Git & GitHub, Maven & Gradle

Cristian Camilo Serna Betancur

Andres Grisales Gonzalez

September 9, 2020

Contents

1 Git

Git is a *free and open source* distributed version control system designed to handle everything from small to very large projects with speed and efficiency [2]. It was created by Linus Torvalds in 2005 for development of the *Linux kernel*, with other kernel developers contributing to its initial development. Since 2005, Junio Hamano has been the core maintainer[3].

It allows us to read the history of the project, its changes, and the thoughts of the person who worked in them [1]. If something is wrong, we can use it to return at the time where all was working well. With Git we can also tag a specific state of our project with a version number if we think some work is done, reliable, and ready to use.

---

[1]Although sometimes we don't write good commit messages, sometimes we don't use more than one line to communicate neither our thoughts nor the purpose of our changes.

2   Github

GitHub, Inc. is an American multinational corporation that provides hosting for software development and version control using Git[4]. It has many features such as pull request (PR), bug tracking, task manager, and wikis for every project. It is one of the most important hosting providers for free and open source projects, since it hosts very important ones.

With GitHub we can improve our own projects and help on different ones, giving an receiving feedback, report issues and share knowledge, and ideas. We can use hosted continous integration services such as Travis CI with it that enables us to test and ship our apps with confidence[8].

Git + GitHub offer us an easy way to work with many people at the same project [2], does'nt matter if even we are in different countries, we can share our code and ideas with everyone in the world.

3   Maven

Maven is a build automation tool used primarily for Java projects. Maven can also be used to build and manage projects written in Java, C#, Ruby, Scala, and other languages[7]. It is an excellent tool for Java developers and it is also possible to use it to manage the lifecycle of your projects.

As a life cycle management tool, Maven works around phases. It handles all phases of the project lifecycle, including validation, code generation, compilation, testing, packaging, integration testing, verification, installation and deployment[3].

In Maven a Project Object Model or POM is the fundamental unit of work in Maven. It is an XML file that contains information about the project and configuration details used by Maven to build the project. It contains default values for most projects[1].

One of Maven's most useful resources is its dependency management support: you simply define the libraries on which your application depends and Maven locates them (either in your local repository or in a central repository), downloads them, and uses them to compile your code[4].

---

[2]Since I've learned how to use Git and GitHub (GitHub or GitLab or Bitbucket, who cares) I've not been able to figure a different way to work with more than one person at the same project.

[3]Since the beginning of my university studies, I have always worked with maven, that is the system with which I have felt most comfortable; Maven is simple.

[4]You need to have an internet connection. Maven the first time you run start downloading things from the internet and if that can't, it's useless.

4   Gradle

Gradle is a build automation tool for multi-language software development. It controls the development process in the tasks of compilation and packaging to testing, deployment, and publishing[5][5]. Gradle is highly customizable and fast as it completes tasks quickly and accurately by reusing outputs from previous runs, only processing inputs that have changes in parallel.

Some features of Gradle that we can highlight are the following: Collaborative debugging, incremental construction, custom repository design, transitive dependencies, built-in Groovy and Scala support, incremental compilation for Java, packaging and distribution of JAR, WAR and EAR, daemon compiler.

Gradle allows you to build from microservices to mobile applications, it can be used by small startups as well as large companies, as it helps teams develop, automate and deliver quality software in a shorter time (depending on its complexity and prior planning of the development process)[5].

5   Maven vs Gradle

Both build systems offer built-in ability to resolve configurable repository dependencies and can cache dependencies locally and download them in parallel. In the world of Java programming, the leader in software development for a considerable time has been Maven. Today, the time has come to leave behind the endless configuration XML files and make way for something more modern and powerful like Gradle. Gradle is more powerful. However, there are times that you really do not need most of the features and functionalities it offers. Maven might be best for small projects, while Gradle is best for bigger projects[6].

---

[5]Gradle build scripts are written using Groovy or Kotlin DSL (Domain Specific Language). Gradle has great flexibility and allows us to make uses of other languages and not just Java, it also has a very stable dependency management system.

References

[1] *Apache Maven - Wikipedia*. URL: https://en.wikipedia.org/wiki/Apache_Maven.

[2] *Git*. URL: https://git-scm.com/.

[3] *Git - Wikipedia*. URL: https://en.wikipedia.org/wiki/Git.

[4] *GitHub - Wikipedia*. URL: https://en.wikipedia.org/wiki/GitHub.

[5] *Gradle Build Tool*. URL: https://gradle.org/.

[6] *Gradle vs. Maven - DZone Java*. URL: https://dzone.com/articles/gradle-vs-maven#:~:text=Gradle%20is%20based%20on%20a,things%20that%20do%20the%20work.%E2%80%9D.

[7] *Maven – Welcome to Apache Maven*. URL: https://maven.apache.org/.

[8] *Travis CI · GitHub Marketplace*. URL: https://github.com/marketplace/travis-ci.