KINGLAND®

Gradle Introduction

2021 April

Agenda

- Gradle Introduction
- Gradle Core Concept
- How to Define Tasks
- Build a Java Application
- Declare Dependences in Gradle Application
- Use Gradle Plugins

Gradle Introduction

What’s Gradle

- **Open-source build automation tool**
- **Written in Java, but the build scripts use Groovy, Kotlin DSL**
- **Support Declare Dependences**
- **Support custom by plugin, a wide plugin community**
- **What about Ant and Maven?**
  - Too many XML
  - Too many configuration

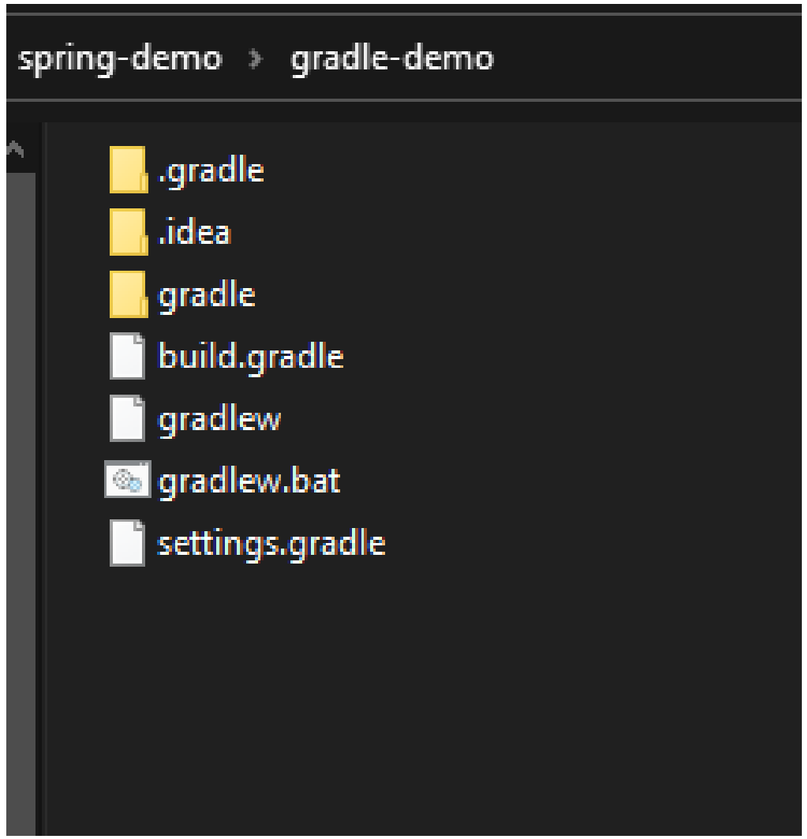
Why do we need Gradle?

- **Automate Build Process**
- **Manage Project Dependencies**
- **Implement CI/CD pipeline**

Gradle Core Concept

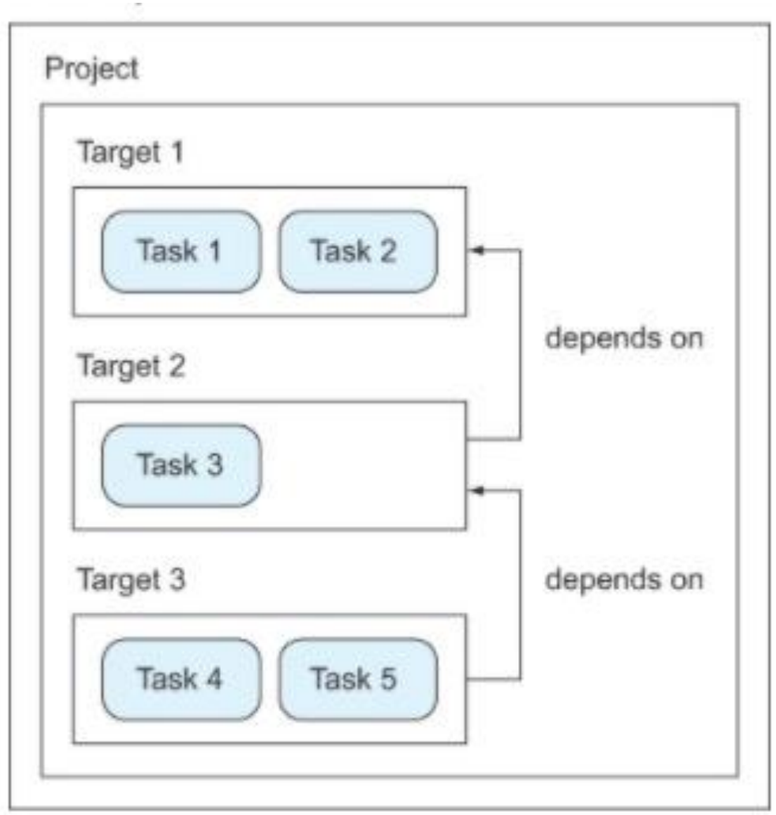
The Directory of a Gradle Project

- build.gradle: build script
- gradlew/gradlew.bat: runnable shell script
- settings.gradle: configuration file
- gradle: wrapper



Project

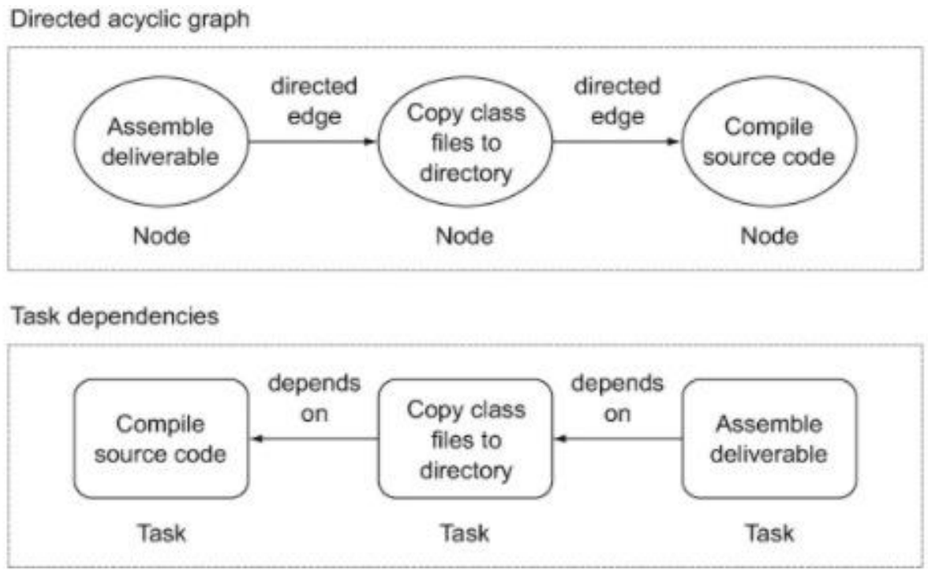
- **The root of a buildable unit**
- **Lifecycle**
  - Initialization
  - Configuration
  - Execution
- **Tasks**
- **Dependence**
- **Multi-project Build**



https://docs.gradle.org/current/dsl/org.gradle.api.Project.html

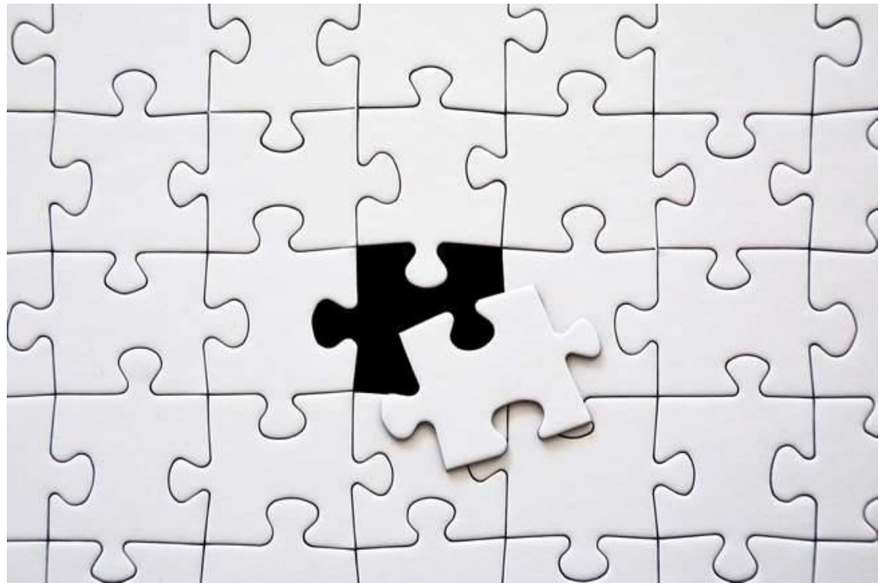
Task

- **The core runnable component**
- **All the build tasks will form a Directed Acyclic Graph(DAG)**



Plugin

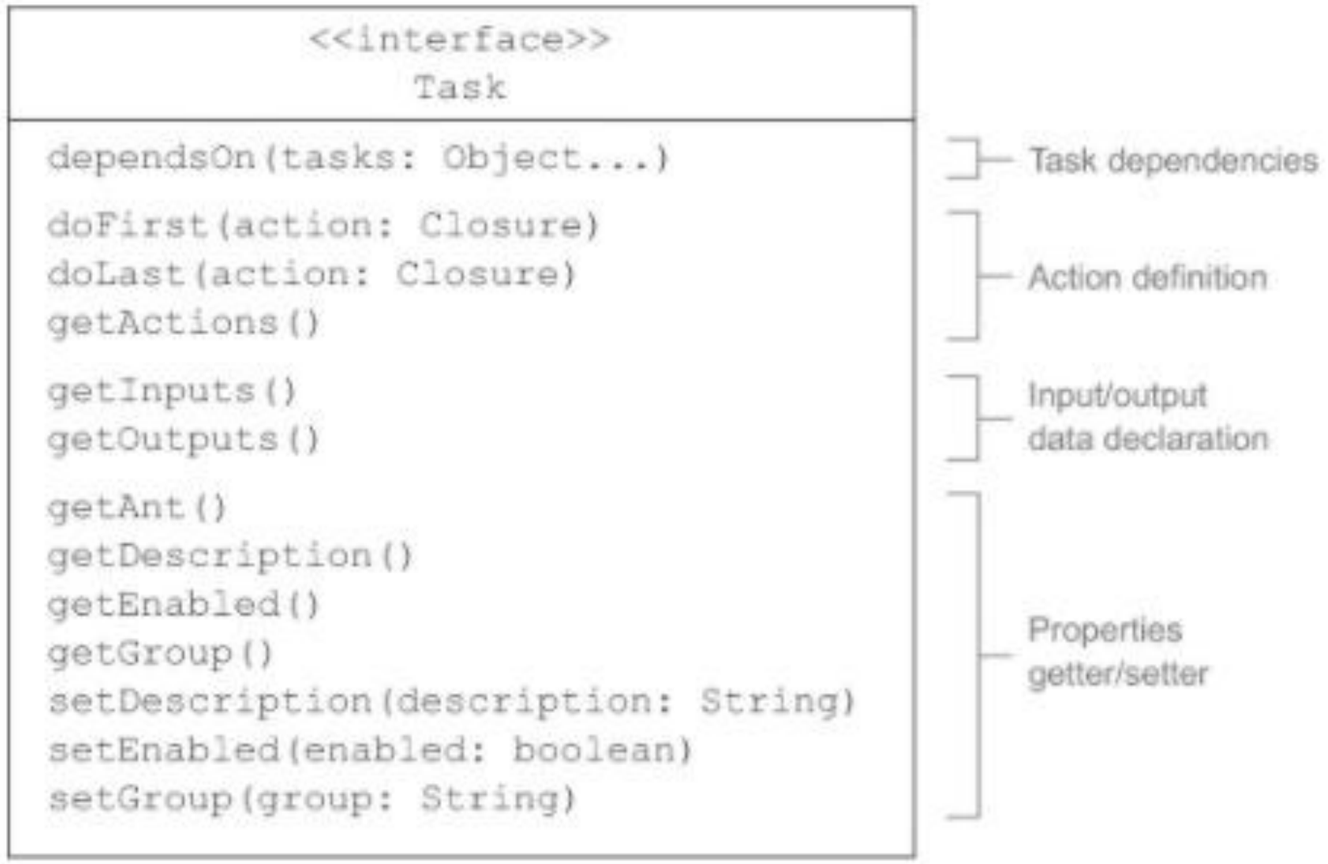
- **Gradle is just a lifecycle kernel.**
- **All the business feature are provided by Plugin**
- **Type:**
  - binary plugins
  - script plugins



How to Define Tasks

Task Interface

- **Task interface provided by Gradle**



Define a Task

- **Define task in build script**

```
task myTest{
    group="Test"
    doLast {
        println 'Hello world!'
    }
}

tasks.register("myTest3"){ Task it ->
    doLast{
        println project.name;
    }
}
```

Define Dependence

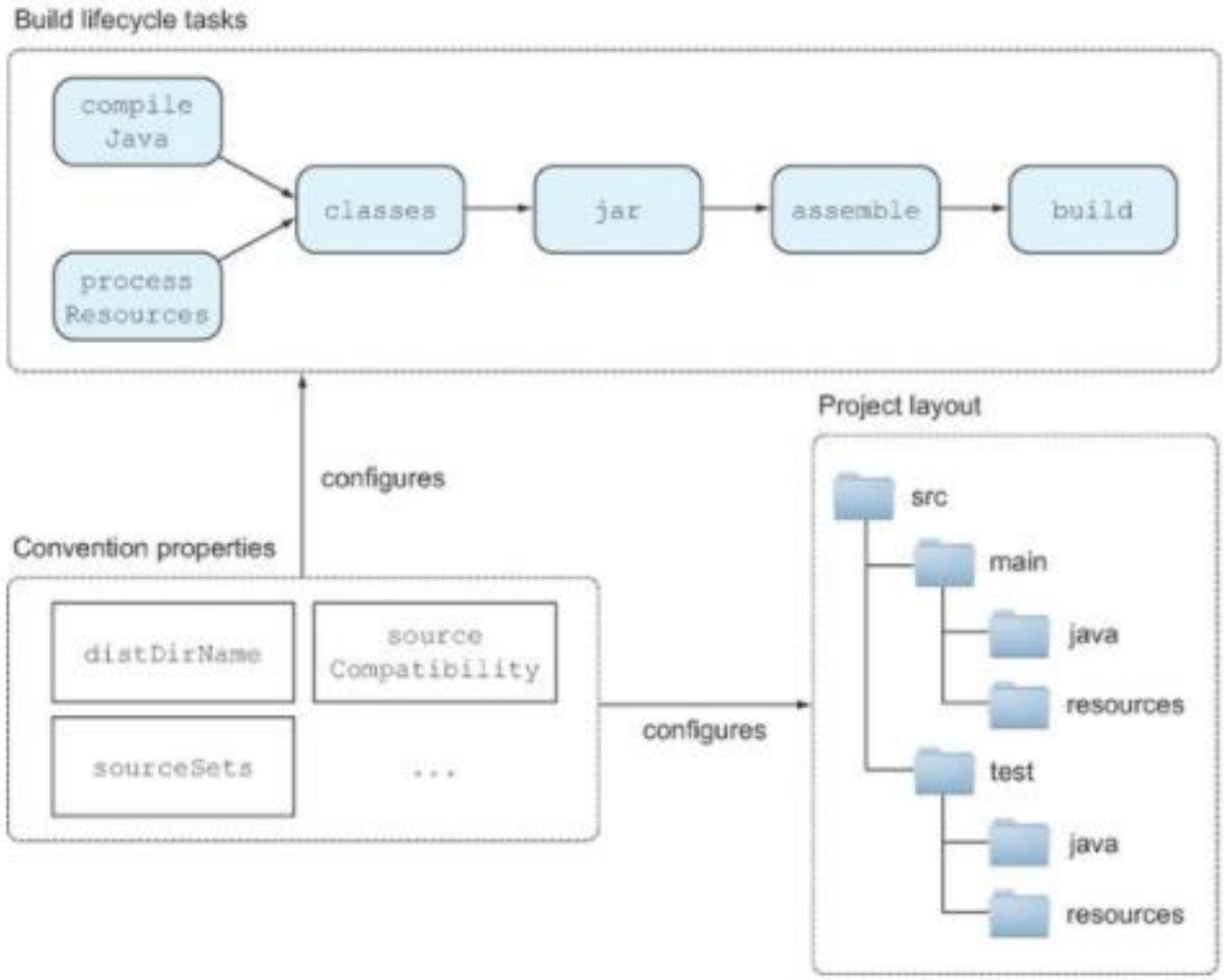
- **Use dependsOn**

```
task myTest2{
    group="Test"
    dependsOn myTest
    doLast {
        println("Hello World 2!")
    }
}
```

Build a Java Application

Project Structure

- **Generate a Gradle Java App**



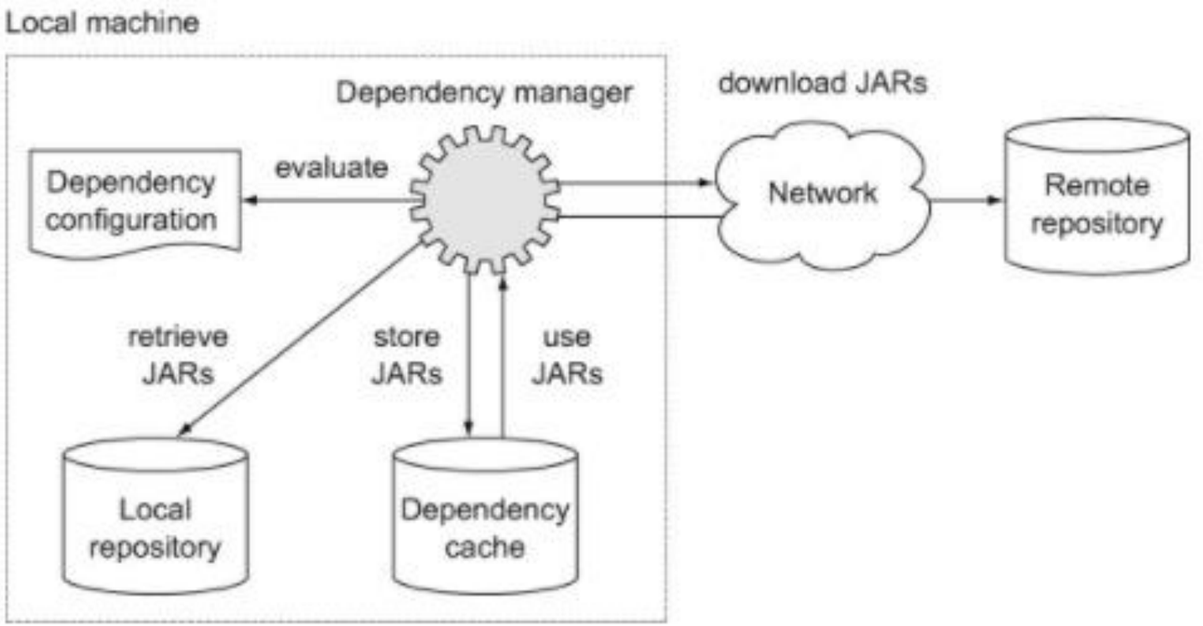
Package the Java Project

- gradle build task
- gradle clean task
- gradle jar task

Declare Dependences in Gradle Application

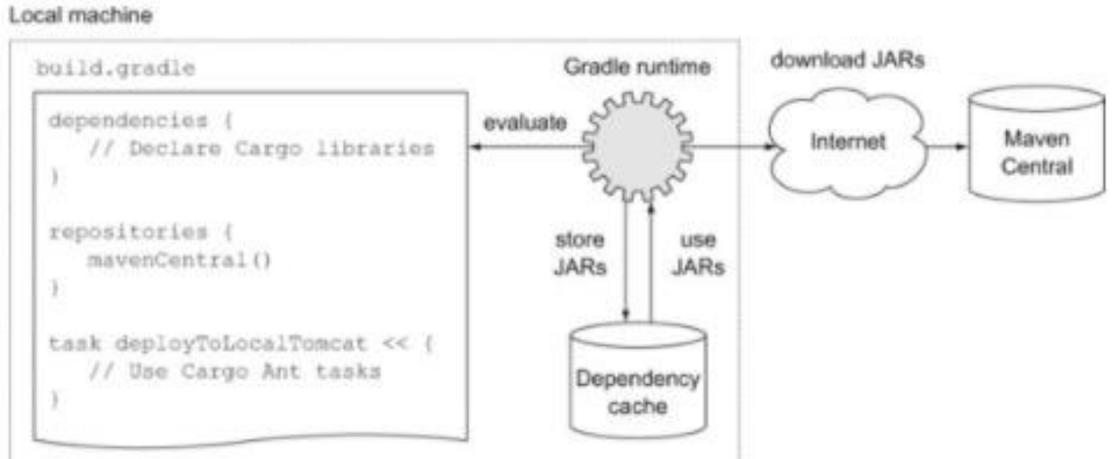
Define Repository

- **Repository: Where to get dependences**



Define Dependences

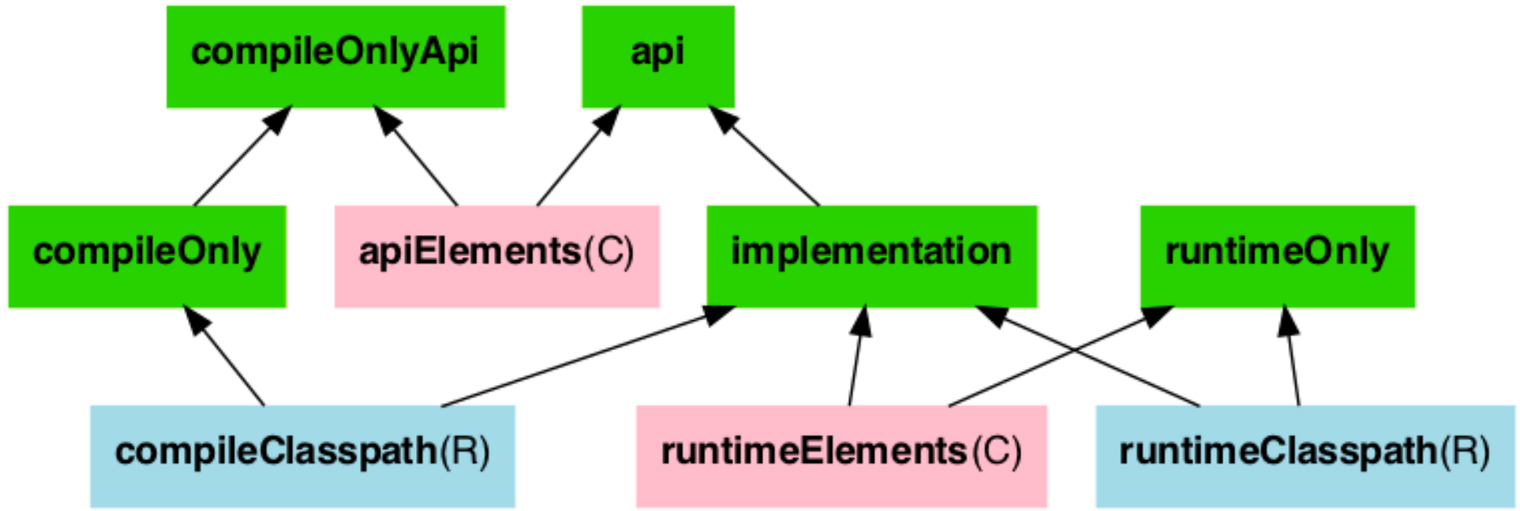
- **How to confirm a dependency**
- **Maven Coordinate**
  - Group
  - Artifact
  - Version




```groovy
dependencies {
    runtimeOnly group: 'org.springframework', name: 'spring-core', version: '2.5'
    runtimeOnly 'org.springframework:spring-core:2.5',
            'org.springframework:spring-aop:2.5'
    runtimeOnly(
        [group: 'org.springframework', name: 'spring-core', version: '2.5'],
        [group: 'org.springframework', name: 'spring-aop', version: '2.5']
    )
    runtimeOnly('org.hibernate:hibernate:3.0.5') {
        transitive = true
    }
    runtimeOnly group: 'org.hibernate', name: 'hibernate', version: '3.0.5', transitive: true
    runtimeOnly(group: 'org.hibernate', name: 'hibernate', version: '3.0.5') {
        transitive = true
    }
}
```

Different Kinds of Dependencies

- api
- implementation
- compileOnly
- runtimeOnly
- testImplementation



https://docs.gradle.org/current/userguide/java_library_plugin.html#sec:java_library_configurations_graph

Exclude dependencies

- **Why should we control the version some dependencies?**
- **Why should we exclude some dependencies?**
    - direct dependencies
    - transitive dependencies

```
dependencies {
    implementation('commons-beanutils:commons-beanutils:1.9.4') {
        exclude group: 'commons-collections', module: 'commons-collections'
    }
}
```

Use Gradle Plugins

What’s Plugin and What can we do with Plugin?

- The components that extend the feature of Gradle
- Extend the Gradle model
- Add new tasks
- Promotes reuse
- Modularization

```
plugins {
    id 'java'
    id 'jacoco'
    id 'pmd'
}
```

How to use PMD Plugin to check the code style

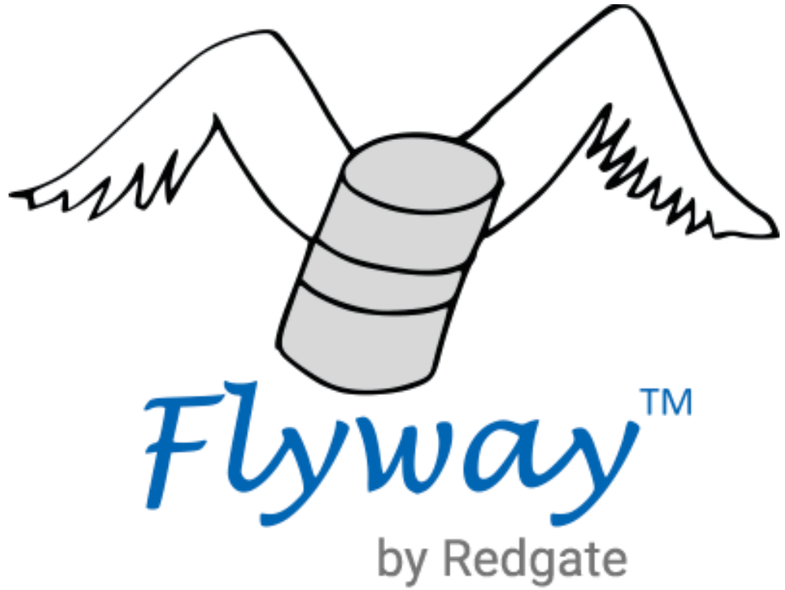
- PMD: https://pmd.github.io/
- Declare Plugin
- Configure Plugin
- Run
    - gradle check

How to use JUnit and JaCoCo Plugin

- **JUnit: the most popular unit test framework**
- **JaCoCo: a code coverage library for Java**
- **Integrate with Gradle**
  - gradle test

How to use Flyway Plugin

- **Maintain database version by Gradle and Flyway**
- **Declare plugin**
- **Configure plugin**
- **Run**
  - gradle flywayMigrate
  - gradle flywayClean



How to use Spring Boot Plugin

- **The Plugin provided by Spring Boot**
- **Run and package a Spring Boot Application**
- **Run**
  - gradle bootRun
  - gradle bootJar
  - gradle bootBuildImage

https://docs.spring.io/spring-boot/docs/2.4.5/gradle-plugin/reference/htmlsingle/

Kingland Systems. Discover Progress.

Our clients **know** that Kingland Systems delivers **faster, smarter,** more **reliable** solutions**.**

INDUSTRY SOLUTIONS

Kingland has been delivering Industry-specific solutions to leading global enterprises for more than 23 years.

SOLUTION PLATFORM

The Kingland Strategic Solution Platform means continuously smarter technology to deliver today and into the future.

EXPERT SERVICES

Kingland brings deep data and software expertise to every solution, helping you realize benefits swiftly — and with less risk.

KINGLAND

KINGLAND®

Thank You!