following URL:

http://<jenkins-url>/pluginManager/api/json/postPluginManager/plugin/<plugin-name>/doUninstall

Replace **<plugin-name>** with the name of the plugin you want to remove.