Gradle_learning (Basics)

Tuesday, November 7, 2017

What's gradle?

- Build by convention - Gradle has building conventions (Overridable) for easy understanding across the team
- Uses Groovy DSL (Build land: Domain Specific Language - written in Groovy)
- Supports dependencies : Ivy and Maven
- Supports multi project builds (Out of the box)
- Supports projects of Java, JS, C++ etc
- Easily customizable
- Declarative Build Language:
    - Expresses the intent of the build - Tell gradle what we wud like to happen, not how we'd like it to happen: Build script is easily understandable
    - Maintainable, understandable, readable

Installing gradle:

- Windows: https://gradle.org/
    - Download and install
        - Mitigate handling versions:
        - Junction gradle <gradleInstallationPath> - This will create a virtual link from a folder called gradle to the physical file
        - We can cd to gradle now, and do other stuffs
        - Set Path variable to *\**gradle**\bin (A virtual directory created just now)
        - Typein gradle in the cmd prompt, it should give some help info
- Linux: gvm - groovy environment manager
    - curl -s get.gvmtool.net | bash
    - source "<downloadDir>/.gvm/bin/gvm-init.sh"
    - gvm help - Gives a list of candidates that we can install
    - gvm install gradle
    - gradle - Gives out gradle help

Gradle builds:

- Gradle has a build file, typically build.gradle
- This contains :
    - Plugins
    - Dependencies
    - Tasks (Mainly)
- build.gradle, Groovy code (A simple one)
    - **build.gradle**:

```
task hello {
    doLast {
        println "Hello task"
    }
}
```

    - Tasks have lifecycle - example, configuration phase, initialization phase, etc. They have methods called doFirst, doLast..

- **Typein gradle hello from cmd (Location - where the above build.gradle is located)**

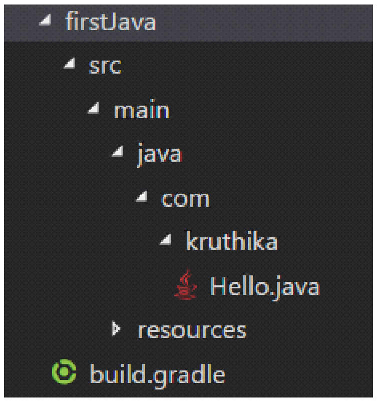
```
>gradle hello
:hello
Hello task

BUILD SUCCESSFUL

Total time: 1.152 secs
```

<u>**Building a simple java project**</u>

- **build.gradle**:

```
apply plugin:"java"
```

- **Create a folder structure like this:**


```
firstJava
  src
    main
      java
        com
          kruthika
            Hello.java
      resources
  build.gradle
```

And place a simple java file, Hello.java in it:

```
public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello");
    }
}
```

- gradle task - gives out details on types of tasks that can be executed:

```
>gradle task
:tasks

------------------------------------------------------------
All tasks runnable from root project
------------------------------------------------------------

Build tasks
-----------
assemble - Assembles the outputs of this project.
build - Assembles and tests this project.
buildDependents - Assembles and tests this project and all projects that depend on it.
buildNeeded - Assembles and tests this project and all projects it depends on.
classes - Assembles main classes.
clean - Deletes the build directory.
```

```
jar - Assembles a jar archive containing the main classes.
testClasses - Assembles test classes.

Build Setup tasks
-----------------
init - Initializes a new Gradle build.
wrapper - Generates Gradle wrapper files.

Documentation tasks
-------------------
javadoc - Generates Javadoc API documentation for the main source code.

Help tasks
----------
buildEnvironment - Displays all buildscript dependencies declared in root project
'firstJava'.
components - Displays the components produced by root project 'firstJava'. [incubating]
dependencies - Displays all dependencies declared in root project 'firstJava'.
dependencyInsight - Displays the insight into a specific dependency in root project
'firstJava'.
dependentComponents - Displays the dependent components of components in root
project 'firstJava'. [incubating]
help - Displays a help message.
model - Displays the configuration model of root project 'firstJava'. [incubating]
projects - Displays the sub-projects of root project 'firstJava'.
properties - Displays the properties of root project 'firstJava'.
tasks - Displays the tasks runnable from root project 'firstJava'.

Verification tasks
------------------
check - Runs all checks.
test - Runs the unit tests.

Rules
-----
Pattern: clean<TaskName>: Cleans the output files of a task.
Pattern: build<ConfigurationName>: Assembles the artifacts of a configuration.
Pattern: upload<ConfigurationName>: Assembles and uploads the artifacts belonging to a
configuration.

To see all tasks and more detail, run gradle tasks --all

To see more detail about a task, run gradle help --task <task>

BUILD SUCCESSFUL

Total time: 1.176 secs
```

- **Typein gradle build from cmd prompt (Location - where build.gradle is present (in**

firstJava folder))

```
>gradle build
:compileJava
:processResources NO-SOURCE
:classes
:jar
:assemble
:compileTestJava NO-SOURCE
:processTestResources NO-SOURCE
:testClasses UP-TO-DATE
:test NO-SOURCE
:check UP-TO-DATE
:build

BUILD SUCCESSFUL

Total time: 1.746 secs
```

- **<u>Gradle Wrappers</u>**
  - build.gralde:

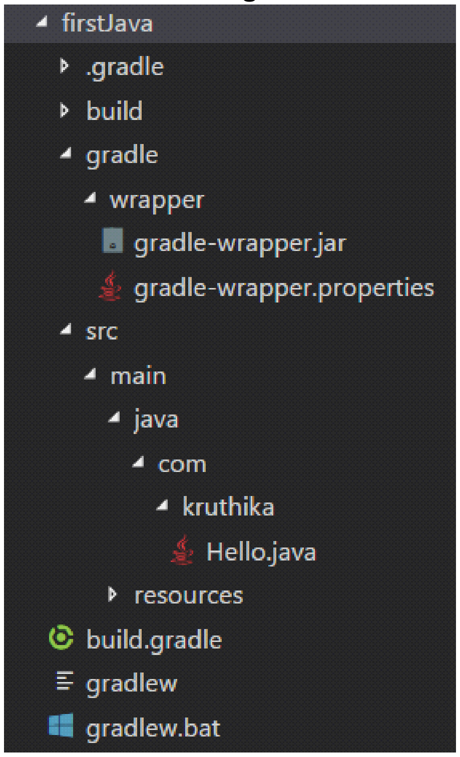
```
apply plugin:"java"

task wrapper(type: Wrapper) {
	gradleVersion = '3.5'
}

>gradle build
:compileJava
:processResources NO-SOURCE
:classes
:jar
:assemble
:compileTestJava NO-SOURCE
:processTestResources NO-SOURCE
:testClasses UP-TO-DATE
:test NO-SOURCE
:check UP-TO-DATE
:build

BUILD SUCCESSFUL

Total time: 3.151 secs
```

  - **gradle wrapper**

```
>gradle wrapper
:wrapper

BUILD SUCCESSFUL
```

Total time: 1.501 secs

- We can observe a gradlew file in the directory



```
firstJava
  .gradle
  build
  gradle
    wrapper
      gradle-wrapper.jar
      gradle-wrapper.properties
  src
    main
      java
        com
          kruthika
            Hello.java
      resources
  build.gradle
  gradlew
  gradlew.bat
```

- Do a gradle**w** build
  It downloads the specified version of gradle (3.5 here) and then builds the project
  This comes in very handy to live out of "difference in version of build scripts" related problems while building.
  The first time, we execute, it downloads. From the next time, it picks it up from the download location.
  >gradlew build
  Downloading https://services.gradle.org/distributions/gradle-3.5-bin.zip

```
.........................................................................................................
.........................................................................................................
.........................................................................................................
.........................................................................................................
.........................................................................................................
.........................................................................................................
.........................................................................................................
.........................................................................................................
.........................................................................................................
.........................................................................................................
.........................................................................................................
.........................................................................................................
.........................................................................................................
.........................................................................................................
.........................................................................................................
.........................................................................................................
.........................................................................................................
.........................................................................................................
```

```
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
.......................................
Unzipping C:\Users\kruthis\.gradle\wrapper\dists\gradle-3.5-bin
\daoimhu7k5rlo48ntmxw2ok3e\gradle-3.5-bin.zip to C:\Users\kruthis\.gradle\wrapper
\dists\gradle-3.5-bin\daoimhu7k5rlo48ntmxw2ok3e
:compileJava UP-TO-DATE
:processResources NO-SOURCE
:classes UP-TO-DATE
:jar UP-TO-DATE
:assemble UP-TO-DATE
:compileTestJava NO-SOURCE
:processTestResources NO-SOURCE
:testClasses UP-TO-DATE
:test NO-SOURCE
:check UP-TO-DATE
:build UP-TO-DATE

BUILD SUCCESSFUL

Total time: 5 mins 46.397 secs
```

>gradlew build

```
:compileJava UP-TO-DATE
:processResources NO-SOURCE
:classes UP-TO-DATE
:jar UP-TO-DATE
:assemble UP-TO-DATE
:compileTestJava NO-SOURCE
:processTestResources NO-SOURCE
:testClasses UP-TO-DATE
:test NO-SOURCE
:check UP-TO-DATE
:build UP-TO-DATE

BUILD SUCCESSFUL

Total time: 1.324 secs
```

Basic gradle tasks

- Task is the code that gradle executes.
  - It has a lifecycle
    Initialization phase
    Configuration phase

    Execution phase
    - It has properties - are to be configured during configuration phase of the lifecycle
    - It has actions
        - First action
        - Last action
    - It has dependencies
- Gradle uses groovy (a JVM language - object oriented)
- Build.gradle:

```
project.task("Task1")
project.task("Task2")
task "Task3"
task Task4
Task1.description = "This is task 1's desc"
Task4.description = "Lol"
```

```
> gradle tasks --all
:tasks

------------------------------------------------------------
All tasks runnable from root project
------------------------------------------------------------

Build Setup tasks
-----------------
init - Initializes a new Gradle build.
wrapper - Generates Gradle wrapper files.

Help tasks
----------
buildEnvironment - Displays all buildscript dependencies declared in root project
'gradleTasks'.
components - Displays the components produced by root project 'gradleTasks'.
[incubating]
dependencies - Displays all dependencies declared in root project 'gradleTasks'.
dependencyInsight - Displays the insight into a specific dependency in root project
'gradleTasks'.
dependentComponents - Displays the dependent components of components in root
project 'gradleTasks'. [incubating]
help - Displays a help message.
model - Displays the configuration model of root project 'gradleTasks'. [incubating]
projects - Displays the sub-projects of root project 'gradleTasks'.
properties - Displays the properties of root project 'gradleTasks'.
tasks - Displays the tasks runnable from root project 'gradleTasks'.

Other tasks
-----------
Task1 - This is task 1's desc
Task2
```

```
Task3
Task4 - Lol

BUILD SUCCESSFUL

Total time: 1.176 secs
```

Gradle knows project is a top level objects and it delegates everything to the object (project object)

Running gradle tasks

Build.gradle

```
project.task("Task1")
project.task("Task2")
task "Task3"
task Task4
Task1.description = "This is task 1's desc"
Task4.description = "Lol"
Task1.doLast {
println "This is my first gradle task"
}
Task2 << {
println "Another way to do task is <taskname> << {//Task code}"
}
```

```
> gradle Task1
:Task1
This is my first gradle task

BUILD SUCCESSFUL

Total time: 1.307 secs
```

```
> gradle Task2
The Task.leftShift(Closure) method has been deprecated and is scheduled to be removed in
Gradle 5.0. Please use Task.doLast(Action) instead.
        at build_wbh8z7bxmrkibvgv3p6ffhfo.run(D:\ExperimentZone\Gradle_learning\gradleTasks
\build.gradle:13)
:Task2
Another way to do task is <taskname> << {//Task code}

BUILD SUCCESSFUL

Total time: 1.448 secs
```

The above method is deprecated in newer gradle versions

Build phases:

1. Initialization: Used to configure multi project builds
2. Configuration phase: Executes the code in the task that not the action
3. Execution phase: Executes the task actions

Task dependencies

```
project.task("Task1")
project.task("Task2")
task "Task3"
task Task4
Task1.description = "This is task 1's desc"
Task4.description = "Lol"
Task1.doLast {
println "This is my first gradle task"
}
Task2 << {
println "Another way to do task is <taskname> << {//Task code} This
deprecated though"
}
task Task5 {
description = "Task5 description"
doFirst {
println "First action"
}
doLast {
println "Last action"
}
}
Task5.doFirst {
println "Another first"
}
Task5.doLast {
println "Another last"
}
task Task6 {
description = "This task depends on Task5"
doLast {
println "Task6 executed"
}
dependsOn Task5
}
Task5.dependsOn Task1
Task5.dependsOn Task2
Task2.dependsOn Task3
```

```
> gradle Task6
The Task.leftShift(Closure) method has been deprecated and is scheduled to be removed in
Gradle 5.0. Please use Task.doLast(Action) instead.
	at build_wbh8z7bxmrkibvgv3p6ffhfo.run(D:\ExperimentZone\Gradle_learning\gradleTasks
\build.gradle:13)
:Task1
```

```
This is my first gradle task
:Task3 UP-TO-DATE
:Task2
Another way to do task is <taskname> << {//Task code} This deprecated though
:Task5
Another first
First action
Last action
Another last
:Task6
Task6 executed

BUILD SUCCESSFUL

Total time: 1.208 secs
```

Create dependencies however you want, except circular dependencies.

Defining variables in gradle

- def varName = "6"
  Usage: $varName
- Using extended properties, to use the variable across projects:
  ext.varName = "4"
  Usage: $varName within the file
  This is equivalent to project.ext.varName = "4"
  Usage equivalents: $project.ext.varName
- Example:

  ```
  def var1 = "1"
  ext.var2 = "2"
  project.ext.var3 = "3"
  task PrintVariables {
  doLast {
  println "$var1 $var2 $var3"
  }
  }
  ```

  ```
  > gradle PrintVariables
  :PrintVariables
  1 2 3

  BUILD SUCCESSFUL

  Total time: 1.266 secs
  ```

mustRunAfter

If 2 tasks execute, one must run after the other.
If there are circular dependencies, gradle will throw an error

shouldRunAfter

This ignores circular dependencies.
finalizeBy

*/adventure/kruthikacs/learnings