Plugin

What is a plugin?

A plugin is a piece of software that provides additional functionality to the gradle build system

Plugins add new tasks

Plugins add new configurations

Plugins add other build related capabilities

For example, the “java-library” plugin

This is used to define and build java libraries!

For example, the “com.google.gms:google-services plugin

For example, the “com.jfrog.bintray” plugin

Allows you to publish artifacts to bintray!

Plugin distribution

1. Core plugins – developed by gradle
2. Community plugins – gradle’s community shares plugins
3. Local plugins – users can create custom plugins!

How do you apply a plugin?

plugins {

id «plugin id» version «plugin version»

}

For example,

plugins {

id 'application'

}

Now we talk about core plugins

* Included in the gradle distribution itself
* Some examples include java, groovy, ear
* plugins {
* id("java")
* }

Now we talk about community plugins

* These are developed by the gradle community
* As an example, below
* plugins {
* id("org.springframework.boot") version "3.1.5"
* }

Now, we talk about local plugins

class HelloPlugin : Plugin<Project> {

override fun apply(project: Project) {

// Define the 'hello' task

val helloTask = project.tasks.register("hello") {

doLast {

println("Hello, Gradle!")

}

}

}

}

In the above, we wrote a plugin class

Then, build the plugin

plugins {

`maven-publish`

}

publishing {

publications {

create<MavenPublication>("mavenJava") {

from(components["java"])

}

}

repositories {

mavenLocal()

}

}

Finally, apply the plugin

// Apply the plugin

plugins {

id("com.example.hello") version "1.0"

}