

بسم الله الرحمن الرحيم

مروری بر چرخه زندگی یک پروژه Maven

استاد پورسلطانی

سیده فاطمه موسوی

تابستان ۱۴۰۳

دانشگاه صنعتی شریف

فهرست

3مقدمه
3Maven چیست؟
3Maven چگونه کار می کند؟
4دستورات مورد استفاده در Maven
4mvn clean
4mvn compile
4mvn test
4mvn package
5mvn install
5mvn deploy
5جمع بندی
6منابع

مقدمه

دنیای برنامه نویسی دنیای پیچیده ای است، هرچه در این دنیا بیشتر غرق شوید بیشتر دانش کسب می کنید. در این تحقیق به زبانی ساده در مورد Maven صحبت می کنیم و توضیح می دهیم که Maven چیست، چگونه کار می کند و قرار است در چه بخشی کاربرد داشته باشد.

Maven چیست؟

Maven یک نرم افزار مدیریت پروژه و ابزار درک بر اساس مفهوم Project Object Model (POM) است. Maven ساخت پروسه ها را ساده می کند و وابستگی ها، گزارش ها، مستندات و نسخه های پروژه را مدیریت می کند. Maven به زبان جاوا نوشته شده است و می توان از آن برای ساخت پروژه در زبان های برنامه نویسی مختلف استفاده کرد، اما بیشتر برای پروژه های مبتنی بر جاوا استفاده می شود. ویژگی های کلیدی Maven عبارتند از:

- مدیریت وابستگی: Maven دانلود و مدیریت وابستگی های پروژه را انجام می دهد و اطمینان حاصل می کند که از نسخه های ثابت در سراسر پروژه استفاده می شود.
- Maven: Build Automation یک چرخه عمر ساخت استاندارد شده با مراحل از پیش تعریف شده برای کامپایل، آزمایش، بسته بندی و استقرار کد ارائه می دهد.
- ساختار پروژه: این ابزار یک ساختار استاندارد پروژه را اعمال می کند و درک و هدایت پروژه را برای توسعه دهندگان آسان تر می کند.
- گزارش و مستندسازی: ابزار Maven می تواند گزارش های مختلفی مانند نتایج تست واحد، پوشش کد و اطلاعات وابستگی، شفافیت و اسناد پروژه را بهبود بخشد.
- پروژه های چند ماژول: Maven از مدیریت پروژه های چند ماژول پشتیبانی می کند و به توسعه دهندگان این امکان را می دهد تا روی چندین پروژه مرتبط به طور همزمان کار کنند.

Maven چگونه کار می کند؟

POM: فایل پیکربندی مرکزی پروژه Maven فایل pom.xml است که ساختار پروژه، وابستگی ها، فرآیند ساخت و سایر تنظیمات را تعریف می کند.

Maven: Build Lifecycle دارای یک چرخه عمر ساخت از پیش تعریف شده با چندین مرحله مانند کامپایل، تست، بسته بندی و نصب است. این مراحل را می توان به صورت جداگانه یا متوالی اجرا کرد تا فرآیند ساخت را خودکار کند.

مدیریت وابستگی: Maven وابستگی های پروژه را با دانلود و مدیریت کتابخانه ها و چارچوب های مورد نیاز از مخازن راه دور مانند Maven Central Repository مدیریت می کند.

پلاگین ها: Maven از پلاگین ها برای انجام وظایف خاص مانند کامپایل کد، اجرای آزمایش ها و تولید گزارش استفاده می کند. توسعه دهندگان همچنین می توانند پلاگین های سفارشی برای گسترش عملکرد Maven ایجاد کنند.

نمایه‌های ساخت: Maven از استفاده از نمایه‌های ساخت پشتیبانی می‌کند، که به توسعه‌دهندگان اجازه می‌دهد تنظیمات مختلف ساخت را برای محیط‌های مختلف (مانند توسعه، آزمایش، تولید) پیکربندی کنند.

دستورات مورد استفاده در Maven

mvn clean

این دستور پوشه **target** را که محل ذخیره‌سازی خروجی‌های ساخته شده توسط **Maven** است را پاک می‌کند. این دستور معمولاً قبل از ساخت یک بسته جدید اجرا می‌شود تا مطمئن شویم از آخرین نسخه ساخته شده استفاده می‌کنیم. این دستور به صورت خودکار پوشه **target** را حذف می‌کند، به طوری که در هر بار اجرای فرآیند ساخت، شما مطمئن خواهید بود که از آخرین نسخه خروجی استفاده می‌شود. این روش به ویژه زمانی مفید است که در یک پروژه بزرگ کار می‌کنید و می‌خواهید مطمئن شوید که هیچ قطعه قدیمی در خروجی باقی نمانده است.

mvn compile

این دستور کد منبع برنامه را کامپایل می‌کند. این دستور کد منبع را بررسی کرده و فایل‌های کلاس را در پوشه **target** ایجاد می‌کند. این دستور معمولاً به عنوان اولین قدم در فرآیند ساخت برنامه اجرا می‌شود.

این دستور ابتدا پیکربندی پروژه را بررسی می‌کند و سپس به کامپایل کد منبع می‌پردازد. اگر در طول کامپایل هر گونه خطایی رخ دهد، **Maven** خطاها را شناسایی کرده و گزارش می‌دهد. این مرحله بسیار مهم است زیرا قبل از اجرای تست‌ها و بسته‌بندی برنامه، باید مطمئن شد که کد منبع بدون هیچ گونه خطا کامپایل می‌شود.

mvn test

این دستور تست‌های واحد برنامه را اجرا می‌کند **Maven**. از پلاگین **Surefire** برای اجرای تست‌ها استفاده می‌کند. این دستور ابتدا کد را کامپایل کرده و سپس تست‌ها را اجرا می‌کند. نتایج تست‌ها در پوشه **target/surefire-reports** ذخیره می‌شود.

این دستور ابتدا کد را کامپایل می‌کند تا مطمئن شود که هیچ خطایی در کد وجود ندارد. سپس تست‌های واحد را اجرا می‌کند و نتایج را در پوشه **target/surefire-reports** ذخیره می‌کند. این مرحله بسیار مهم است زیرا آزمایش عملکرد برنامه قبل از بسته‌بندی و انتشار آن ضروری است.

mvn package

این دستور برنامه را بسته‌بندی می‌کند. بسته‌بندی می‌تواند به انواع مختلفی مانند **JAR**، **WAR**، **EAR** یا شکل‌های دیگر باشد که به نوع پروژه بستگی دارد. این دستور ابتدا کد را کامپایل کرده و تست‌ها را اجرا می‌کند و در صورت موفقیت، برنامه را بسته‌بندی می‌کند.

این دستور ابتدا کد را کامپایل کرده و تست‌های واحد را اجرا می‌کند. سپس برنامه را بسته‌بندی کرده و فایل بسته‌بندی شده را در پوشه **target** ذخیره می‌کند. این مرحله بسیار مهم است زیرا محصول نهایی برای انتشار و استفاده آماده می‌شود.

mvn install

این دستور بسته‌بندی شده را در مخزن محلی Maven نصب می‌کند. این به معنای ذخیره‌سازی بسته در مخزن محلی است تا در صورت نیاز بتوان از آن استفاده کرد.

این دستور ابتدا کد را کامپایل کرده، تست‌ها را اجرا می‌کند و سپس بسته‌بندی شده را در مخزن محلی Maven نصب می‌کند. این مرحله به سایر پروژه‌هایی که از این پروژه به عنوان وابستگی استفاده می‌کنند این امکان را می‌دهد که بتوانند از این بسته استفاده کنند.

mvn deploy

این دستور ابتدا بسته را ساخته و سپس آن را در مخزن راه‌دور قرار می‌دهد. این به سایر توسعه‌دهندگان امکان می‌دهد تا از این بسته در پروژه‌های خود استفاده کنند.

جمع‌بندی

Maven یک ابزار مدیریت پروژه و ساخت است که برای توسعه نرم‌افزارهای جاوا و زبان‌های مشابه استفاده می‌شود. این ابزار امکاناتی مانند مدیریت وابستگی‌ها، ساخت خودکار، مدیریت عملیات جانبی، مدیریت تنظیمات پروژه و انتشار را فراهم می‌کند.

Maven با استفاده از فایل pom.xml تنظیمات پروژه را مدیریت می‌کند. با اضافه کردن وابستگی‌ها به فایل pom.xml، می‌توان آن‌ها را به‌طور خودکار دریافت و مدیریت کرد. همچنین، Maven قابلیت ساخت خودکار پروژه را دارد و با دستور mvn clean install، مراحل ساخت را انجام می‌دهد و خروجی نهایی را تولید می‌کند.

در ضمن، Maven می‌تواند عملیات جانبی مرتبط با توسعه نرم‌افزار را مدیریت کند و عملیاتی مانند کامپایل، تست، بسته‌بندی، نصب و انتشار را به صورت خودکار انجام دهد. همچنین، این ابزار امکان مدیریت تنظیمات پروژه را فراهم می‌کند و قابلیت انتشار خروجی نهایی را دارد.

بنابراین، Maven به توسعه‌دهندگان کمک می‌کند تا فرآیند ساخت و مدیریت پروژه را بهبود بخشند و باعث سهولت و بهینه‌سازی فرآیند توسعه نرم‌افزار شود.

- [1] <https://www.geeksforgeeks.org/apache-maven/>
- [2] <https://www.tutorialspoint.com/maven/index.htm>
- [3] <https://www.javatpoint.com/maven-tutorial>
- [4] <https://maven.apache.org/plugins/maven-deploy-plugin/>
- [5] <https://maven.apache.org/plugins/maven-install-plugin/>
- [6] <https://maven.apache.org/plugins/maven-assembly-plugin/>
- [7] <https://maven.apache.org/surefire/maven-surefire-plugin/>
- [8] <https://maven.apache.org/plugins/maven-compiler-plugin/>
- [9] <https://maven.apache.org/plugins/maven-clean-plugin/>
- [10] <https://maven.apache.org/plugins/maven-deploy-plugin/usage.html>