Gradle JVM build tool

CONCEPTS

Use ./gradlew (the wrapper) instead of a global gradle: it pins the version the project was built with.

Build settings.gradle(.kts)
Project build.gradle(.kts)
Task build, test, ...

Build structure

Tasks run with their dependencies, and up-to-date tasks are skipped. Use --rerun-tasks to force everything.

Prefix a task with : for the root project, or use sub:task to target a subproject.

RUN TASKS

./gradlew *<task>*
Run a task and its dependencies

./gradlew *<task1>* *<task2>*
Run several tasks in order

./gradlew build
Assemble and run all checks

./gradlew check
Run all verification tasks

./gradlew run
Assemble and run the application

./gradlew clean
Delete the build directory contents

./gradlew clean build
Rebuild from scratch

./gradlew *:<sub>:<task>*
Run a task in one subproject

./gradlew *<task>* **-x** *<name>*
Skip a task, even if others need it

./gradlew *<task>* **-m**
Dry run: show what would execute

./gradlew *<task>* **--continue**
Keep going after a failure

./gradlew *<task>* **-t**
Re-run when inputs change

./gradlew *<task>* **--rerun-tasks**
Ignore up-to-date checks

TESTING

./gradlew test
Run all tests

./gradlew test --tests *<Class>*
Run one test class

./gradlew test --tests *<Class>.<method>*
Run one test method

./gradlew test --tests *'<pkg>**'*
Run a package recursively

./gradlew test -i
Show test output in the console

OPTIONS

-i / --info
Log at info level

-s / --stacktrace
Print stack traces for user errors

-S / --full-stacktrace
Print the full stack trace

--scan
Create a Build Scan

--offline
Do not use the network

-U / --refresh-dependencies
Refresh dependency state

--parallel
Build projects in parallel

--build-cache
Enable the build cache

--console=plain
Plain output (no rich console)

-D*<prop>*=*<value>*
Set a JVM system property

-P*<prop>*=*<value>*
Set a project property

DAEMON

./gradlew --status
List running daemons

./gradlew --stop
Stop all daemons

--no-daemon
Run without the daemon

INSPECT

./gradlew tasks
List main tasks

INSPECT (cont.)

./gradlew tasks --all
Include tasks of all groups

./gradlew help --task *<name>*
Show usage of one task

./gradlew projects
List subprojects as a tree

./gradlew dependencies
Dependency tree per configuration

./gradlew dependencies --configuration *<name>*
Tree for one configuration

./gradlew dependencyInsight --dependency *<name>* **--configuration** *<cfg>*
Explain why a dependency is there

./gradlew -q *<task>*
Quiet output

SETUP AND WRAPPER

gradle init
Create a new build interactively

gradle init --type *<type>*
Pick a project type

gradle wrapper
Generate gradlew and wrapper files

gradle wrapper --gradle-version *<ver>*
Pin or upgrade the wrapper version

./gradlew --version
Show the Gradle version in use

RECIPES

Rebuild and debug a failing build

```
./gradlew clean build --
stacktrace --info
```

Why is this version chosen?

```
./gradlew dependencyInsight \
  --dependency guava \
  --configuration
runtimeClasspath
```

Fast loop on one test

```
./gradlew test --tests
UserServiceTest -t -i
```

Upgrade the wrapper

```
./gradlew wrapper --gradle-
version 9
./gradlew --version
```