How to create a java project with gradle in eclipse

Posted on April 19, 2017

In this tutorial, we will learn how to create a java application using the Gradle build tool and Eclipse IDE.

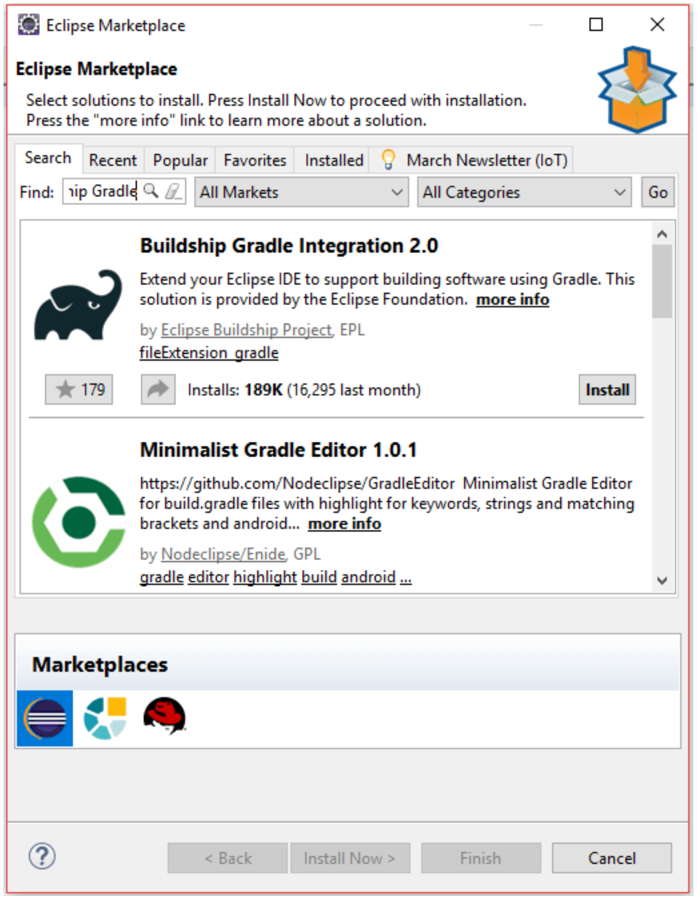
Tools and technologies used-

- JDK1.8.0_71
- Eclipse Neon.3
- Buildship, Eclipse Plug-ins for Gradle
- Gradle 3.4
- JUnit 4.12

Let's begin step by step to see how a gradle java project is created in eclipse.

Install Gradle plug-ins (Buildship)

You can install the Buildship gradle tool from Eclipse Marketplace. Go to `Help` → `Eclipse Marketplace…` menu to install the Buildship gradle tool.



Eclipse Marketplace

Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.

Find: ıip Gradle | All Markets | All Categories

Buildship Gradle Integration 2.0

Extend your Eclipse IDE to support building software using Gradle. This solution is provided by the Eclipse Foundation. **more info**

by Eclipse Buildship Project, EPL

fileExtension gradle

179 Installs: **189K** (16,295 last month)

Minimalist Gradle Editor 1.0.1

https://github.com/Nodeclipse/GradleEditor Minimalist Gradle Editor for build.gradle files with highlight for keywords, strings and matching brackets and android... **more info**

by Nodeclipse/Enide, GPL

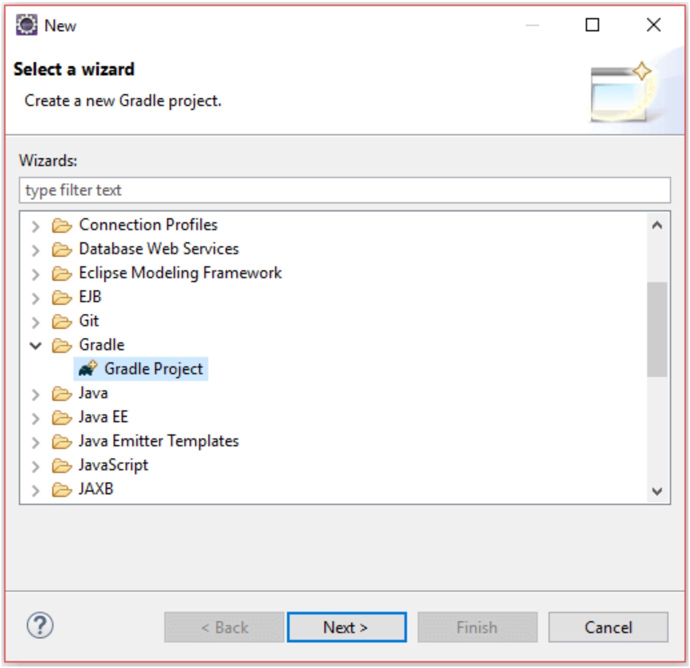
gradle editor highlight build android ...

Marketplaces

Create Gradle project

In Eclipse, go to `File` → `New` → `Other…` menu path to select new project.

In New wizard, select `Gradle Project` and click on `Next` .



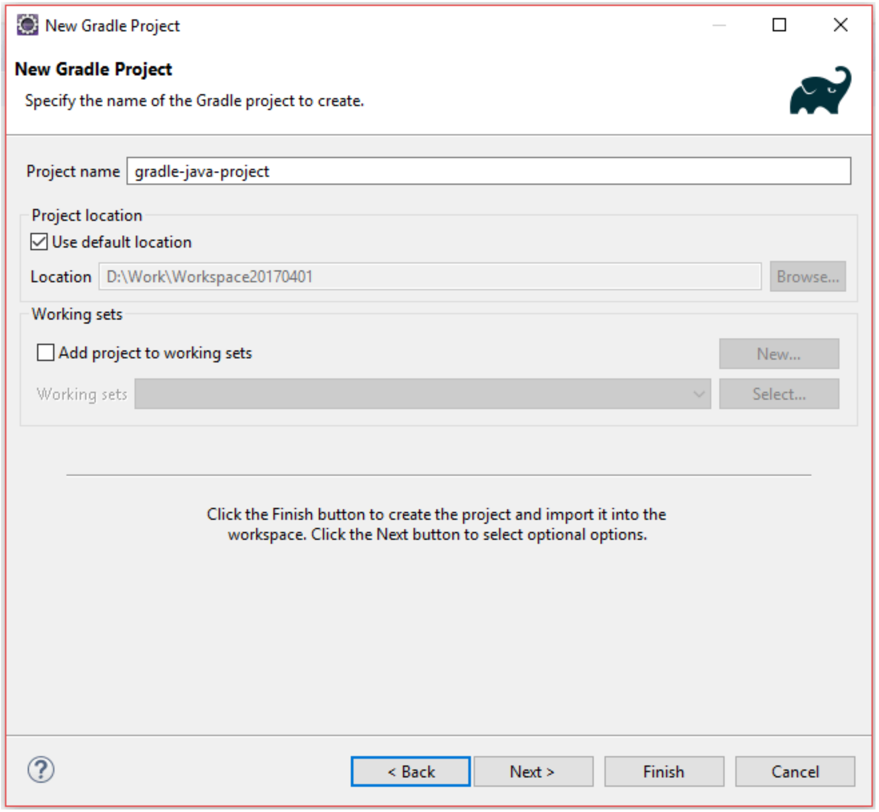
New

Select a wizard

Create a new Gradle project.

Wizards:

In next wizard, type the `Project Name` and click on `Next` .



New Gradle Project

New Gradle Project

Specify the name of the Gradle project to create.

Project name: gradle-java-project

Project location

☑ Use default location

Location: D:\Work\Workspace20170401

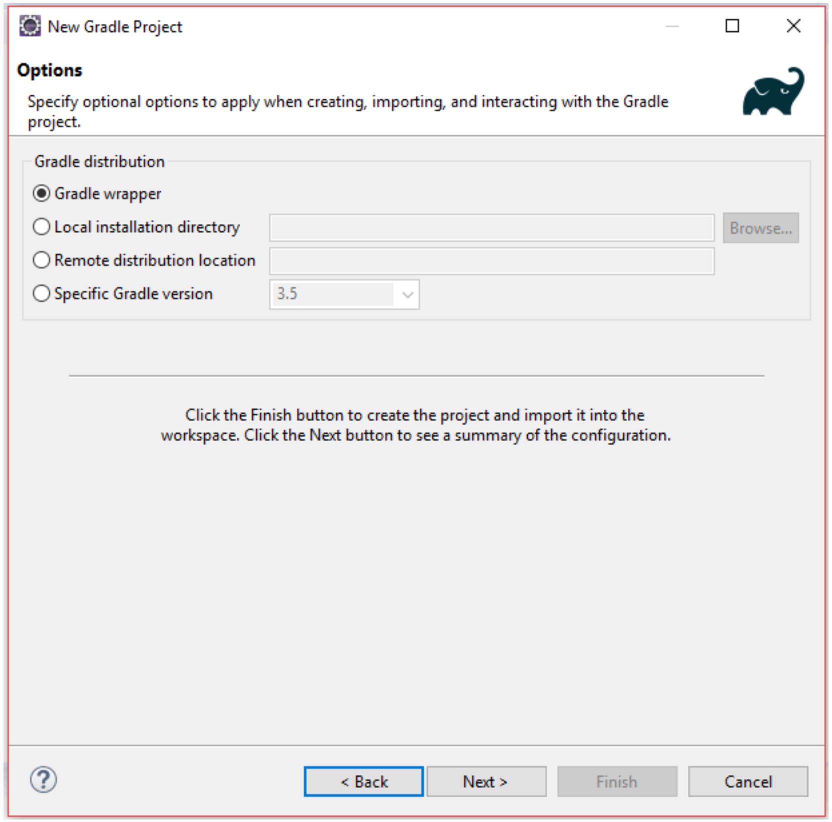
Working sets

☐ Add project to working sets

Working sets:

Click the Finish button to create the project and import it into the workspace. Click the Next button to select optional options.

In next wizard, select gradle distribution and click on `Next` .



Options

Specify optional options to apply when creating, importing, and interacting with the Gradle project.

Gradle distribution

● Gradle wrapper

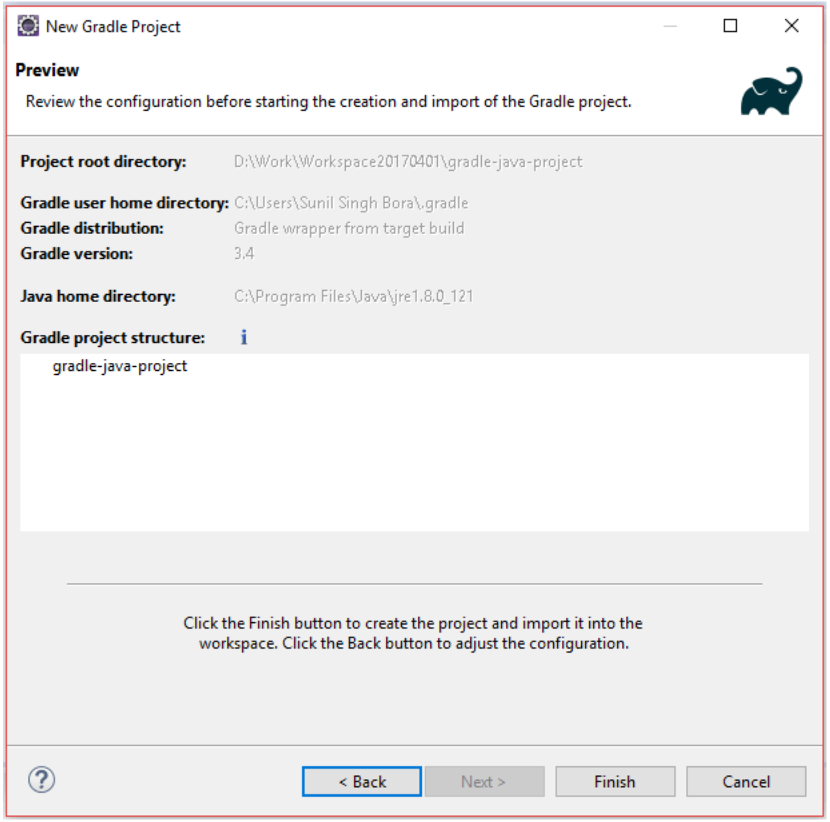
○ Local installation directory

○ Remote distribution location

○ Specific Gradle version 3.5

Click the Finish button to create the project and import it into the workspace. Click the Next button to see a summary of the configuration.

In next wizard, preview the project configuration and click on `Finish` .



Preview

Review the configuration before starting the creation and import of the Gradle project.

Project root directory: D:\Work\Workspace20170401\gradle-java-project

Gradle user home directory: C:\Users\Sunil Singh Bora\.gradle

Gradle distribution: Gradle wrapper from target build

Gradle version: 3.4

Java home directory: C:\Program Files\Java\jre1.8.0_121

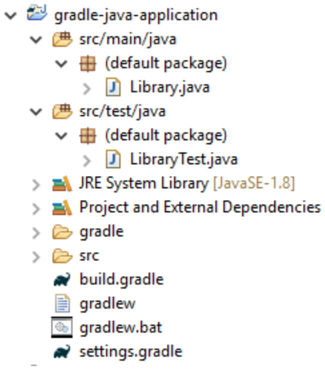
Gradle project structure:

gradle-java-project

Click the Finish button to create the project and import it into the workspace. Click the Back button to adjust the configuration.

Review project structure

Review the following project structure generated by the Gradle build tool.



```
gradle-java-application
  src/main/java
    (default package)
      Library.java
  src/test/java
    (default package)
      LibraryTest.java
  JRE System Library [JavaSE-1.8]
  Project and External Dependencies
  gradle
  src
  build.gradle
  gradlew
  gradlew.bat
  settings.gradle
```

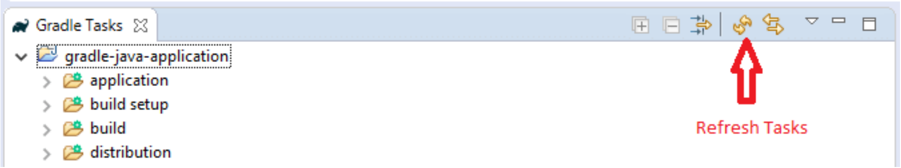
Gradle build file - build.gradle

Open `build.gradle` file and add the following code in it.

build.gradle

```
apply plugin: 'java'
apply plugin: 'application'

repositories {
    jcenter()
}

dependencies {
   compile 'com.google.guava:guava:20.0'  // Google Guava dependency
   testCompile 'junit:junit:4.12'         // JUnit dependency for testing
}


mainClassName = 'com.boraji.tutorial.MainApp'  // Main class with main method
```

Refresh the *Gradle Tasks* view after changes in the `build.gradle` file as follows.



Create Java programs

Remove the java files, generated by the gradle build tool, from project and create new files as follows.

Create `MainApp.java` file under `src/main/java` folder and write the following code in it.

MainApp.java

```
package com.boraji.tutorial;

/**
 * @author imssbora
 *
 */
public class MainApp {

   public String sayHello() {
      return "Hello Gradle";
   }

   public static void main(String[] args) {
      MainApp app = new MainApp();
      System.out.println(app.sayHello());
   }
}
```

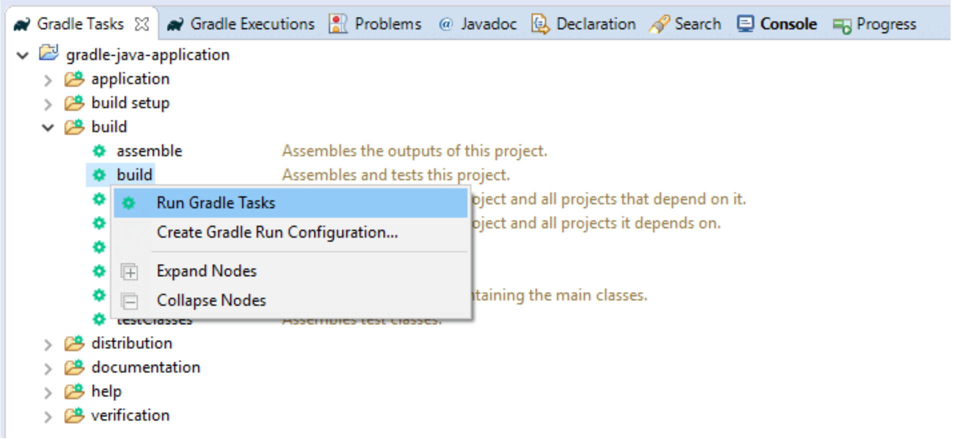
Create `MainAppTest.java` file under `src/test/java` folder and write the following *JUnit* testcase in it.

MainAppTest.java

```
package com.boraji.tutorial;

import org.junit.Test;
import static org.junit.Assert.*;

/**
 * @author imssbora
 *
 */
public class MainAppTest {

   @Test
   public void testSayHello() {
      MainApp app = new MainApp();
      assertNotNull("Success", app.sayHello());
   }
}
```

Build project

In *Gradle Tasks* view/tab, right click on the `build` task and select `Run Gradle Tasks` to build the java project.



Console output-

```
Gradle Tasks: build

:compileJava FAILED

FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':compileJava'.
> Could not find tools.jar. Please check that C:\Program Files\Java\jre1.8.0_121 contains a valid JDK installation.

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output.

BUILD FAILED

Total time: 20.754 secs
```

As you can see in the console output, the project build is failed because the gradle build tool did not find the JDK environment.

To resolve the above error, create `gradle.properties` file under project's root folder and write the following property (i.e. java home path) in it.

gradle.properties

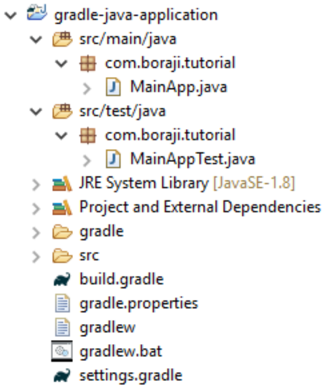
```
org.gradle.java.home=C:\\Program Files\\Java\\jdk1.8.0_71
```

Now execute the `build` task again. After the build is finished, the console output looks like as follows.

```
Gradle Tasks: build

:compileJava
:processResources NO-SOURCE
:classes
:jar
:startScripts
:distTar
:distZip
:assemble
:compileTestJava
:processTestResources NO-SOURCE
:testClasses
:test
:check
:build

BUILD SUCCESSFUL

Total time: 4.911 secs
```

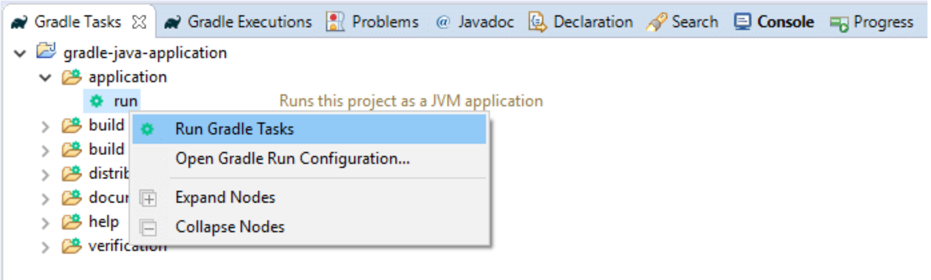
Review final project Structure

The final project structure of gradle java project will look like as follows.



Run project

In *Gradle Tasks* view, right click on the `run` task and select `Run Gradle Tasks` to run the java project.

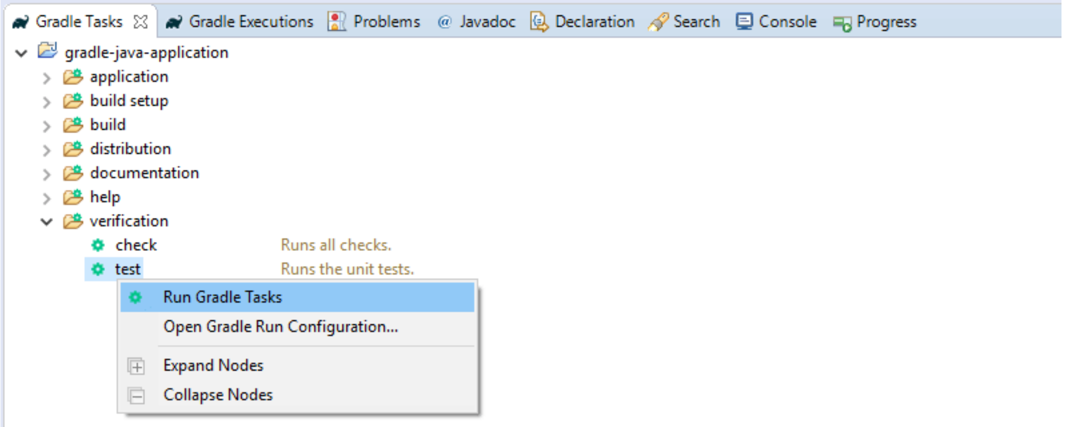


Console output-

```
Gradle Tasks: run

:compileJava UP-TO-DATE
:processResources NO-SOURCE
:classes UP-TO-DATE
:run
Hello Gradle

BUILD SUCCESSFUL

Total time: 0.758 secs
```

Run JUnit testcase

In *Gradle Tasks* view, right click on the `test` task and select `Run Gradle Tasks` to run the JUnit testcase.



Console output-

```
Gradle Tasks: test

:compileJava UP-TO-DATE
:processResources NO-SOURCE
:classes UP-TO-DATE
:compileTestJava UP-TO-DATE
:processTestResources NO-SOURCE
:testClasses UP-TO-DATE
:test UP-TO-DATE

BUILD SUCCESSFUL

Total time: 0.277 secs
```

Related Posts

- How to create a java project with gradle in eclipse (/how-to-create-a-gradle-java-project-in-eclipse)
- How to convert java project to maven project in eclipse (/how-to-convert-java-project-to-maven-project-in-eclipse)
- How to create a maven project in eclipse (/how-to-create-a-maven-project-in-eclipse)
- How to download sources and JavaDoc with maven in Eclipse (/how-to-download-sources-and-javadoc-with-maven-in-eclipse)
- How to create a web project using maven in eclipse (/how-to-create-a-web-project-using-maven-in-eclipse)
- Gradle - How to install Gradle on Window (/gradle-how-to-install-gradle-on-window)

2 Comments **BORAJI.COM** 1 **Login**

♡ **Recommend** 1 Tweet Share **Sort by Best**

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Madhumita V • a year ago

Very useful. Thanks :)

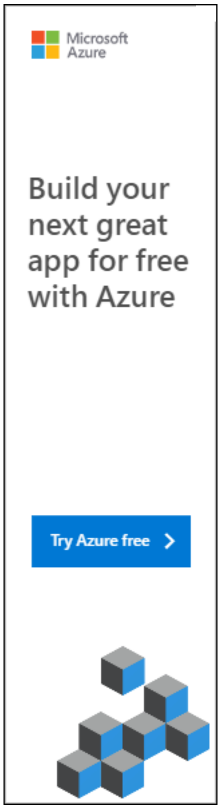
^ | ˅ • Reply • Share ›

SEER • 2 years ago • edited

Helpful.

^ | ˅ • Reply • Share ›

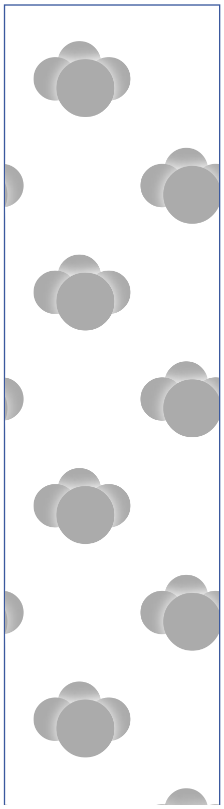
✉ **Subscribe** **Add Disqus to your siteAdd DisqusAdd** **Disqus' Privacy PolicyPrivacy PolicyPrivacy**



- Spring Frameworks ▾ ()
- Hibernate ORM (/category/hibernate-orm)
- Log4j 2 (/category/log4j-2)
- JFreeChart (/category/jfreechart)
- Core Java ▾ ()
- Misc ▾ ()

Recent Post

- iText 7 - How to add an Image to a PDF document (/itext-7-how-to-add-an-image-to-a-pdf-document)
- iText7 - Hello world example (/itext7-hello-world-example)
- Java - How to find free, total disk space in Window or Unix (/java-how-to-find-free-total-disk-space-in-window-or-unix)
- Java - How to post XML request using HttpURLConnection (/java-how-to-post-xml-request-using-httpurlconnection)
- Hibernate 5 - Named query example (/hibernate-5-named-query-example)
- Hibernate 5 - Native SQL query example (/hibernate-5-native-sql-query-example)
- Hibernate 5 - Batch processing example (/hibernate-5-batch-processing-example)
- Jackson API - Collection serialization and deserialization example (/jackson-api-collection-serialization-and-deserialization-example)
- Jackson API - Converting POJOs to JSON example (/jackson-api-converting-pojos-to-json-example)
- Jackson API - Streaming parser and generator example (/jackson-api-streaming-parser-and-generator-example)



Reference links

- Java 9 API Specification (https://docs.oracle.com/javase/9/docs/api/overview-summary.html)
- Java 8 API Specification (https://docs.oracle.com/javase/8/docs/api/)
- Java 7 API Specification (https://docs.oracle.com/javase/7/docs/api/)
- Java Tutorial (https://docs.oracle.com/javase/tutorial/)
- Spring Framework API (https://docs.spring.io/spring/docs/current/javadoc-api/)
- Spring Framework Reference Documentation (https://docs.spring.io/spring/docs/current/spring-framework-reference/)
- Spring Boot API (https://docs.spring.io/spring-boot/docs/current/api/)
- Spring Boot Reference Documentation (https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/)
- Spring Security API (https://docs.spring.io/spring-security/site/docs/current/api/)
- Spring Security Reference Documentation (https://docs.spring.io/spring-security/site/docs/current/reference/htmlsingle/)
- Hibernate JavaDoc (http://docs.jboss.org/hibernate/orm/current/javadocs/)
- Hibernate User Guide (http://docs.jboss.org/hibernate/orm/current/userguide/html_single/Hibernate_User_Guide.html)

About me

Sunil Singh Bora, founder of BORAJI.COM (https://www.boraji.com), loves Java programming and open source technologies. The vision of this website is to teach java programming and java web technologies with short and simple examples. If you liked tutorials and examples on this website, please follow me on

(http://www.facebook.com/imssbora)

(http://twitter.com/intent/follow?source=followbutton&variant=1.0&screen_name=imssbora)