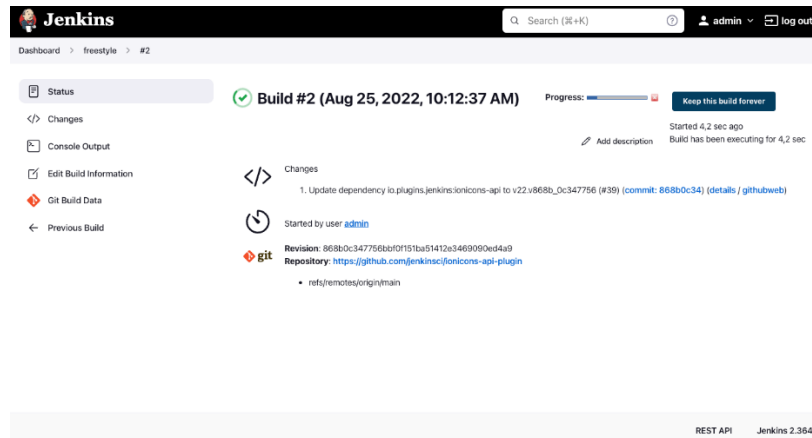
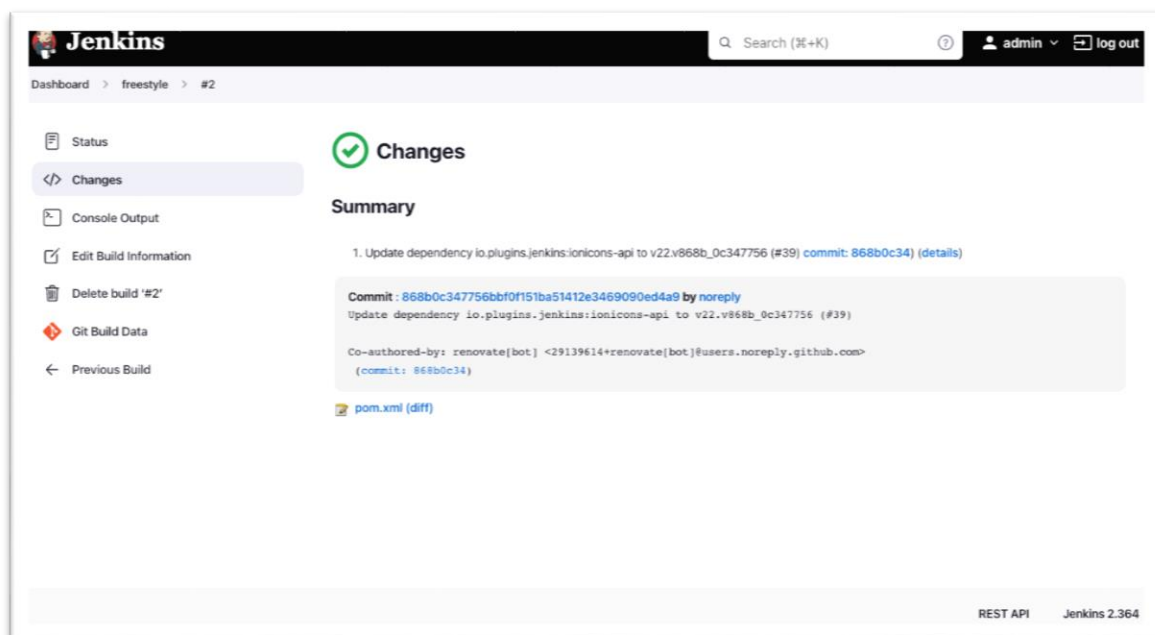


Jenkins : Web-hook

Web hooks are an essential component of continuous integration and continuous delivery (CI/CD) systems like Jenkins.



This screenshot shows the Jenkins web interface for a specific build. The top navigation bar includes the Jenkins logo, a search bar, and user information (admin, log out). The breadcrumb trail indicates the path: Dashboard > freestyle > #2. On the left, a sidebar contains links to Status, Changes, Console Output, Edit Build Information, Git Build Data, and Previous Build. The main content area displays 'Build #2 (Aug 25, 2022, 10:12:37 AM)' with a green checkmark icon. It shows a progress bar, a 'Keep this build forever' button, and a timestamp 'Started 4.2 sec ago'. Below this, it lists changes: '1. Update dependency io.plugins.jenkins:ionicons-api to v22.v868b_0c347756 (#39) (commit: 868b0c34) (details) / githubweb'. It also notes 'Started by user admin' and provides git details: 'Revision: 868b0c347756bbf0f151ba51412e3469090ed4a9' and 'Repository: https://github.com/jenkinsci/ionicons-api-plugin'. A footer bar at the bottom right shows 'REST API' and 'Jenkins 2.364'.



This screenshot shows the Jenkins 'Changes' page. The top navigation bar is identical to the previous screenshot. The breadcrumb trail is 'Dashboard > freestyle > #2'. The left sidebar now highlights 'Changes' and includes a 'Delete build '#2'' option. The main content area features a green checkmark icon and the title 'Changes'. Below the title is a 'Summary' section with the text: '1. Update dependency io.plugins.jenkins:ionicons-api to v22.v868b_0c347756 (#39) (commit: 868b0c34) (details)'. A commit box displays: 'Commit : 868b0c347756bbf0f151ba51412e3469090ed4a9 by noreply', 'Update dependency io.plugins.jenkins:ionicons-api to v22.v868b_0c347756 (#39)', and 'Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com> (commit: 868b0c34)'. A link to 'pom.xml (diff)' is provided. The footer bar at the bottom right shows 'REST API' and 'Jenkins 2.364'.

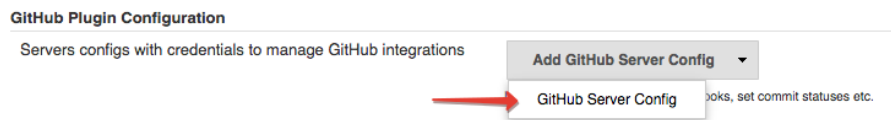
GitHub hook trigger for GITScm polling :

This feature enables builds after [post-receive hooks in your GitHub repositories](#). This trigger only kicks git-plugin internal polling algo for every incoming event against matched repo.

Manual Mode

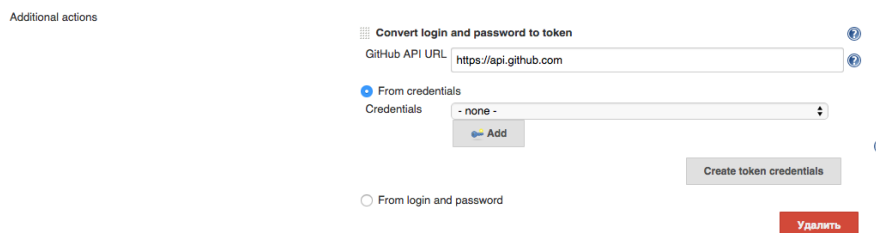
In this mode, you'll be responsible for registering the hook URLs to GitHub. Click the icon (under Manage Jenkins > Configure System > GitHub) to see the URL in Jenkins that receives the post-commit POSTs — but in general the URL is of the form

Step 1. Go to the global configuration and add GitHub Server Config.

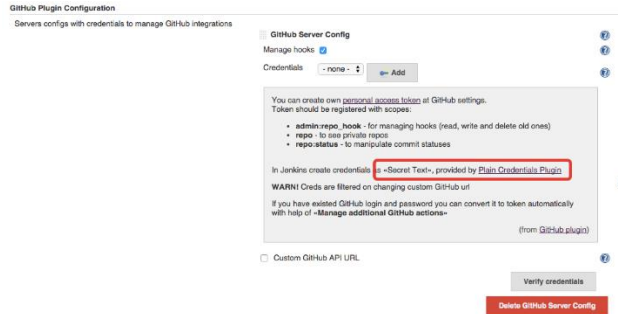


Step 2.1. Create your personal access token in GitHub.

Plugin can help you to do it with all required scopes. Go to **Advanced** -> **Manage Additional GitHub Actions** -> **Convert Login and Password to token**



Step 2.2.: **Select previously created "Secret Text" credentials with GitHub OAuth tokTo be able manage hooks your token should have admin:org_hook scope.**



Step 3. Once that configuration is done, go to the project config of each job you want triggered automatically and simply check "GitHub hook trigger for GIT Scm polling" under "Build Triggers".

Security Implications

This plugin requires that you have an HTTP URL reachable from GitHub, which means it's reachable from the whole internet. So it is implemented carefully with the possible malicious fake post-receive POSTS in mind.

Jenkins inside a firewall:

In case your Jenkins run inside the firewall and not directly reachable from the internet, this plugin lets you specify an arbitrary endpoint URL as an override in the automatic mode. The plugin will assume that you've set up reverse proxy or some other means so that the POST from GitHub will be routed to the Jenkins.