Gradle

可能遇见的问题：

1. Gradel版本与 SpringBooot等不兼容
2. Gradel版本与idea版本不兼容

构建工具	优点	缺点
Ant	使用灵活, 速度快于Maven, gradle	没有强加任何编码约定的项目目录结构, 开发人员需要编写复杂XML文件构建指令, 对开发人员是一个挑战.
Maven	遵循一套约定大于配置的项目目录结构, 使用同意的GAV坐标进行依赖管理, 侧重于包管理.	项目构建过程僵化, 配置文件编写不够灵活, 不方便自定义组件, 构建速度慢于Gradle
Gradle	集Ant脚本的灵活性+Maven约定大于配置的项目目录优势, 支持多种远程仓库和插件, 侧重于大项目构建.	学习成本高, 资料少, 脚本灵活, 版本兼容性差.

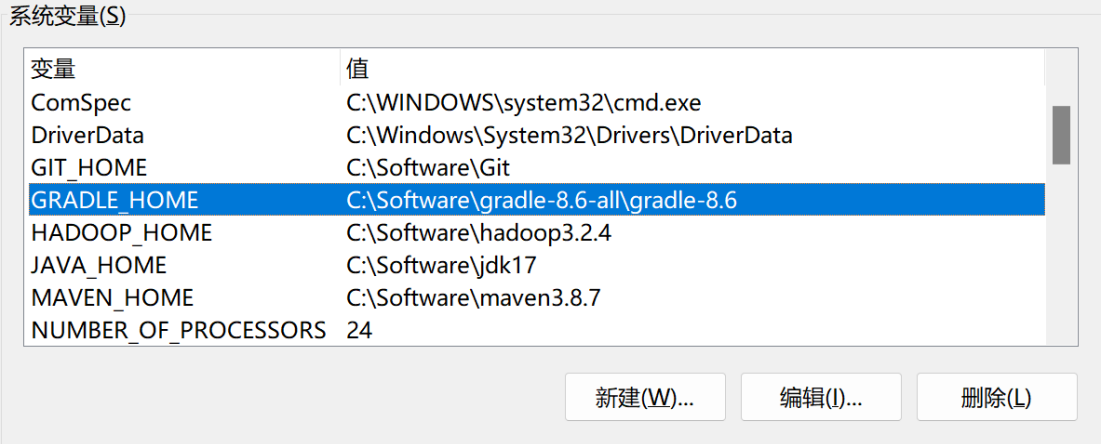
自动化构建工具对比	Ant	Maven	Gradle
构建性能	最高	最低	居中
仓库	开发者自己处理	maven仓库	支持多种远程仓库
依赖管理	ivy管理	GAV坐标管理	GNV坐标管理
插件支持	实现方便	实现较难	实现方便
遵循特定目录结构	No	遵循	同maven
配置文件	xml文件最为繁琐	xml文件	代码脚本,便于写业务逻辑
侧重点	小型项目构建	项目包管理	大型项目构建
目前地位	使用较少	目前主流	未来趋势(spring家族)

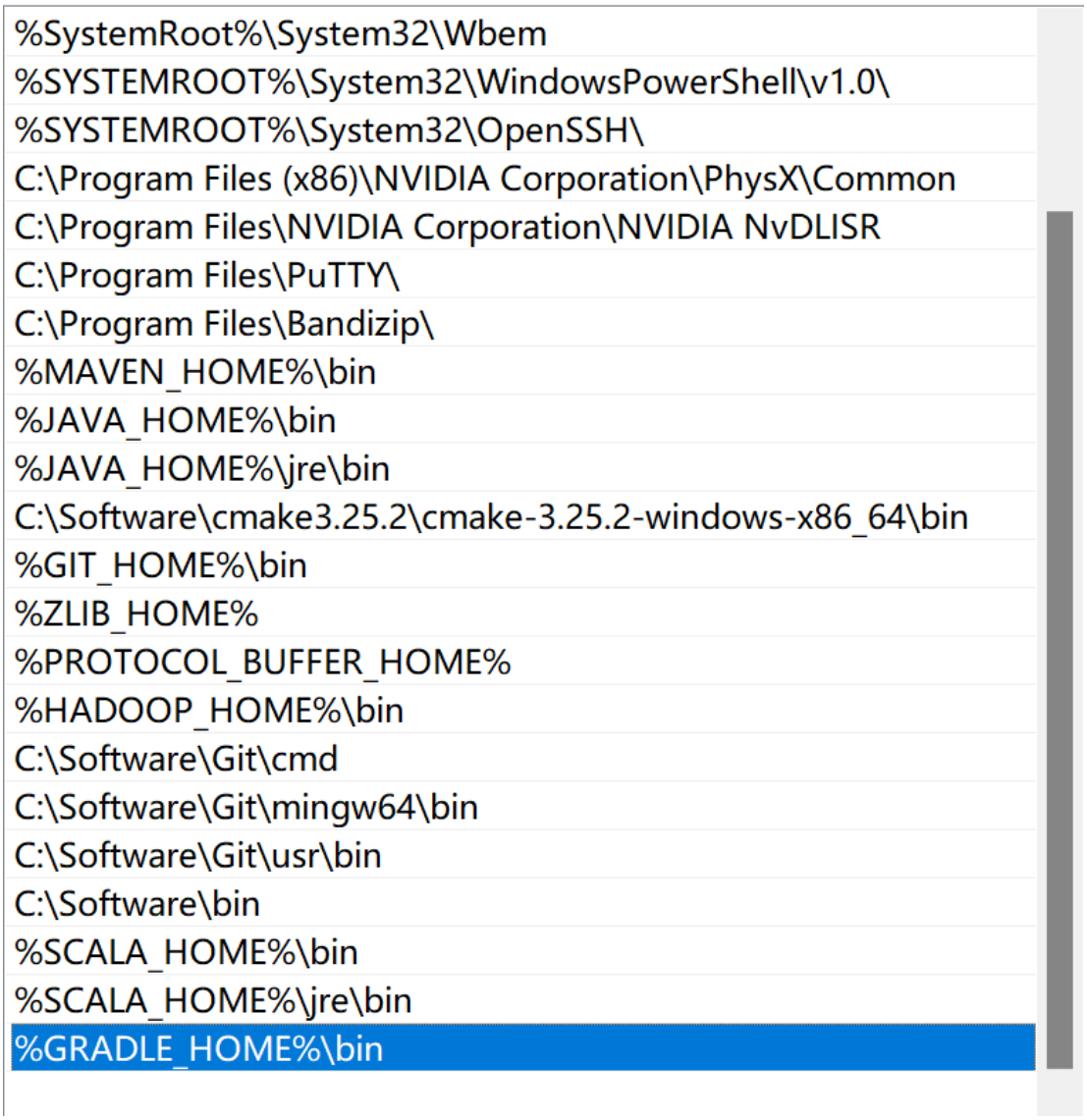
▼ 1. Install Gradle

First go to the C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.3\plugins\gradle\lib directory to confirm the gradle version compatible with the current Idea. At least version 8.4 of gradle must be installed here.

名称	修改日期	类型	大小
ant	29/01/2024 06:38	文件夹	
gradle	25/01/2024 13:01	Executable Jar File	4,959 KB
gradle-api-8.4	25/01/2024 13:01	Executable Jar File	57,982 KB
gradle-api-impldep-8.4	25/01/2024 13:01	Executable Jar File	91,900 KB
gradle-launcher-8.4	25/01/2024 13:01	Executable Jar File	1,291 KB
gradle-tooling-extension-api	25/01/2024 13:01	Executable Jar File	373 KB
gradle-tooling-extension-impl	25/01/2024 13:01	Executable Jar File	1,367 KB
gradle-wrapper-8.4	25/01/2024 13:01	Executable Jar File	130 KB

1. **URL:** https://gradle.org/releases/ →Select the corresponding binary version to download. The bin directory of the complete version may not have a startup command.
2. Configure environment variables My Computer → Properties → Advanced System Settings → Environment Variables



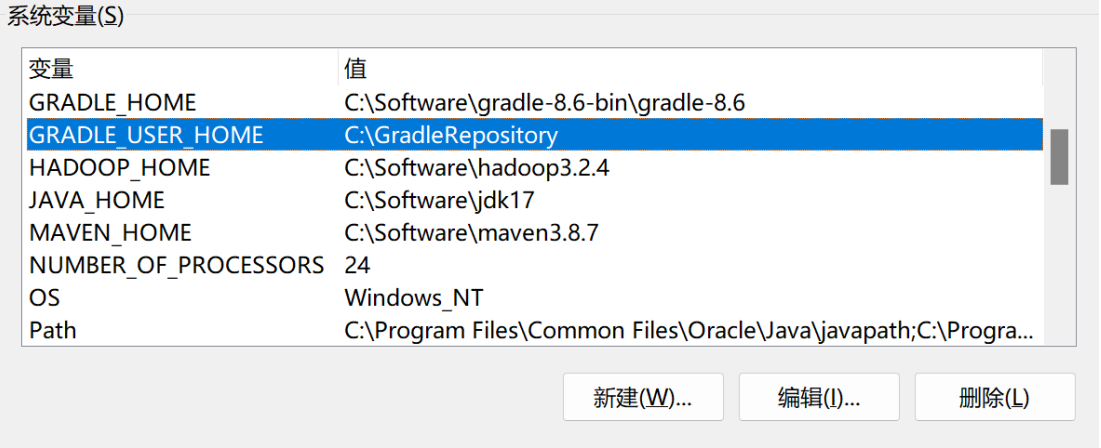


3.

```
C:\Users\ArtistS>gradle -v

Welcome to Gradle 8.6!

Here are the highlights of this release:
 - Configurable encryption key for configuration cache
 - Build init improvements
 - Build authoring improvements

For more details see https://docs.gradle.org/8.6/release-nc
```

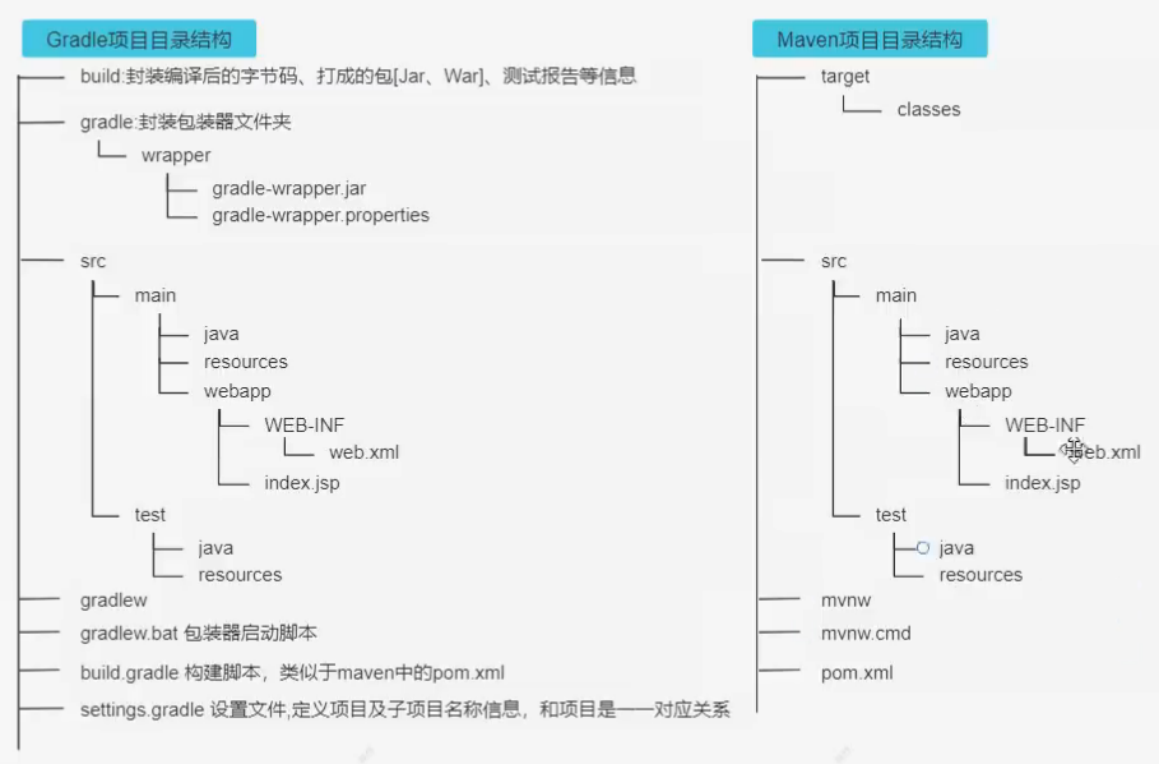
```
------------------------------------------------------------
Gradle 8.6
------------------------------------------------------------

Build time:   2024-02-02 16:47:16 UTC
Revision:     d55c486870a0dc6f6278f53d21381396d0741c6e

Kotlin:       1.9.20
Groovy:       3.0.17
Ant:          Apache Ant(TM) version 1.10.13 compiled on Ja
JVM:          17.0.10 (Oracle Corporation 17.0.10+11-LTS-24
OS:           Windows 11 10.0 amd64
```

4. Configure the gradle local warehouse, My Computer → Properties → Advanced System Settings → Environment Variables, add the following configuration



▼ 2. Gradle project directory structure

The default directory structure of the Gradle project is consistent with that of the Maven project, both based on Convention Over Configuration. The complete project directory structure is as follows:



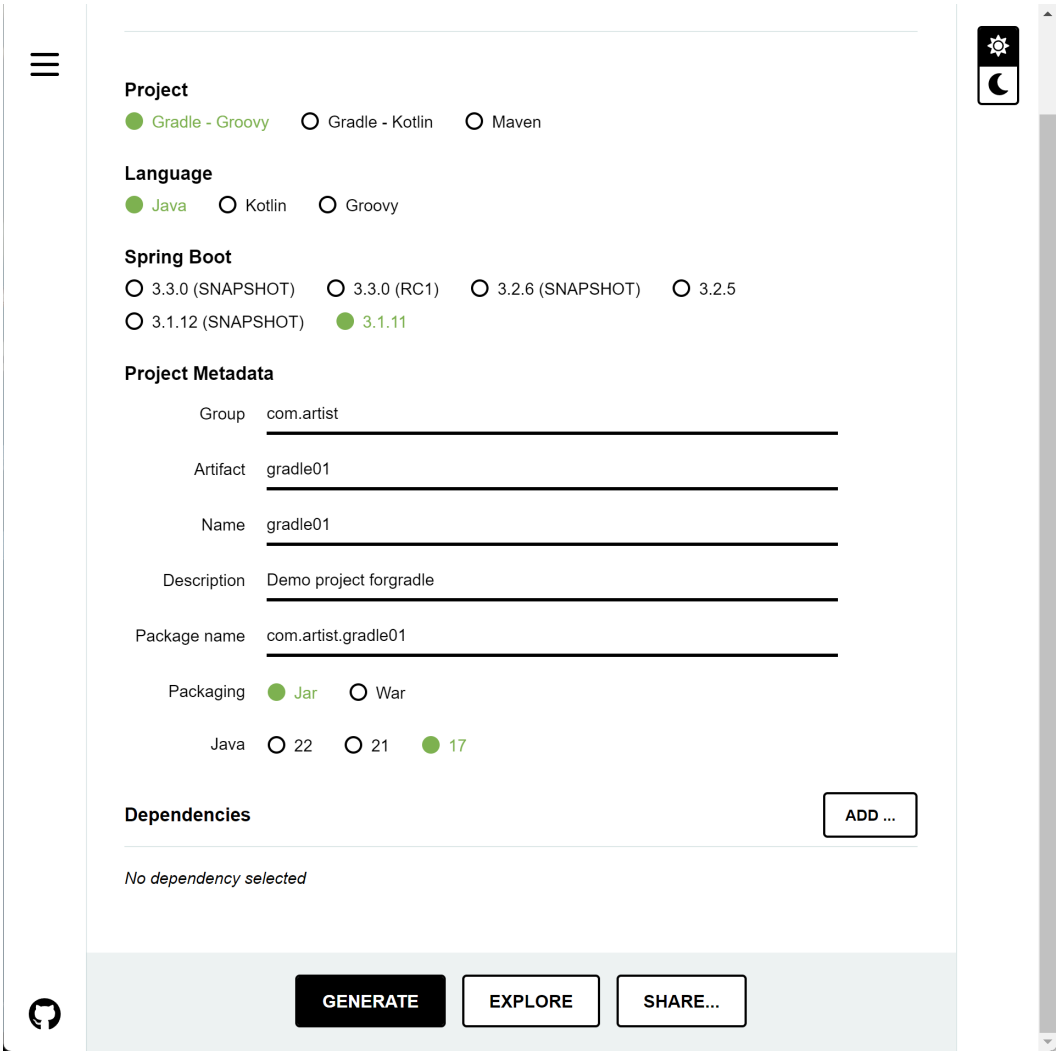
Gradle Directory	Maven Directory
build → Encapsulates compiled bytecode, packaged [Jar, War], test report and other information	target → The compiled classes files will be stored inside.
src → To put source code	src → To put source code
settings.gradle → Settings file, defines project and sub-project name information, and has a one-to-one correspondence with the project	
build.gradle → Build script, each Project (can be understood as a maven module) has a build.gradle	pom.xml → Build script, store dependencies

Tip: gradlew.bat and gradlew execute the gradle instructions in the wrapper version specified in gradle, not the locally installed gradle instructions. So if you only compile locally, there is no need for these three folders to exist.

▼ 3. Create Gradle Project

▼ 3.1 Use spring initializr to create project

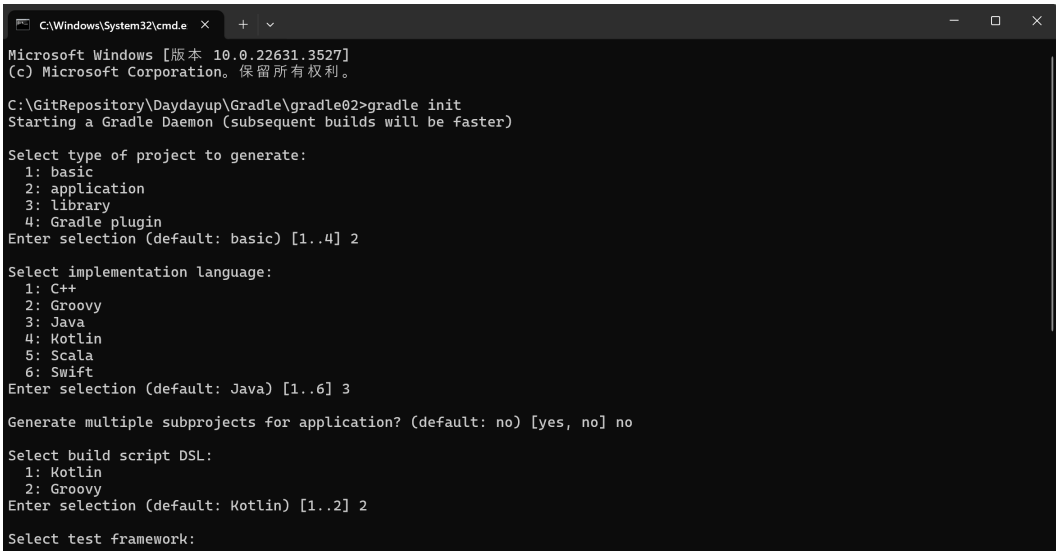
1. Open https://start.spring.io/
2. Fill in the project details



3. Drag the automatically downloaded file to the Git repository and decompress it.

▼ 3.2 Use Gradle Command to create project

1. Create a folder named gradle02 , input command `gradle init` by cmd



2. Select as needed to complete creation.

▼ 4. Gradle Command

gradle command must execute under a directory with build.gradle

Command	Description
`gradle build`	Build
`gradle build -x test`	Build without test
`gradle classes`	Compile business code and configuration file
`gradle clean`	Clear the build directory
`gradle test`	Compile test code and generate test report

▼ 5. init.d Folder

5.1 init.d folder

> We can create **xxx.gradle** file in **C:\Software\gradle-8.6-bin\gradle-8.6\init.d**, xxx.gradle file can be executed before build, so you can config some preload operations.

5.2 Create init.gradle in init.d

```
// all projects will use the following configuration
allprojects{
```

```
/*
    Dependencies require for the project will download from
    the following repositories
*/
    repositories{
    /*
        It will try to find the dependencies in maven local
        repository, this need M2_HOME environment variable
    */
        mavenLocal()
        // Third-party repository
        maven { name "Alibaba" ; url"https://maven.aliyun.c
        // Third-party repository
        maven { name "Bstek" ; url"https://nexus.bsdn.org/c
    }

    /*
        Use for build.gradle build script(e.g. plugins), if
        need some dependencies, it will download from the
        following repositories.
    */
    buildscript{
        maven { name "Alibaba" ; url"https://maven.aliyun.c
        maven { name "Bstek" ; url"https://nexus.bsdn.org/c
        maven { name "M2" ; url'https://plugins.gradle.org/
    }
}
```

5.3 How to enable init.gradle?

> **If there are more than 2 of the following methods, gradle will follow the order to execute them. If there are more than 2 init script under same folder, gradle will execute them in the order of a-z. Each init script will has a gradle instance, the methods and properties you called in the init script, will delegate to this gradle instance.**

1. Use command line

```
# You can enter this command multiple times to specify m
gradle --init-script [DIR_PATH]/init.gradle -q [TASK_NAM
```

2. Put **init.gradle** file into **[USER_HOME]/.gradle/**

   e.g. C:\Users\ArtistS\.gradle

3. Put **xxx.gradle** into **[USER_HOME]/.gradle/init.d/**
4. Put **xxx.gradle** into **[GRADLE_HOME]/init.d/**

   e.g. C:\Software\gradle-8.6-bin\gradle-8.6\init.d

5.4 Repository instructions

mavenLocal() → Gradle will find the repository by the repository path in maven settings.xml. The order in which gradle searches for jar packages is as follows:

[USER_HOME]/.m2/settings.xml → **[M2_HOME]/conf/settings.xml** → **[USER_HOME]/.m2/repository**

maven{[URL address]} → e.g. private repository, alibaba repository

mavenCentral() → Maven central repository, no need to config, you can use it by directly declaring it

> Gradle can avoid downloading from the remote repository every time by combining the specified repository and remote repository. But here is a problem, if the local maven repository has this dependency, gradle will load it directly. But if the local maven doesn't have this dependency, Gradle will download it from a remote repository. Keep in mind, that this jar download from the remote repository will not be stored in the maven

> repository, it will be put into the cache directory, the default path is **[USER_HOME]/.gradle/cashes**. If you didn't configure the **GRADLE_USER_HOME** environment variable. It will be put into **[GRADLE_USER_HOME]/cashes**. There is no other way to put the downloaded jar into Maven repository, because the format of the jar downloaded in cashes folder is different from the jar stored in the maven repository.

Repository URL: https://developer.aliyun.com/mvn/guide