COURSEWARE

- Professional Skills
- Agile Fundamentals
- Jira
- Git
- Databases Introduction
- Java Beginner
- Maven
- Testing (Foundation)
- Java Intermediate
- HTML
- CSS
- Javascript
- Spring Boot
- Selenium
- Sonarqube
- Advanced Testing (Theory)
- Cucumber
- MongoDB
- Express
- NodeJS
- React
- Express-Testing
- Networking
- Security
- Cloud Fundamentals
- AWS Foundations
- AWS Intermediate
- Linux
- DevOps

Plugins

Contents

- Overview
- Why Plugins?
- Tutorial
- Exercises

Overview

Jenkins has a **powerful extension and plugin system** that allows developers to write plugins affecting nearly every aspect of Jenkins' behaviour.

Even a large part of the "core" functionality of Jenkins uses plugins, with some of the most popular ones being installed on set-up (should you choose to "install suggested plugins").

Why Plugins?

Some of the reasons why we might want to use plugins include:

- **Source Code Management (SCM) implementations**: integrating SCMs like Git into Jenkins.
- **Build steps**: providing everything from a convenient UI to configure build tools, to sending emails post-build.
- **Authentication realms**: integrating Jenkins with Single Sign-On systems, or external user directories (such as Azure Active Directory).

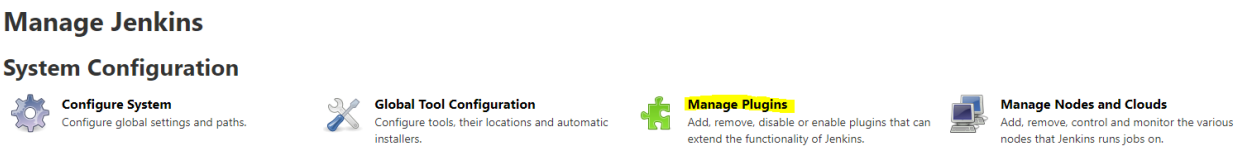
And much, much more!

Tutorial

In this tutorial, we will look at installing a plug in, and seeing what effect that has on our Jenkins instance.

Before starting this tutorial, make sure you have a Jenkins instance up and running.

1. Go to `Manage Jenkins` and click on `Manage Plugins`:



2. Click on the Available Tab and search "maven". Check `Maven Integration` and select `Install without restart`:

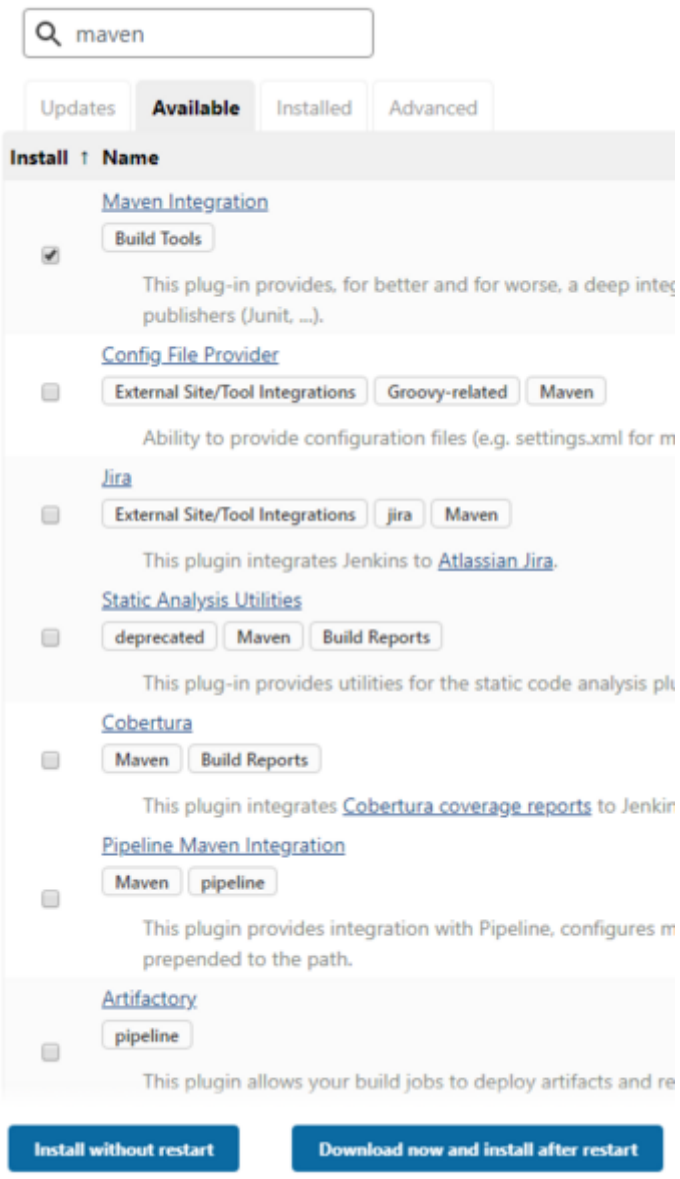
Jenkins Introduction

- Installation Wizard
- Jobs
- Freestyle Project
- Builds
- Plugins

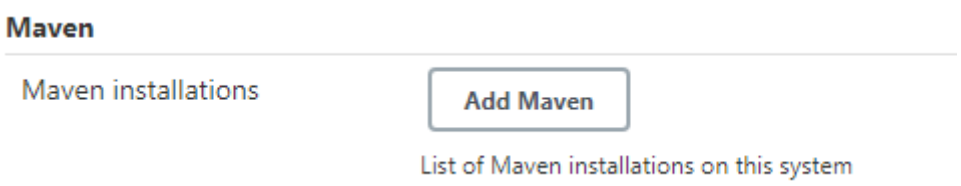
Jenkins Pipeline

Markdown

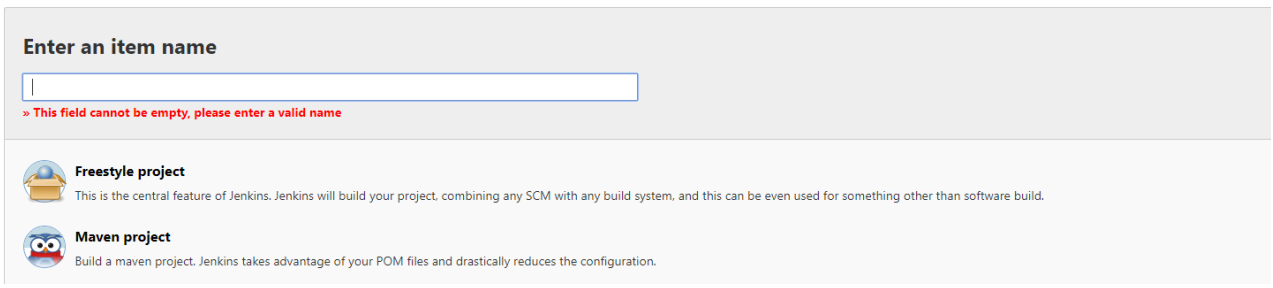
IDE Cheatsheet



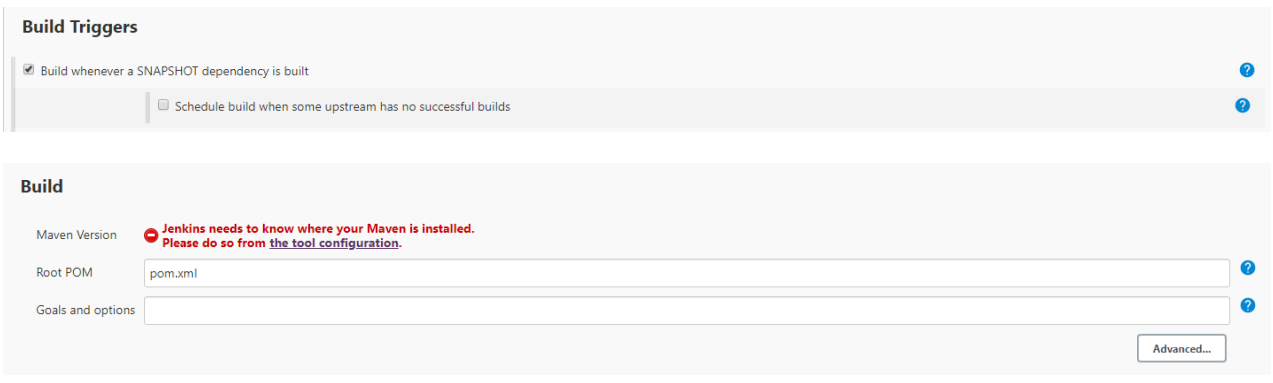
3. In `Manage Jenkins` > `Global Tool Configuration`, you will now see an option to configure the path to the Maven file:



4. If you create a `New item`, you can now see the `Maven project` option:

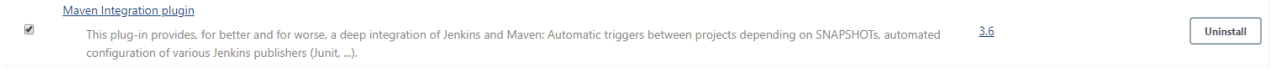


5. Inside this project, there are some options available that make integrating with Maven much easier:



The Maven plugin we installed allows much easier integration with Maven, and makes it simple to configure; this is the power of plugins!

To uninstall a plugin, you simply go back to `Manage Plugins` and go to the `Installed` tab:



Exercises

Take a look at this link to browse the available plugins for Jenkins.