Settings.gradle

Settings文件中可以作为一个根配置，包含自模块，并描述这些子模块之间的关系

在构建的初始化阶段gradle会读取这个文件，如果完全找不到这些指定的子模块，则gradle会将这个项目当作单独的模块项目来构建

既然gradle初始化会读取这个文件，那么我们可以将一些初始化的逻辑放在这里

如：

def initFC(){

println '================='

}

initFC()

使用rootProj.name指定工程的名称

rootProject.name = 'resources' #工程名称

include指令将多个模块入到根模块，可以单独加入也可以以数组的形式加入

如：

include 'portal','service','dao'

include 'model'

重命名项目名（先定义原项目名）

findProject(':portal')?.name = 'web'

**配置示例：**

//设置工程的名字

rootProject.name = 'Demo'

include 'portal','service','dao' //可以以数组的形式，该命令会相对于当前目录查找这些子模块

include 'model' //也可以以单个子模块的形式

include 'utils'

findProject(':portal')?.name = 'web'

def initFC(){

println '+----------------------------------+'

println '+ init all projects +'

println '+----------------------------------+'

}

initFC()

Build.gradle

* 根项目build.gradle的配置，一般用于多项目中，对所有的项目配置通用的配置）

1. 指定报名，group
2. 指定版本，version
3. 指出应用的插件，apply plugin
4. 指定编译java文件的jdk版本以及，sourceCompatibility
5. 可运行class文件的jvm的版本要求，targetCompatibility
6. 任务执行通用参考项，tasks.withType,tasks.xxxx（如编码格式等，另外上面的某些命令也可以放在这里）
7. 通用仓库
8. 通用依赖

**配置示例**

allprojects {//配置工程和它的所有子项目,配置所有项目的通用配置，如果没有子项目的话，则配置的就是当前项目

// 1. 指定报名，group

// 2. 指定版本，version

// 3. 指出应用的插件，apply plugin

// 4. 指定编译java文件的jdk版本以及，sourceCompatibility

// 5. 可运行class文件的jvm的版本要求，targetCompatibility

// 6. 任务执行通用参考项，tasks.withType,tasks.xxxx（如编码格式等，另外上面的某些命令也可以放在这里）

// 7. 通用仓库

// 8. 通用依赖

group 'com.test.gradle'

version '1.0-SNAPSHOT'

apply plugin: 'java' //可以使用compile和testCompile指令

apply plugin: 'maven'

apply plugin: 'idea'

apply plugin: 'eclipse'

apply plugin: 'groovy'

apply plugin: 'visual-studio'

sourceCompatibility = 1.8 //指定jkd的版本

targetCompatibility = 1.8 //指定运行class文件的版本

//设置全局编译器的版本

// tasks.withType(JavaCompile) {

// sourceCompatibility = JavaVersion.VERSION\_1\_7

// targetCompatibility = JavaVersion.VERSION\_1\_7

// }

tasks.withType(JavaCompile) {

options.encoding = "UTF-8"

}

repositories{ //通用仓库配置

maven { url 'http://maven.aliyun.com/nexus/content/groups/public/' }

//配置本地仓库

maven { url 'http://192.168.128.1:11022/repository/maven-central/' }

mavenCentral()

}

dependencies{//通用依赖配置

testCompile 'junit:junit:4.11'

}

}

allprojects {

group 'com.htiiot.resources' #包名，类似添加依赖时的group

version '0.0.1' #版本，类似添加依赖时的版本

apply plugin: 'maven' #maven插件

apply plugin: 'eclipse' #eclipse插件

apply plugin: 'java' #java插件

sourceCompatibility = 1.8 #指定jdk版本

targetCompatibility = 1.8 #指定运行class文件的jvm的要求

tasks.withType(JavaCompile) { #任务

options.encoding = 'UTF-8'

}

repositories { #通用依赖仓库

maven { url 'http://maven.aliyun.com/nexus/content/groups/public/' }

maven { url 'http://192.168.207.21:62125/repository/maven-central/' }

maven { url 'http:/192.168.207.21:62125/repository/maven-snapshots/' }

mavenCentral()

}

configurations.all {

// check for updates every build

resolutionStrategy.cacheChangingModulesFor 0,'seconds'

}

dependencies { #通用依赖

testCompile group: 'junit', name: 'junit', version: '4.11'

}

}

子模块配置：

1. 子模块依赖配置，dependencies，配置中可能需要：

exclude解决冲突的包

compile project 依赖同项目组工程内的其他项目

1. 子模块任务,task
2. 子模块架构，archive
3. 子模块其他配置，如上传jar包(uploadArchives)，

如dao模块：

dependencies {

testCompile group: 'junit', name: 'junit', version: '4.12'

compile group: 'mysql', name: 'mysql-connector-java', version: '5.1.38'

compile('com.fasterxml.jackson.core:jackson-databind:2.8.6')

compile('org.apache.hbase:hbase-client:1.1.2') {

exclude group: 'org.slf4j'

exclude group: 'org.apache.httpcomponents'

exclude group: 'log4j'

exclude group: 'org.apache.hadoop'

}

compile('org.apache.hadoop:hadoop-common:2.7.1') {

exclude group: 'javax.servlet'

}

compile project(':model')

compile project(':utils')}

如，service模块

dependencies {

testCompile group: 'junit', name: 'junit', version: '4.12'

compile('com.fasterxml.jackson.core:jackson-databind:2.8.6')

compile group: 'redis.clients', name: 'jedis', version: '2.9.0'

compile(':dao')

compile(':model')

compile(':utils')

}

uploadArchives {

repositories {

mavenDeployer {

repository(url: "http://192.168.207.21:62125/repository/maven-snapshots") {

authentication(userName: "uploader", password: "uploader123")

}

pom.version = "${version}-SNAPSHOT"

pom.artifactId = "visual-shared"

pom.groupId = "com.xxx.xxx"

}

}

}