

Aim: Using GRADDLE for CICD Pipeline

Description:

GRADLE BUILD TOOL:



Gradle is Build Tool for Java artifacts construction with many features supporting enterprise solutions

- Incremental Builds
- Build Caching
- Subtask strategies
- Annotation processing
- Compilation of Artifacts
- Continuous Build with Dry Runs

Dependency Management in Gradle:

Substitution of Compatible Libraries:

Use dependency substitution rules to identify that dependency should be treated as similar. Tell Gradle that only one should be selected and use Gradle conflict resolution to pick the newest version from both of them. Similar use cases are situations where you have libraries like spring-all and spring-core in dependency graph. Without properly modeling this the proper behavior of your application depends on the very fragile order in your class path.

Enhanced Metadata Resolution Support:

Dependency metadata can be modified after repository metadata is downloading but before it is chosen by Gradle as the final resolved version. This allows the creation of custom rules to do things like declare modules as changing go on.

Replacement of external and project dependencies:

Dynamically replace external dependencies for project dependencies and vice versa. Especially helpful when only a subset of your modules are checked out locally.

Resolved dependency versions can be dynamically:

Gradle supports the Maven snapshot mechanism but is more powerful than that. You can declare a dependency on the latest release, most current development version, or even the latest 5.X build.

Dynamic Dependency Locking:

Allow builds to remain deterministic and reproducible when using dynamic dependency versions.

Dynamic Dependencies Selection Rules:

Define custom rules to select a specific version when a dynamic dependency is declared. The rules can be based on names and version but also extended metadata like branch or status. The rules can also differ based on the environment the build is happening, e.g. local or CI.

Dependency Version Alignment:

Dependency alignment allows different modules in a logical group. Like JSON Modules aligned by versions.

Maven and Ivy Repository Compatible:

Gradle is compatible with the POM & IVY Metadata formats and can retrieve dependencies from any Maven or IVY compatible repository.

Native BOM support:

Platform definitions, aka Maven BOM dependencies are natively supported, allowing importing things like the Spring Boot platform definition without using an external plug-in.

Installing Gradle:

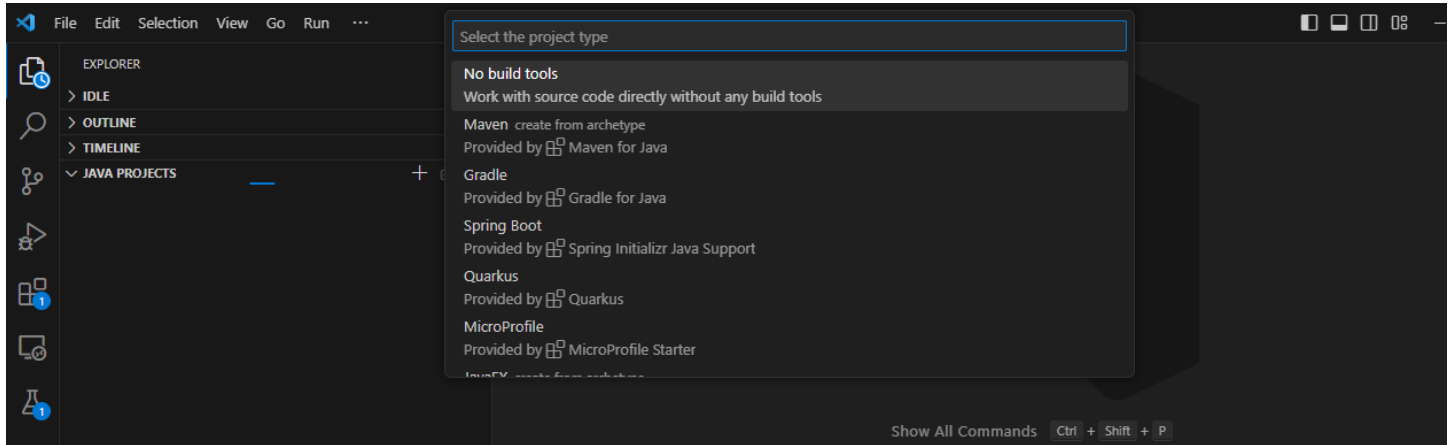
Open Visual Studio Code

Select Extensions Tab and type 'Gradle' in search bar

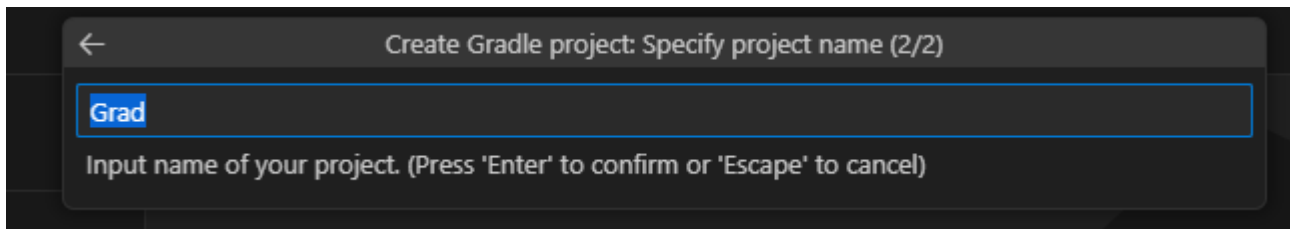
List of recommendations appear choose the 'Gradle Extension pack'

Press <Install> to your VS-Code

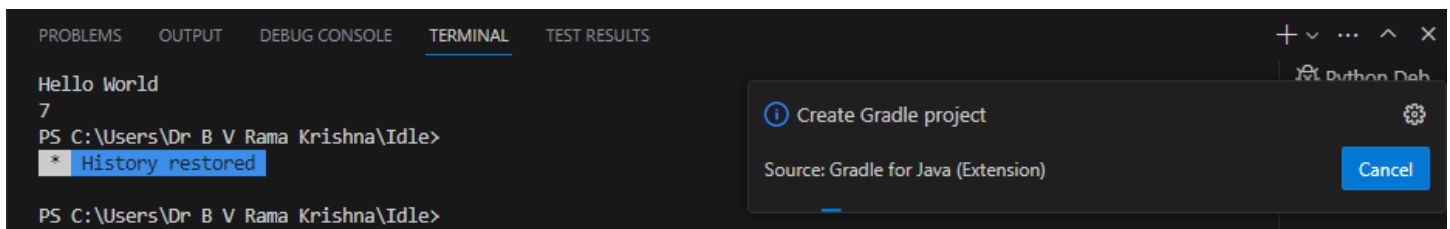
Now Go to JAVA PROJECTS Tab to verify Gradle Installation



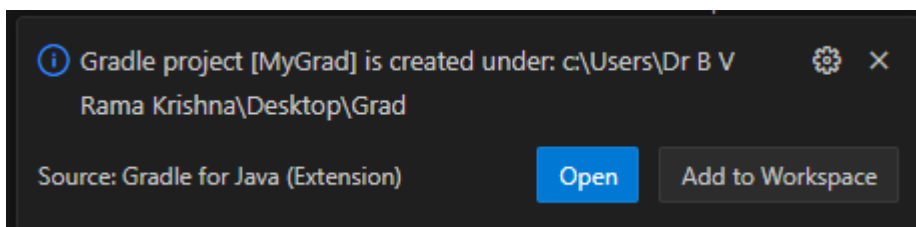
Looks Gradle installed in VS Code press Gradle and choose local directory JAVA Class Path must be correct in windows to work Gradle correctly.



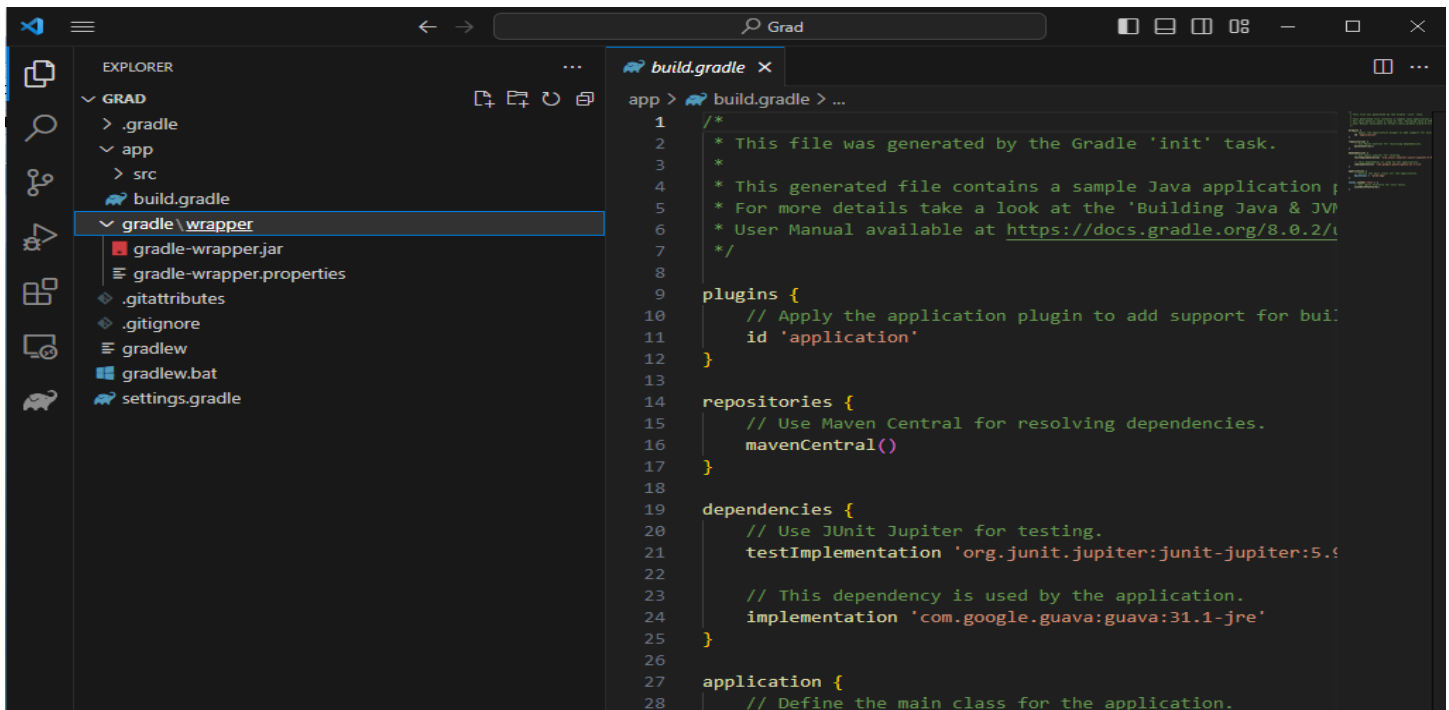
Select 'Groovy' build approach and press <Enter>



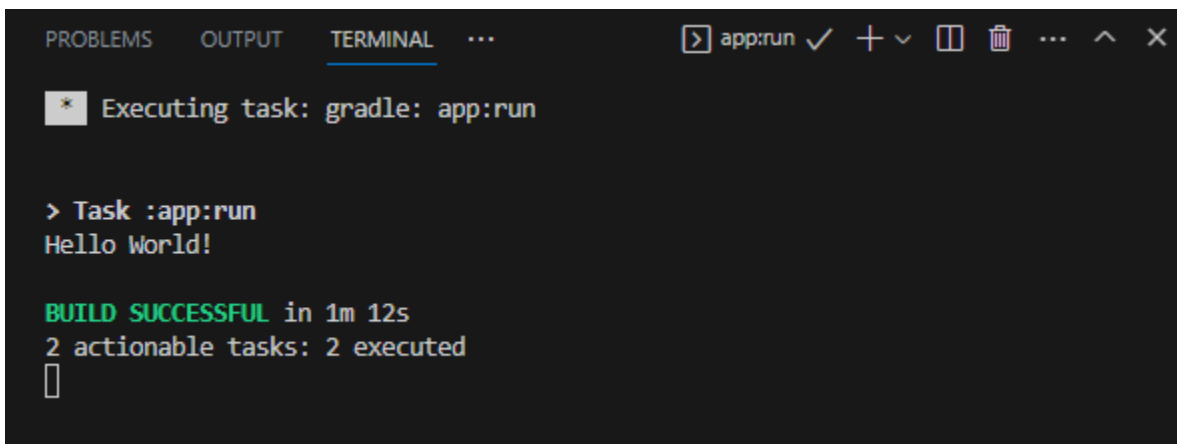
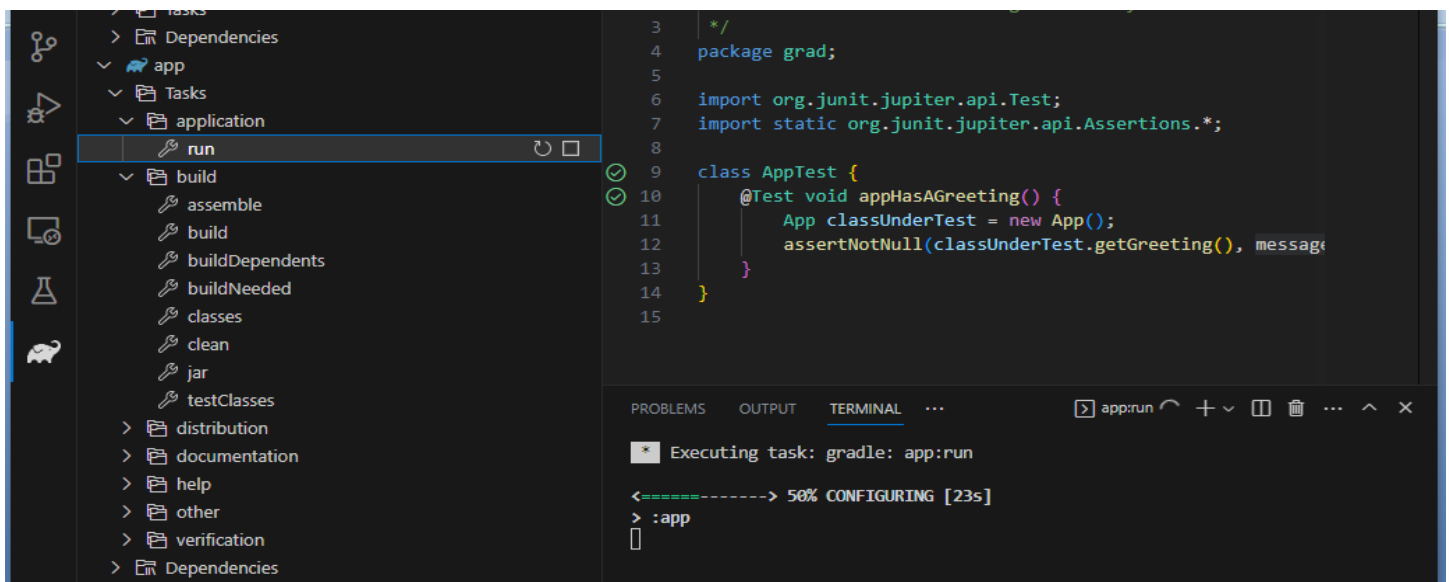
GRADLE Project construction begins for JAVA



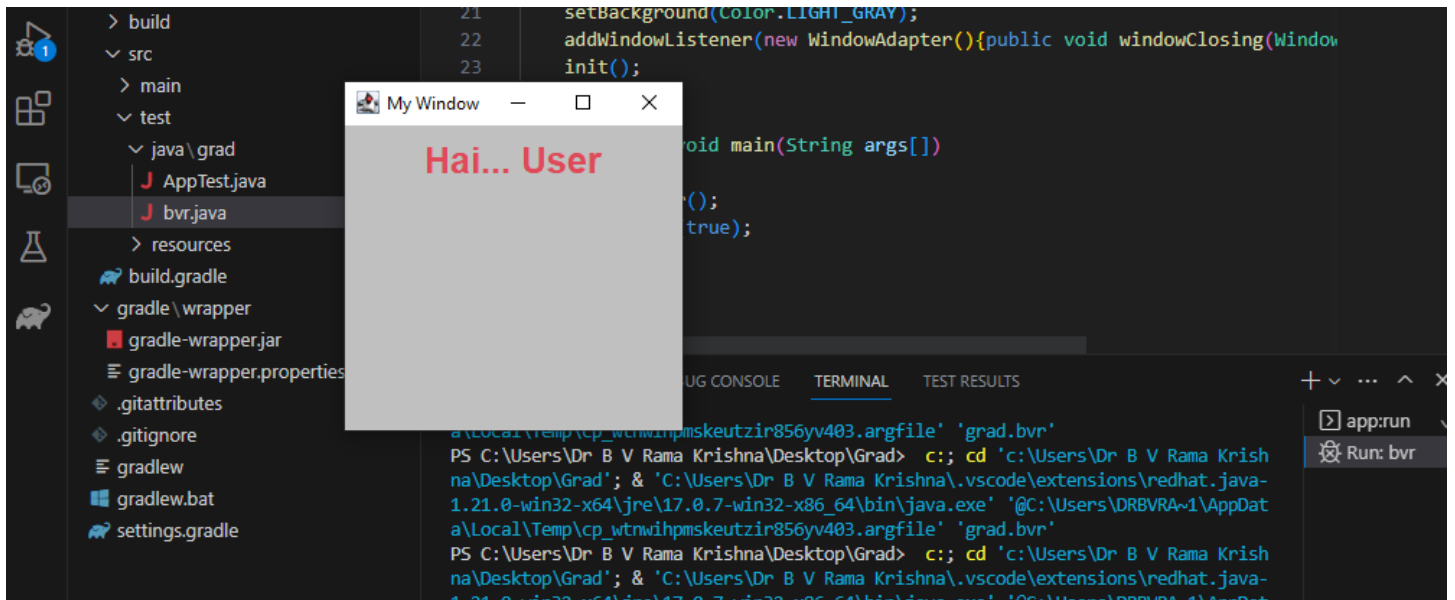
Project created successfully under workspace with dependencies...Press Open button to open workspace
Now you can see all dependency , main, src, wrapper and libraries are added to workspace for JRE support artifact construction.



Working with Build in App tab press 'Run' to build artifact



GRADLE running another app to wish through Http local server using builded artifacts



GRADLE also capable to run environments using built in JAVA dependencies.

-

