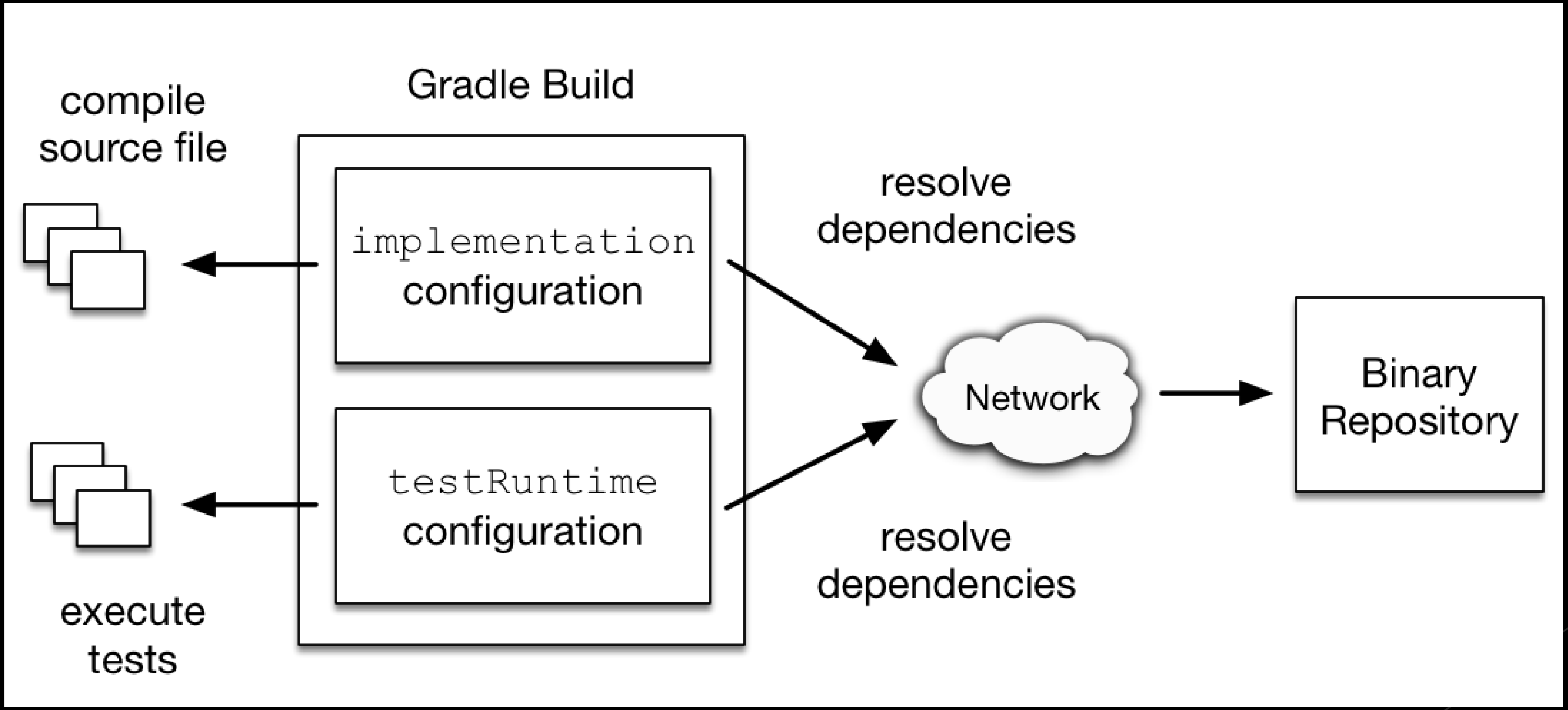
Gradle

A brief introduction to Gradle using the KotlinDSL

By Jonas Bürgel

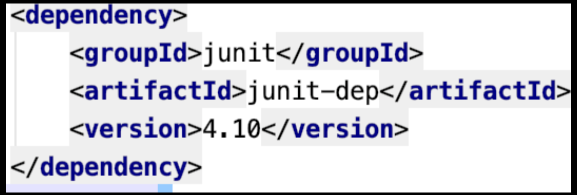
What does Gradle do?



Why KotlinDSL?

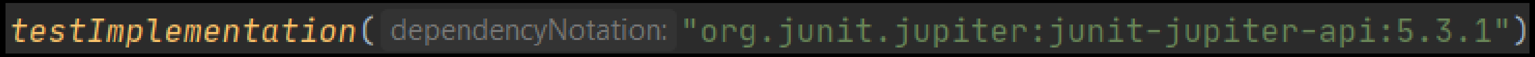
- Type safety
  - Code Completion
  - Compile time errors
- Only one language?

Maven:



```
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit-dep</artifactId>
    <version>4.10</version>
</dependency>
```

KotlinDSL:



```
testImplementation( dependencyNotation: "org.junit.jupiter:junit-jupiter-api:5.3.1")
```

Gradle vs Maven

Gradle

- Performance
  - Between 2x and 100x faster
- User experience
  - KotlinDSL
- Flexibility
  - Support for C/C++
- Android Development

Maven

- Traditional
  - Already familiar
  - More help on the internet?
- Simpler?

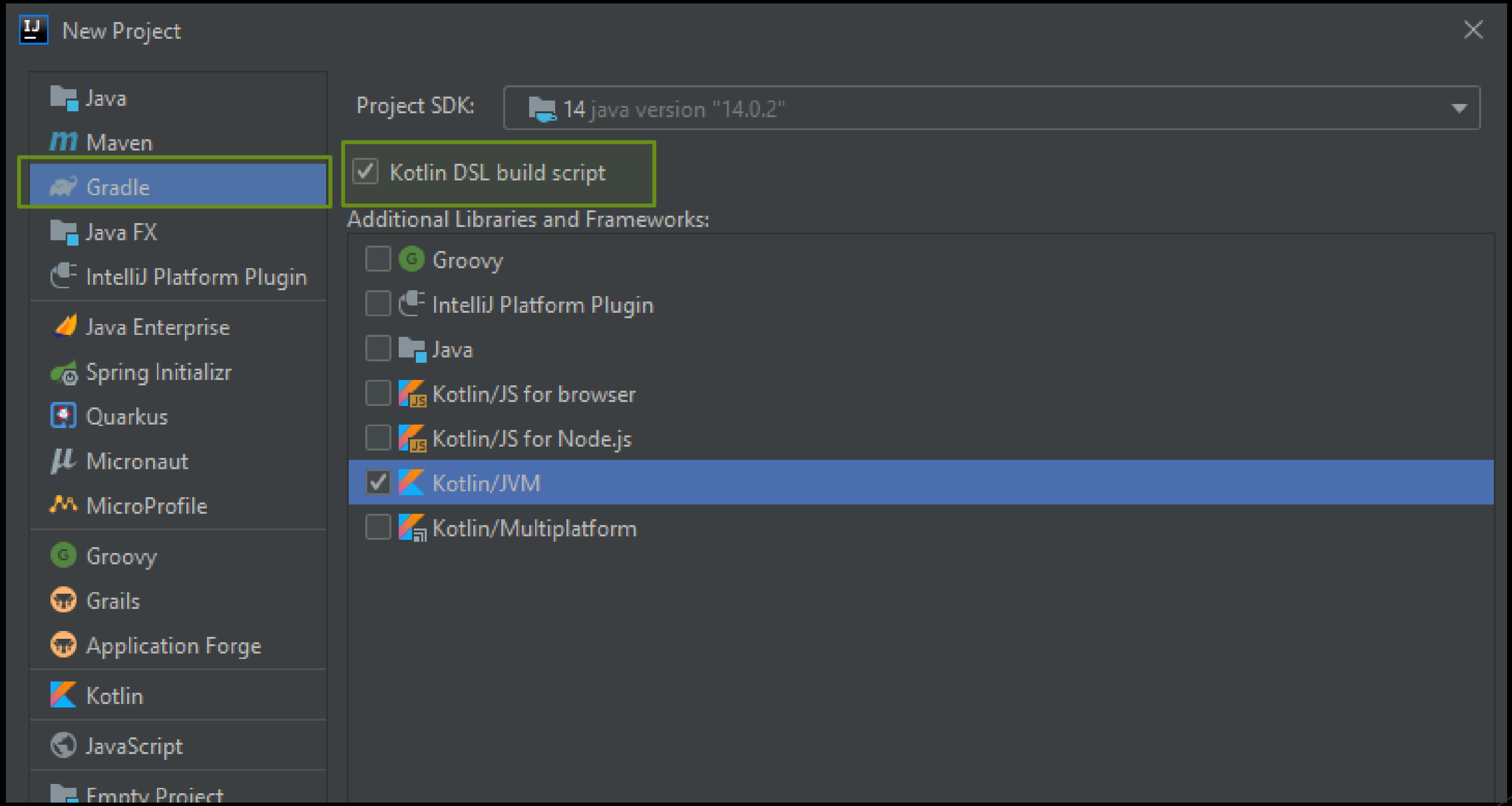
Presentation Structure

1. Basics
   1. Project structure
   2. Manage dependencies

2. Modules
3. Tasks
   1. Custom tasks
   2. Custom task types

Basics

Creating a Gradle Project



New Project

- Java
- Maven
- Gradle
- Java FX
- IntelliJ Platform Plugin
- Java Enterprise
- Spring Initializr
- Quarkus
- Micronaut
- MicroProfile
- Groovy
- Grails
- Application Forge
- Kotlin
- JavaScript
- Empty Project

Project SDK: 14 java version "14.0.2"

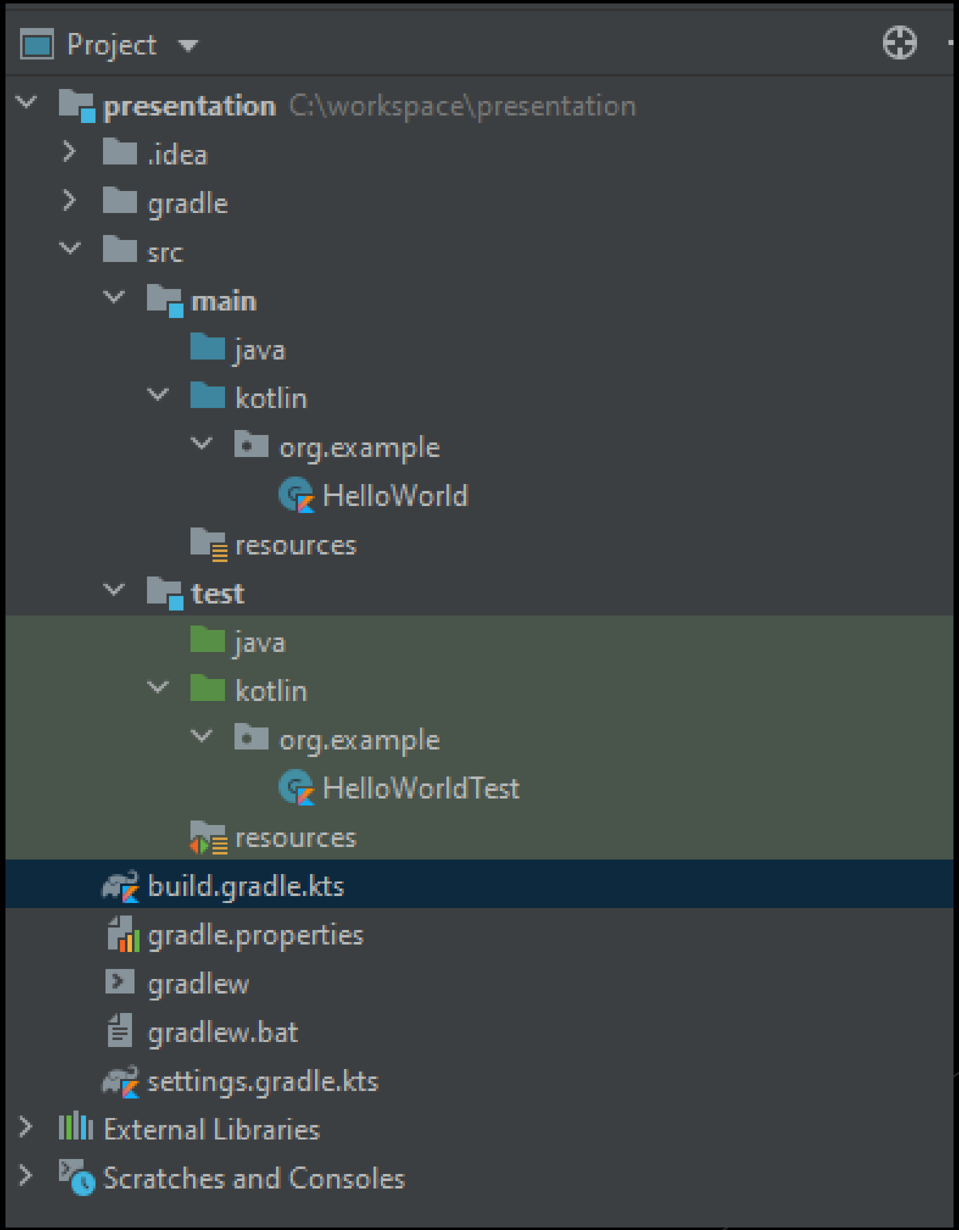
[✓] Kotlin DSL build script

Additional Libraries and Frameworks:

- [ ] Groovy
- [ ] IntelliJ Platform Plugin
- [ ] Java
- [ ] Kotlin/JS for browser
- [ ] Kotlin/JS for Node.js
- [✓] Kotlin/JVM
- [ ] Kotlin/Multiplatform

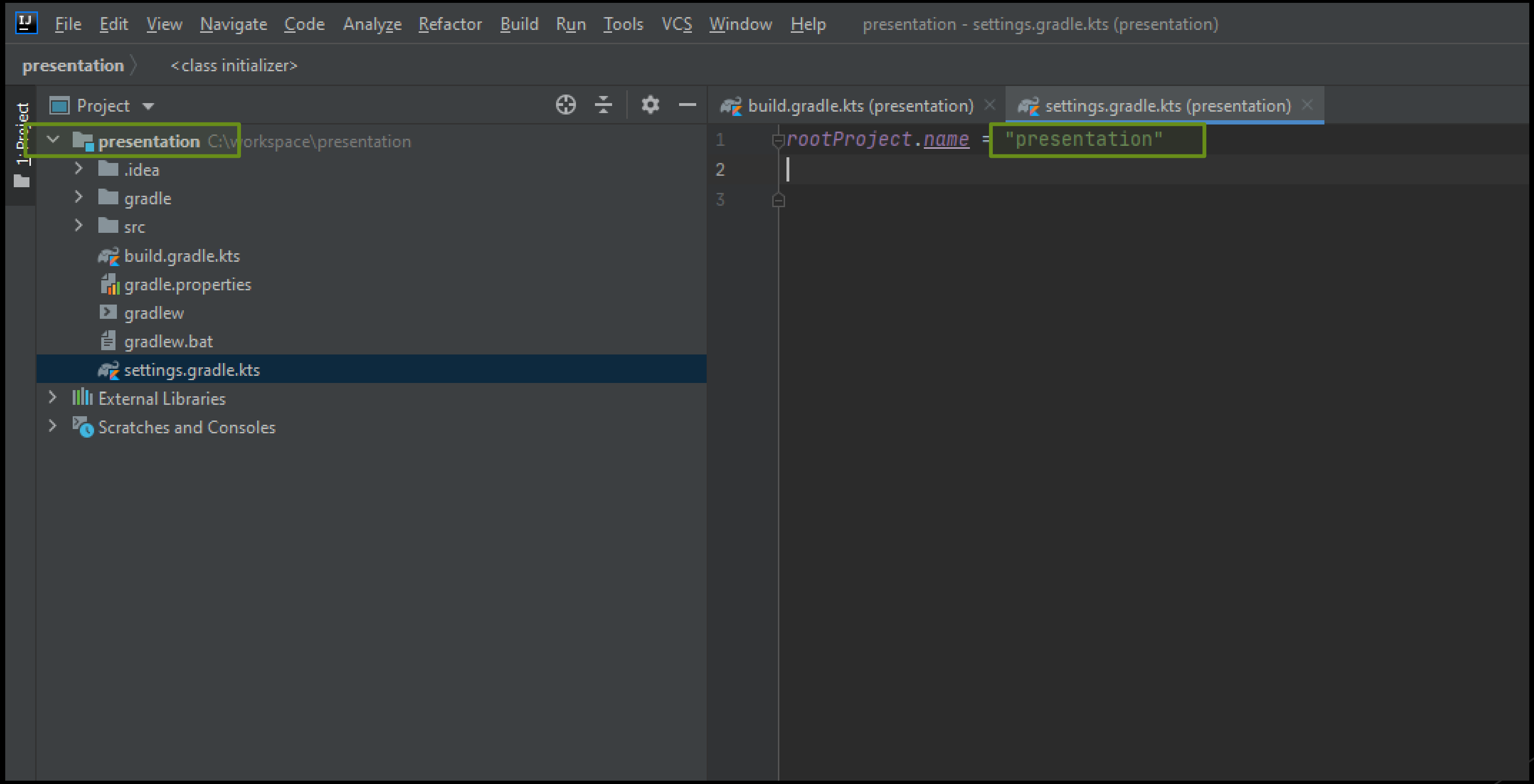
Project structure: src

- Sort code by language
- Resources: static files
- Mirror “main” and “test” paths
- Use “reverse-url” packaging
  - Avoid name conflicts with imports



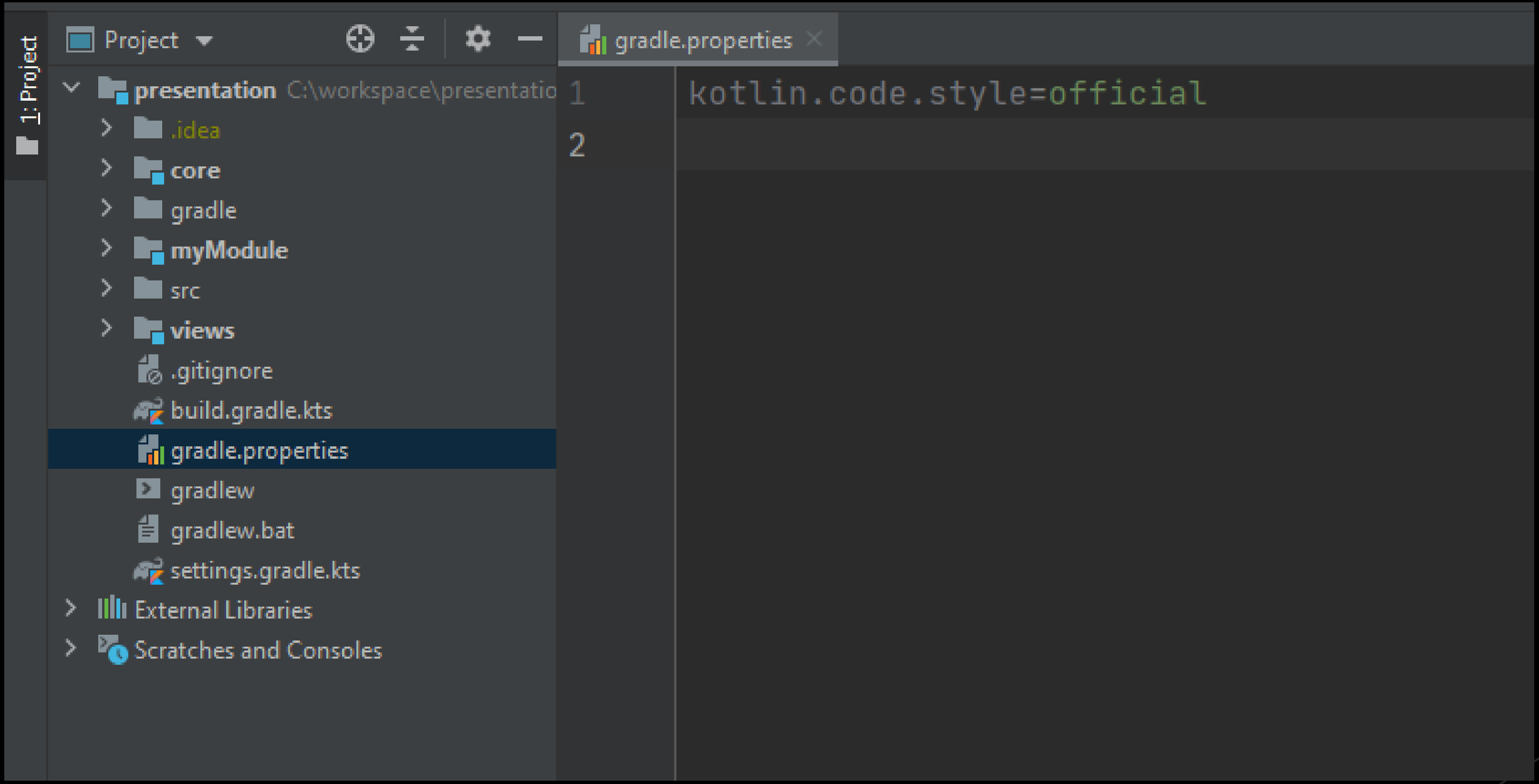
```
Project
presentation  C:\workspace\presentation
  .idea
  gradle
  src
    main
      java
      kotlin
        org.example
          HelloWorld
      resources
    test
      java
      kotlin
        org.example
          HelloWorldTest
      resources
  build.gradle.kts
  gradle.properties
  gradlew
  gradlew.bat
  settings.gradle.kts
External Libraries
Scratches and Consoles
```

Project structure: settings.gradle.kts


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
presentation - settings.gradle.kts (presentation)
presentation
<class initializer>
Project
build.gradle.kts (presentation)
settings.gradle.kts (presentation)
presentation C:\workspace\presentation
.idea
gradle
src
build.gradle.kts
gradle.properties
gradlew
gradlew.bat
settings.gradle.kts
External Libraries
Scratches and Consoles

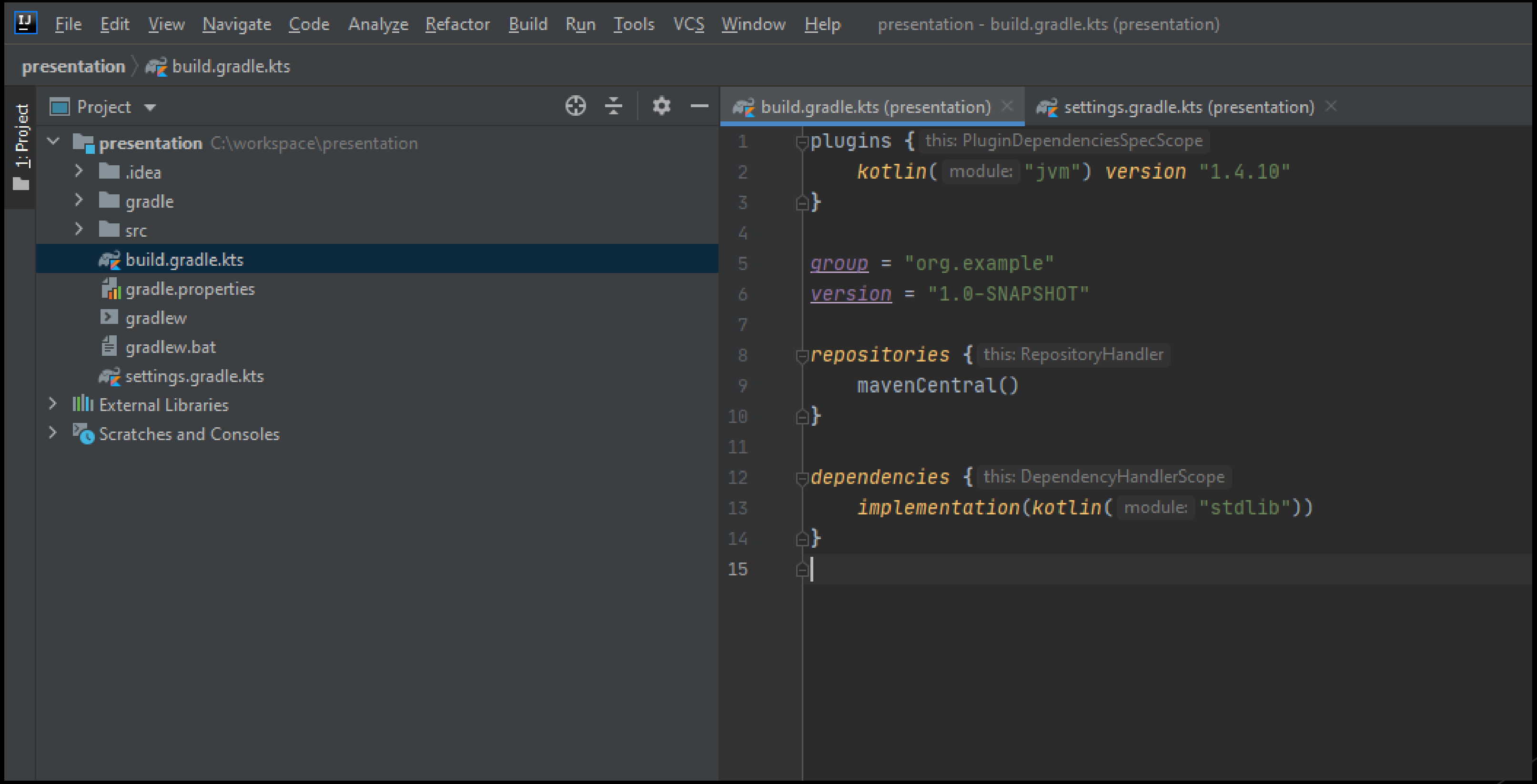
```kotlin
rootProject.name = "presentation"
```

Project structure: gradle.properties



```
kotlin.code.style=official

```

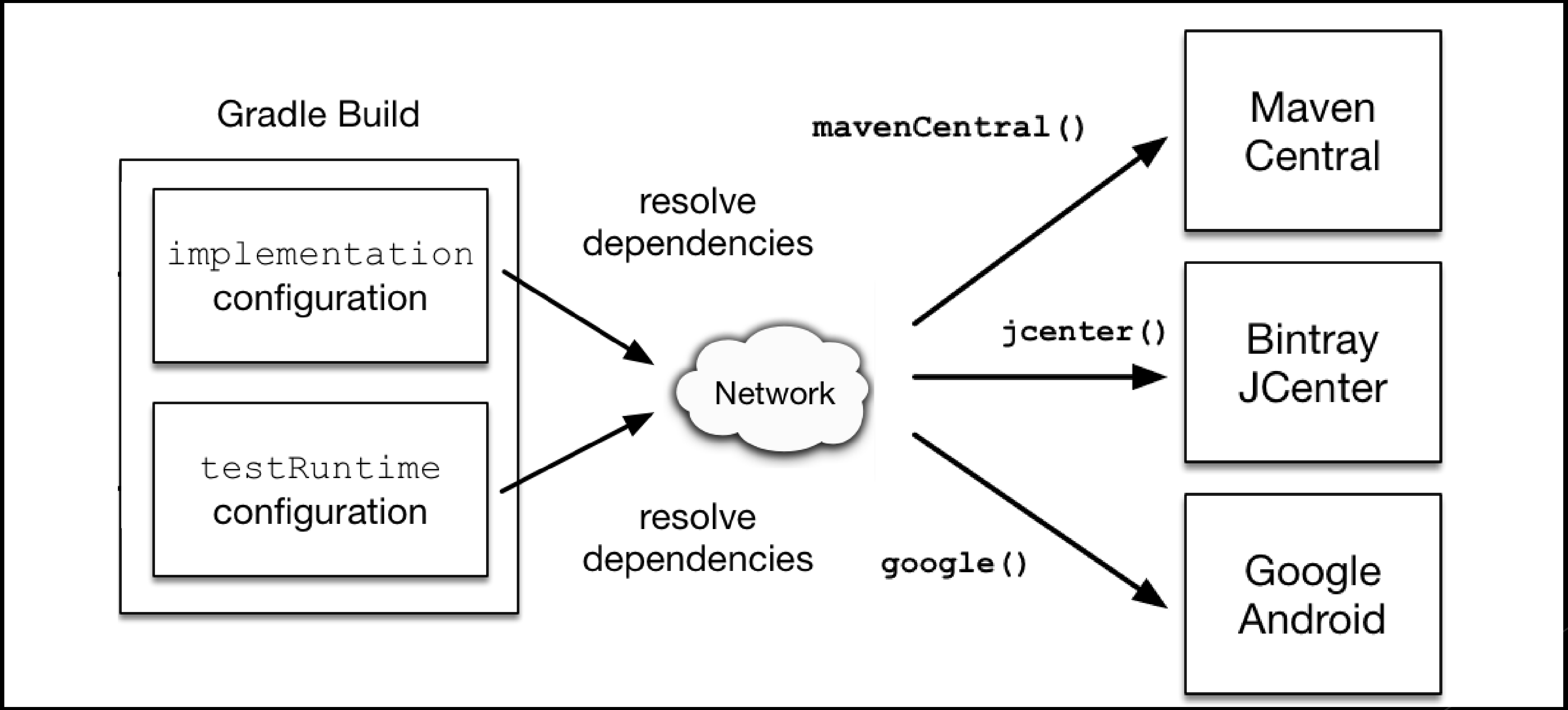
Project structure: build.gradle.kts


File
Edit
View
Navigate
Code
Analyze
Refactor
Build
Run
Tools
VCS
Window
Help
presentation - build.gradle.kts (presentation)
presentation
build.gradle.kts
1: Project
Project
build.gradle.kts (presentation)
settings.gradle.kts (presentation)

```
presentation C:\workspace\presentation
    .idea
    gradle
    src
    build.gradle.kts
    gradle.properties
    gradlew
    gradlew.bat
    settings.gradle.kts
External Libraries
Scratches and Consoles
```

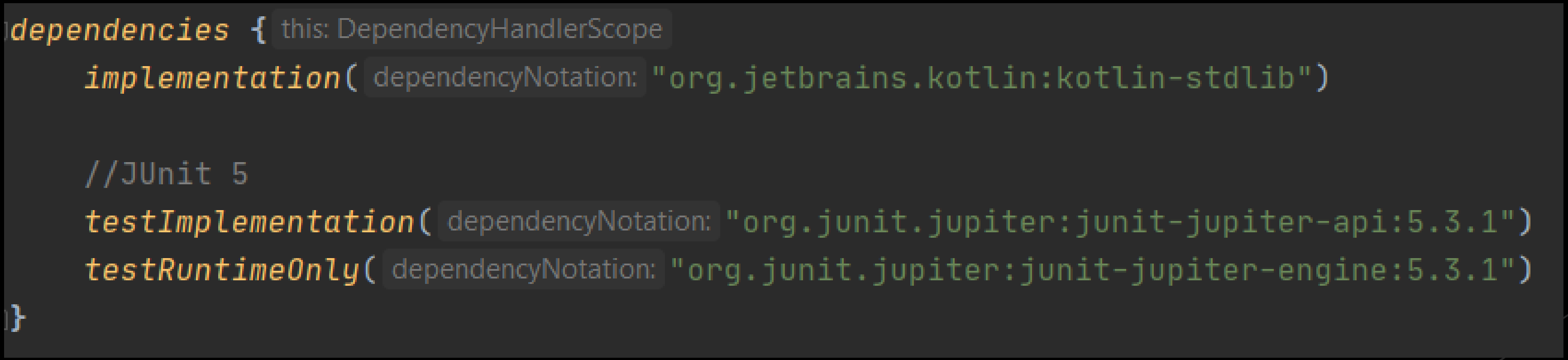
```kotlin
plugins { this: PluginDependenciesSpecScope
    kotlin( module: "jvm") version "1.4.10"
}

group = "org.example"
version = "1.0-SNAPSHOT"

repositories { this: RepositoryHandler
    mavenCentral()
}

dependencies { this: DependencyHandlerScope
    implementation(kotlin( module: "stdlib"))
}
```

Repositories



Dependencies

- Structure: <configuration>("<group>:<artifact>:<version>")
- Configurations
  - implementation: import dependency
  - testImplementation: import dependency ONLY for tests



```
dependencies { this: DependencyHandlerScope
    implementation( dependencyNotation: "org.jetbrains.kotlin:kotlin-stdlib")

    //JUnit 5
    testImplementation( dependencyNotation: "org.junit.jupiter:junit-jupiter-api:5.3.1")
    testRuntimeOnly( dependencyNotation: "org.junit.jupiter:junit-jupiter-engine:5.3.1")
}
```

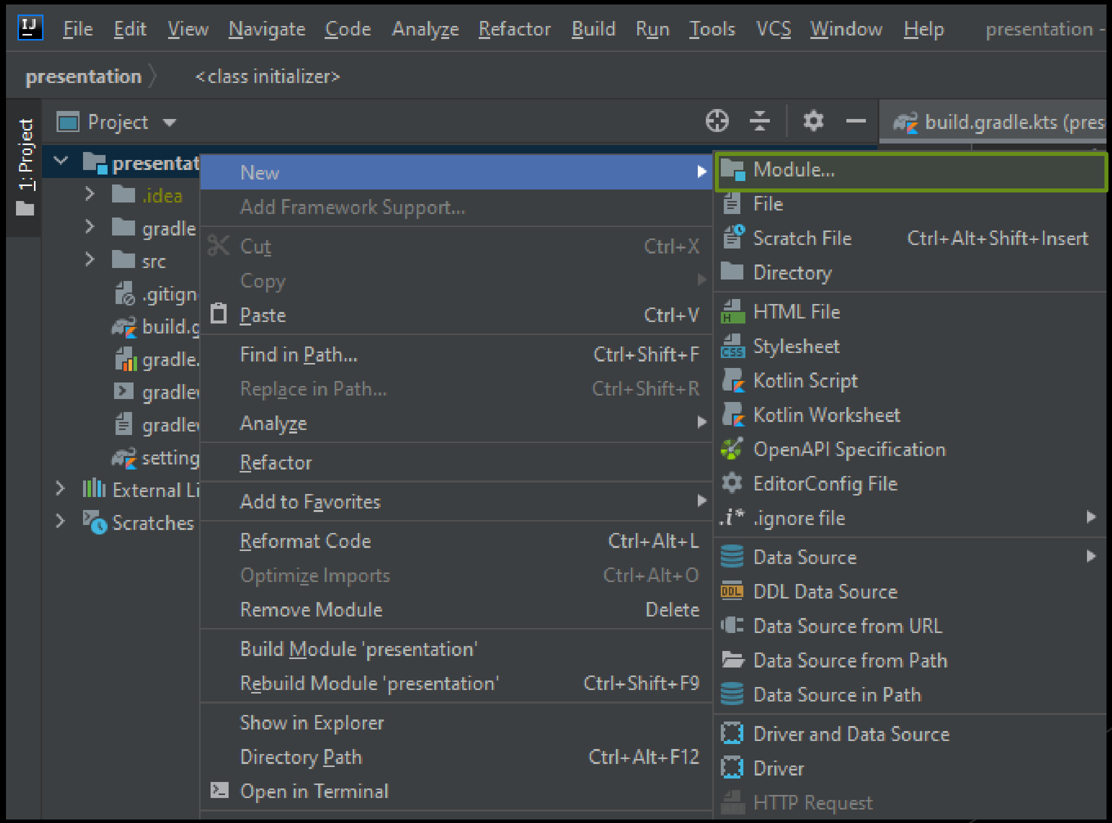
Modules

What are modules?

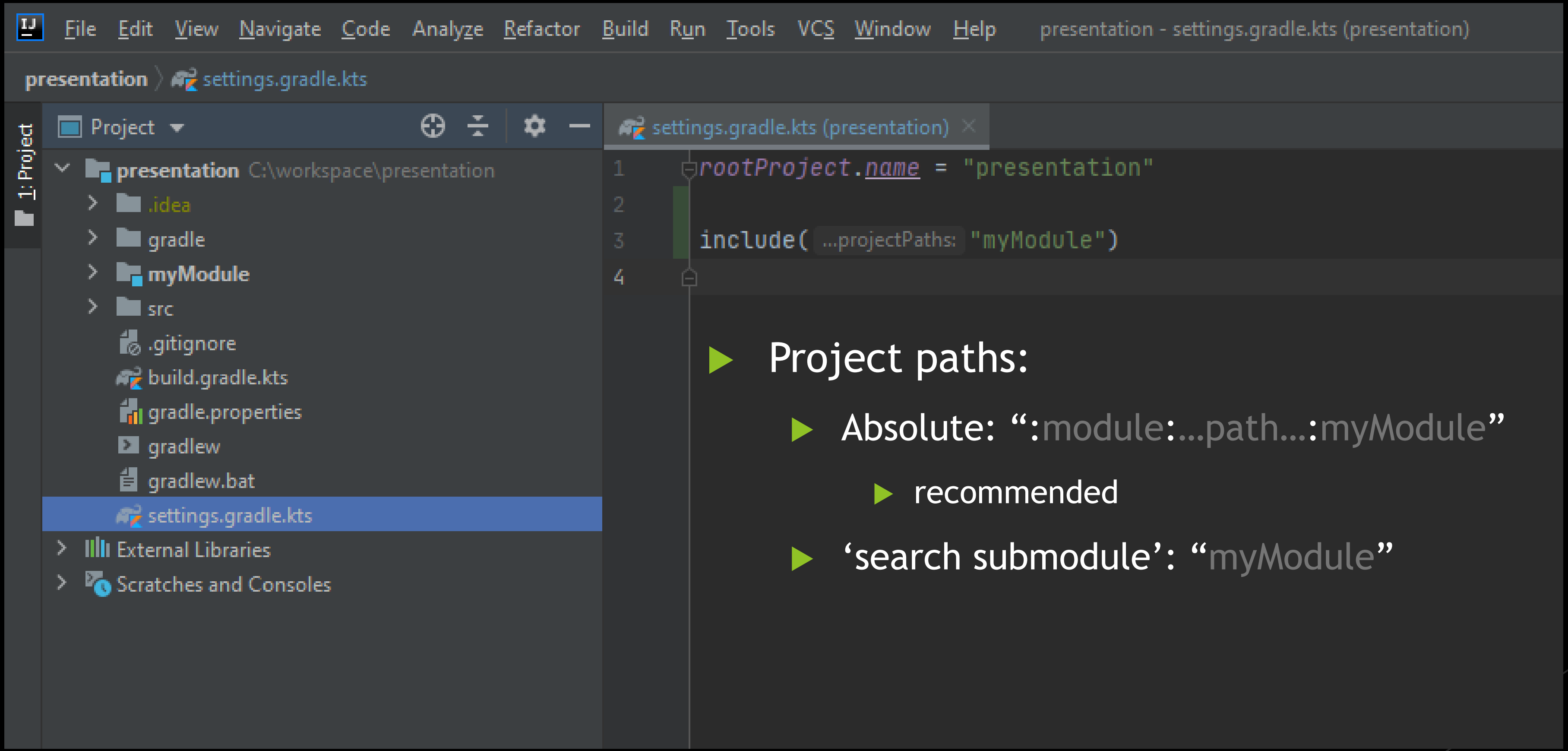
- “Encapsulated packages”
  - Own dependencies/build-script/…
  - Consist of packages
- Properties of good modules
  - Low coupling
  - Strong encapsulation
  - Do one thing and do it well

Create a module in Gradle

- name
  - unique
  - kebab-name-convention
- IntelliJ:
  - Remove generated versions from plugins



Include a module in Gradle


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
presentation - settings.gradle.kts (presentation)
presentation 〉 settings.gradle.kts
1: Project
Project
presentation C:\workspace\presentation
.idea
gradle
myModule
src
.gitignore
build.gradle.kts
gradle.properties
gradlew
gradlew.bat
settings.gradle.kts
External Libraries
Scratches and Consoles
settings.gradle.kts (presentation)

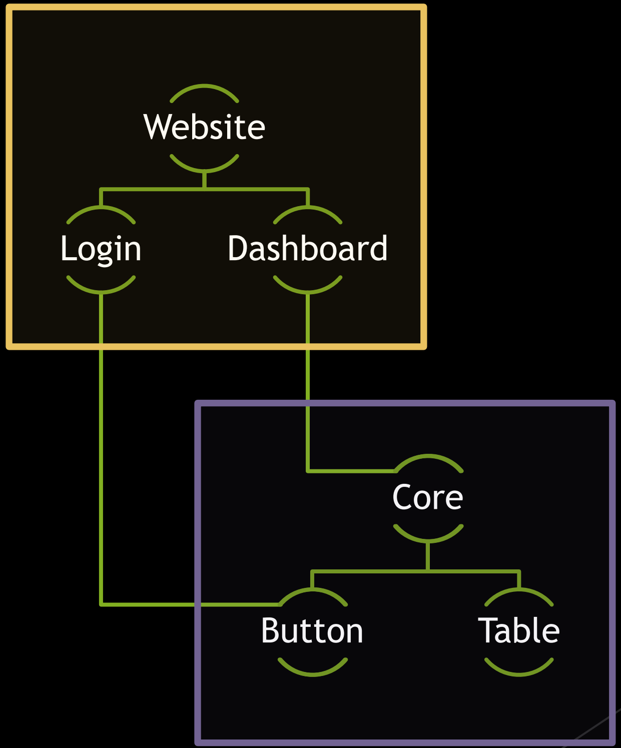
```
rootProject.name = "presentation"

include( ...projectPaths: "myModule")
```

- Project paths:
  - Absolute: ":module:…path…:myModule"
    - recommended
  - 'search submodule': "myModule"

Module Dependencies

- Login: implementation(project(“:core:button”))
  - Website doesn’t know about “button”
  - Usually best
- Core: api(project(“:core:button”))
  - Dashboard knows about “button”
  - Implement + expose

- Website: subprojects {}
  - Affects: “:Website:Dashboard”, “:Website:Login”
- Any module: allprojects {}
  - Affects all modules -> use carefully!



Tasks

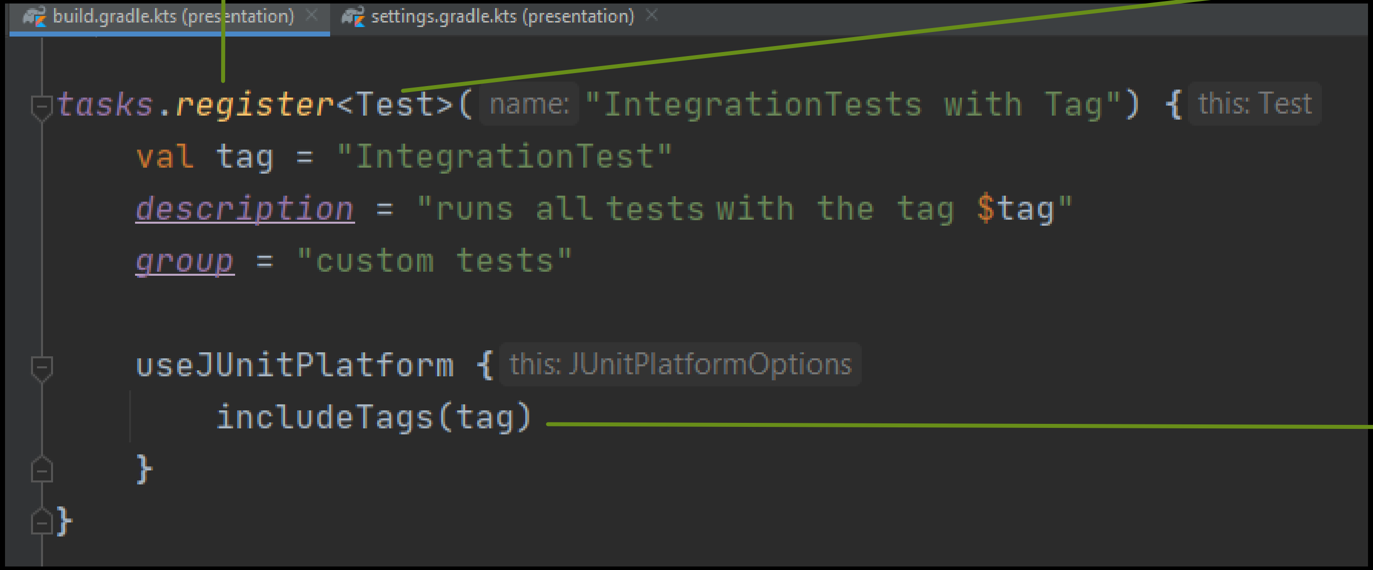
What are Gradle tasks?

- *”atomic piece of work for a build”*
- Examples
  - update versioning
  - run tests
  - deploy project to server
  - build project
  - …

Custom Task: run tests with tag

register task in Gradle

Predefined Task Type -> runs all tests

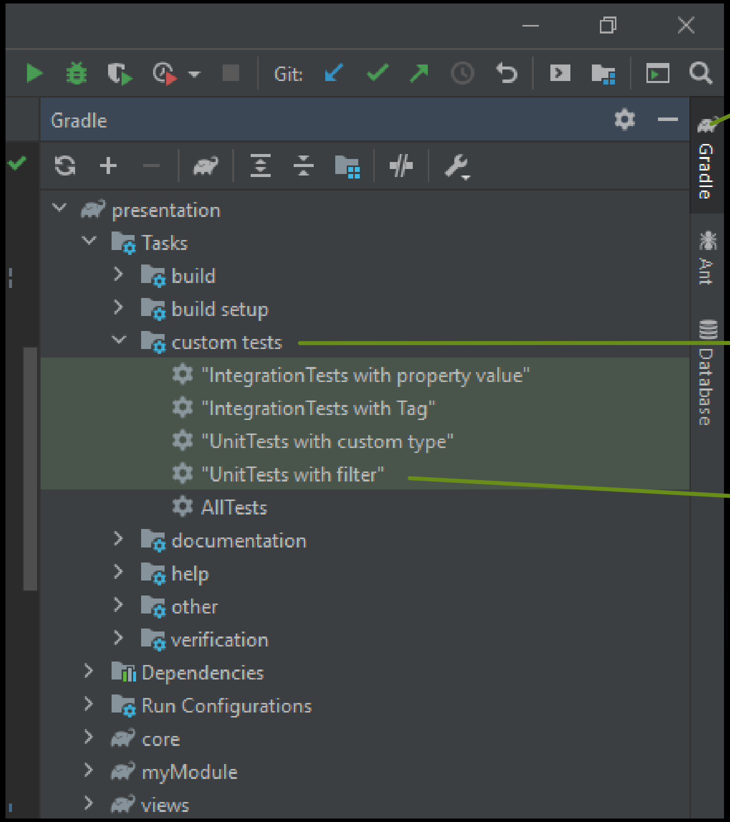

build.gradle.kts (presentation)
settings.gradle.kts (presentation)

```kotlin
tasks.register<Test>( name: "IntegrationTests with Tag") { this: Test
    val tag = "IntegrationTest"
    description = "runs all tests with the tag $tag"
    group = "custom tests"

    useJUnitPlatform { this: JUnitPlatformOptions
        includeTags(tag)
    }
}
```

Abuse JUnit tags -> only run tasks with specific tag

@Tag("IntegrationTest")

Task Overview in IntelliJ



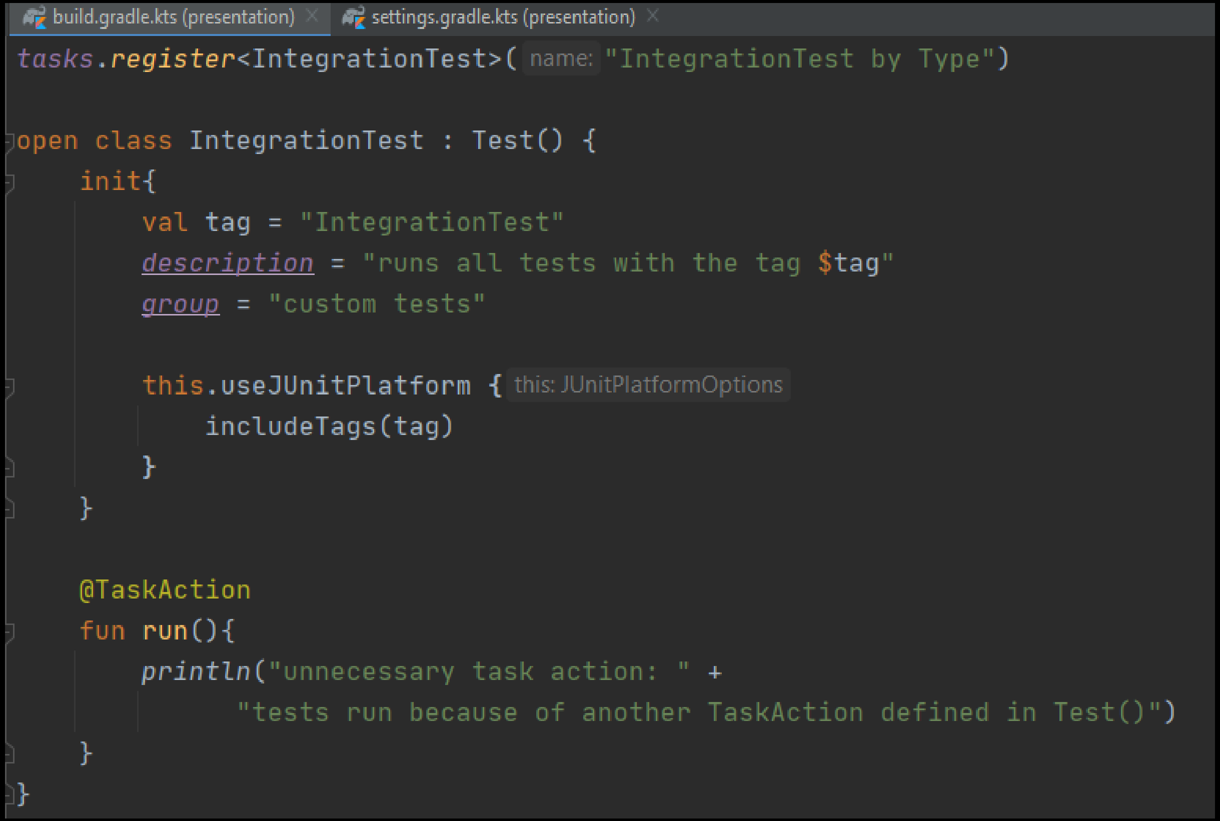
```
Gradle
presentation
  Tasks
    build
    build setup
    custom tests
      "IntegrationTests with property value"
      "IntegrationTests with Tag"
      "UnitTests with custom type"
      "UnitTests with filter"
      AllTests
    documentation
    help
    other
    verification
  Dependencies
  Run Configurations
  core
  myModule
  views
```

Click here for the task view

Task group

Task name

Custom Task Type


build.gradle.kts (presentation)
settings.gradle.kts (presentation)

```
tasks.register<IntegrationTest>( name: "IntegrationTest by Type")

open class IntegrationTest : Test() {
    init{
        val tag = "IntegrationTest"
        description = "runs all tests with the tag $tag"
        group = "custom tests"

        this.useJUnitPlatform { this: JUnitPlatformOptions
            includeTags(tag)
        }
    }

    @TaskAction
    fun run(){
        println("unnecessary task action: " +
                "tests run because of another TaskAction defined in Test()")
    }
}
```

Useful tips/tricks

- tasks.create -> same as tasks.register but eager and not lazy
- tasks.withType<Test> { group = “tests” }

- Access gradle.properties: project.properties[“org.example.setting”]

- open class MyTask : DefaultTask() {
  - @Input
    var s: String = “”
  - doFirst() / doLast()
  - dependsOn() / mustRunAfter() / onlyIf()

  }

Useful links

- https://docs.gradle.org/
- https://stackoverflow.com/questions/7249871/what-is-a-build-tool
- https://medium.com/@riag_digital/codify-your-configs-with-kotlin-dsl-and-gradle-5-35081e513088
- https://stackoverflow.com/questions/30170539/how-to-use-gradle-properties-in-build-gradle
- https://stackoverflow.com/questions/53654190/what-is-the-difference-between-registering-and-creating-in-gradle-kotlin-dsl
- https://stackoverflow.com/questions/38910129/what-is-the-difference-between-an-app-dependency-and-a-module-dependency-plugin

pictures

- Gradle Banner: https://miro.medium.com/max/700/1*EOdsCmvfHf37LvjpBAz5rg.png
- Gradle workflow: https://docs.gradle.org/current/userguide/img/dependency-management-configurations.png
- Gradle repositories: https://docs.gradle.org/current/userguide/img/dependency-management-shortcut-repositories.png
- Maven dependency: https://resources.jetbrains.com/help/img/idea/2020.2/maven_pom_dependency.png

Thank you for your attention