

Spring Framework & Spring Boot

SC363204

Java Web Application Development

การพัฒนาโปรแกรมประยุกต์บนเว็บด้วยภาษาจาวา



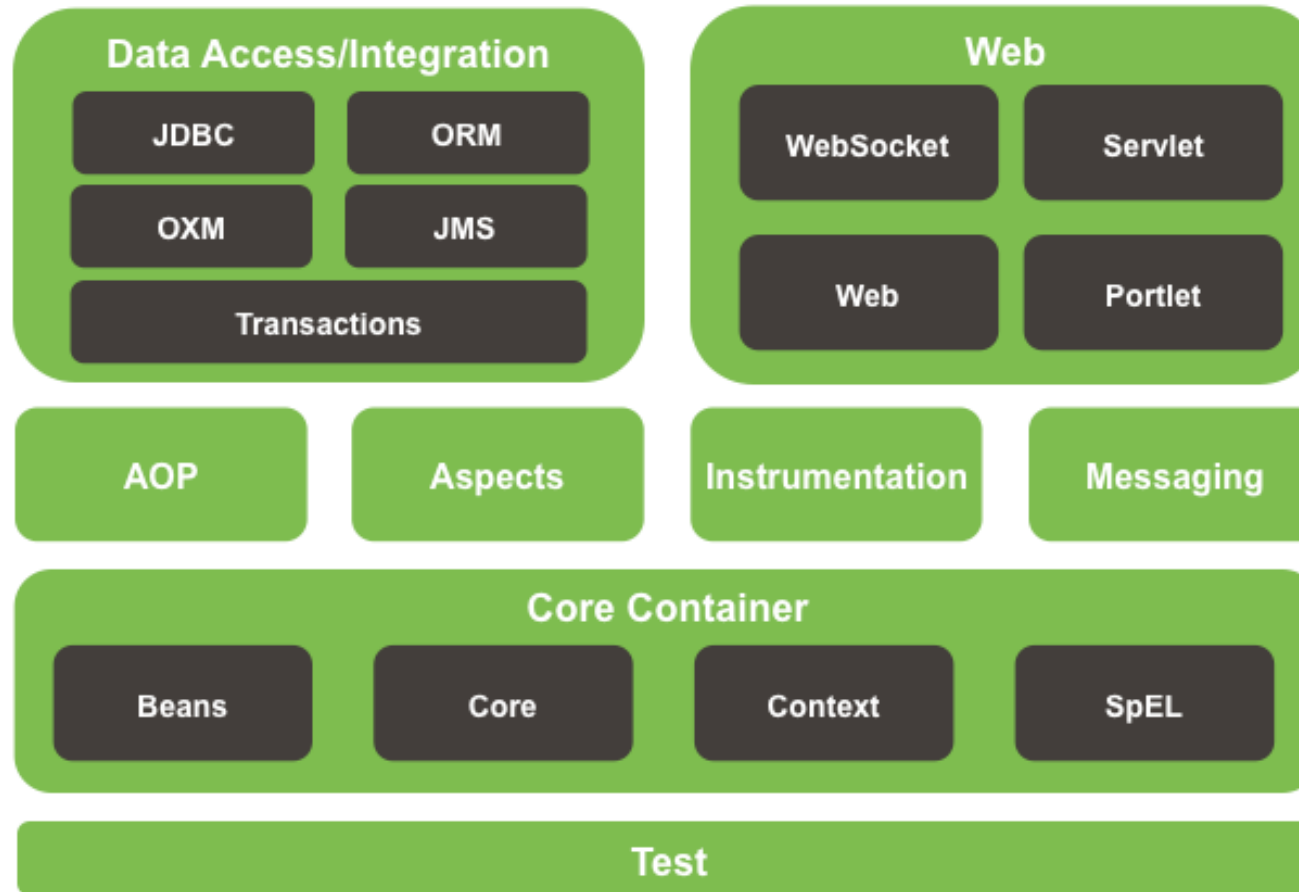
Spring Framework

- Framework สำหรับพัฒนาแอปพลิเคชันสำหรับองค์กรบนแพลตฟอร์มของจาวา
- เป้าหมายของ Spring คือ **ลดเวลาในการพัฒนา** และเสนอแนวปฏิบัติที่ดีให้กับนักพัฒนา
- เป็น **Open source** ที่มีขนาดเล็ก (Lightweight) และสามารถเพิ่มขยายได้ง่าย

Spring Framework



Spring Framework Runtime



Dependency Injection

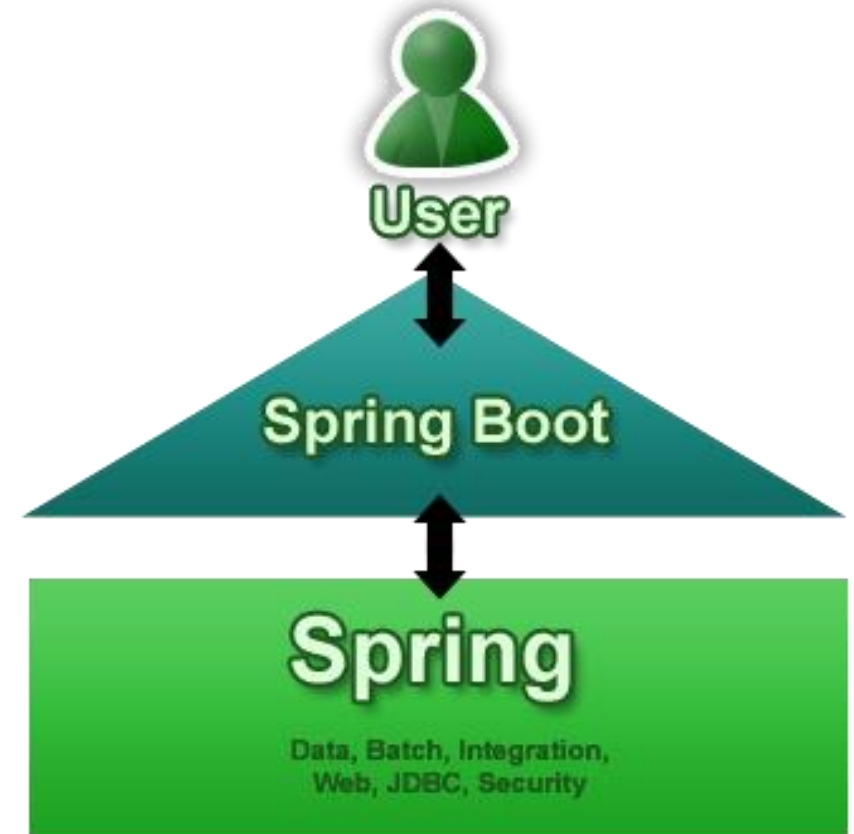
- Dependency Injection เรียกอีกชื่อหนึ่งว่า **IOC** (Inversion of Control Principle)
- เป็นวิธีการออกแบบซอฟต์แวร์ตาม pattern ที่มีการ**แยก component ต่าง ๆ** ให้มีความเกี่ยวข้องกัน**น้อยที่สุด** (loosely coupled)
- Component แต่ละส่วนสามารถนำมาประกอบกัน และสามารถทดสอบได้อย่างง่ายดาย
- ช่วยให้ reuse code ไปใช้ใน application อื่น ๆ ได้ง่าย
- **ไม่มีการ Hard code** หรือกำหนดค่าภายในคลาสต่างๆ โดยตรง แต่จะแยกไฟล์สำหรับ Config ในรูปแบบ XML configuration files หรือ จาวาคลาสสำหรับ Config ค่าโดยเฉพาะ

Dependency Injection

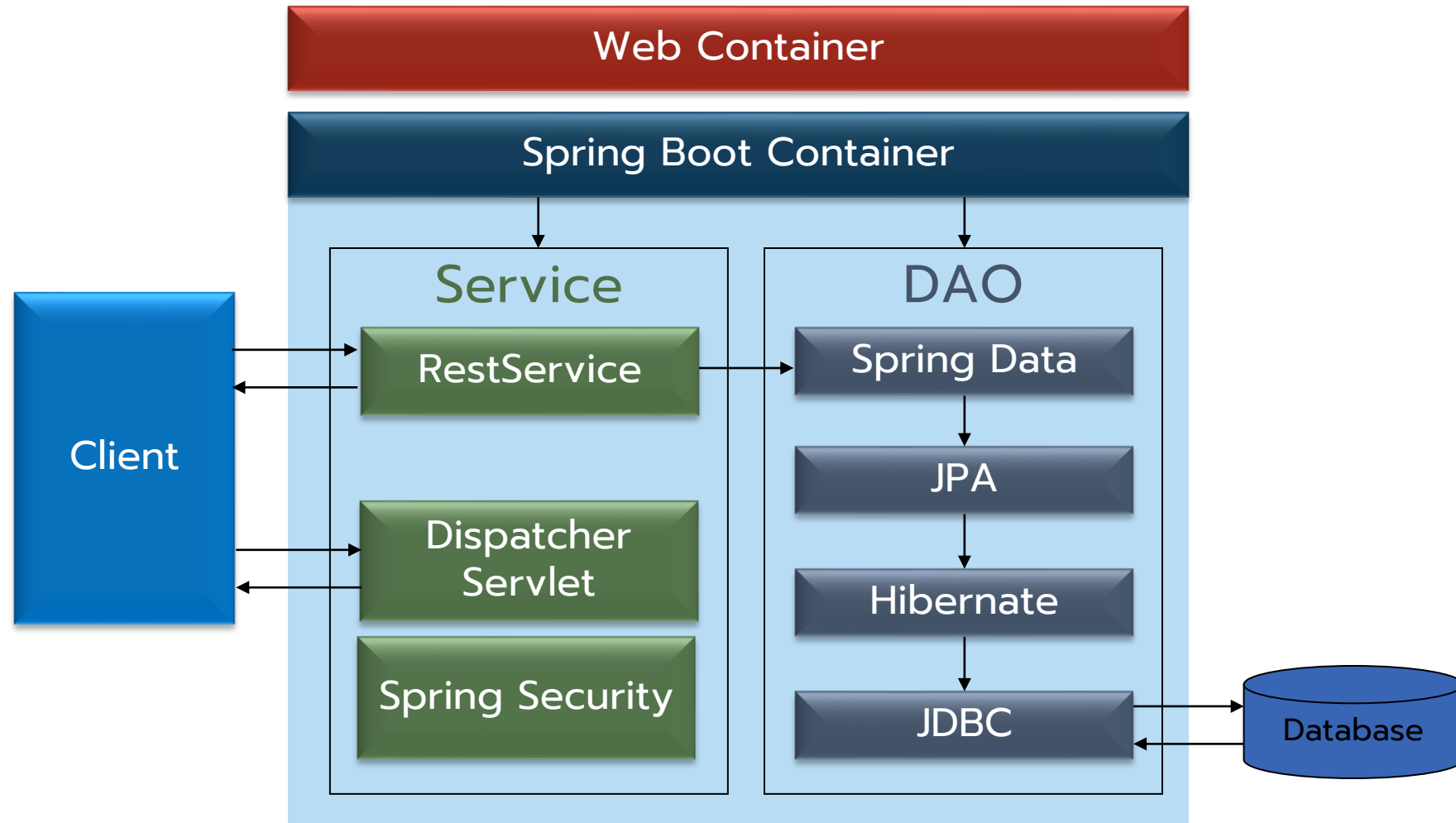
- **Spring Container** คือ ผู้ที่ทำหน้าที่ในการฉีด (Injecting) หรือสร้าง object เข้าไปยัง object ที่มีความเกี่ยวข้องกัน (dependencies of objects) เพื่อให้เกิดการทำงานร่วมกันในแอปพลิเคชัน
- Dependency Injection แบ่งออกเป็น 2 ประเภท ได้แก่ **Setter Injection** และ **Constructor Injection**

Spring Boot

- Spring Boot **ไม่ใช่ Framework** แต่เป็นเครื่องมือที่ช่วยสร้าง package และรันแอปพลิเคชันที่สร้างด้วย **Spring Framework**
- มีชุด Starter Library ให้**เรียกใช้งานได้สะดวก** ลดขั้นตอนการเพิ่ม library อื่นๆ ที่เกี่ยวข้อง
- มี **Auto Configuration** เพื่อให้ลดขั้นตอนการกำหนดค่าต่างๆ
- มี **Built-in web server** เช่น Tomcat ช่วยให้สร้างแอปพลิเคชันแบบแยกส่วนการทำงานได้ง่าย

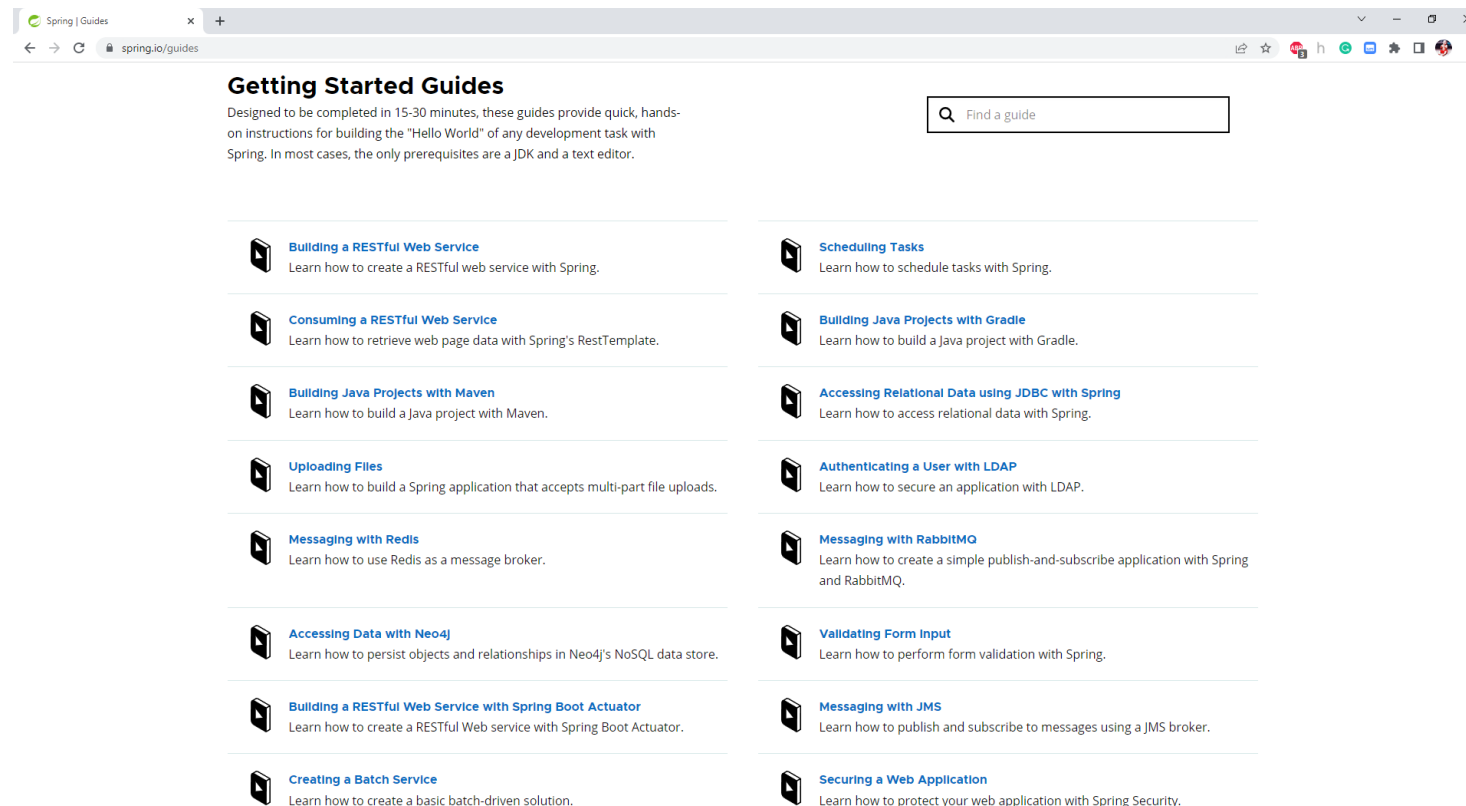


Spring Boot Architecture



Spring Boot

- Spring Boot ทำอะไรได้บ้าง
- ตัวอย่างโค้ดและคำอธิบาย Spring Boot ในเคสต่างๆ -> <https://spring.io/guides>



Spring Boot Application Starters

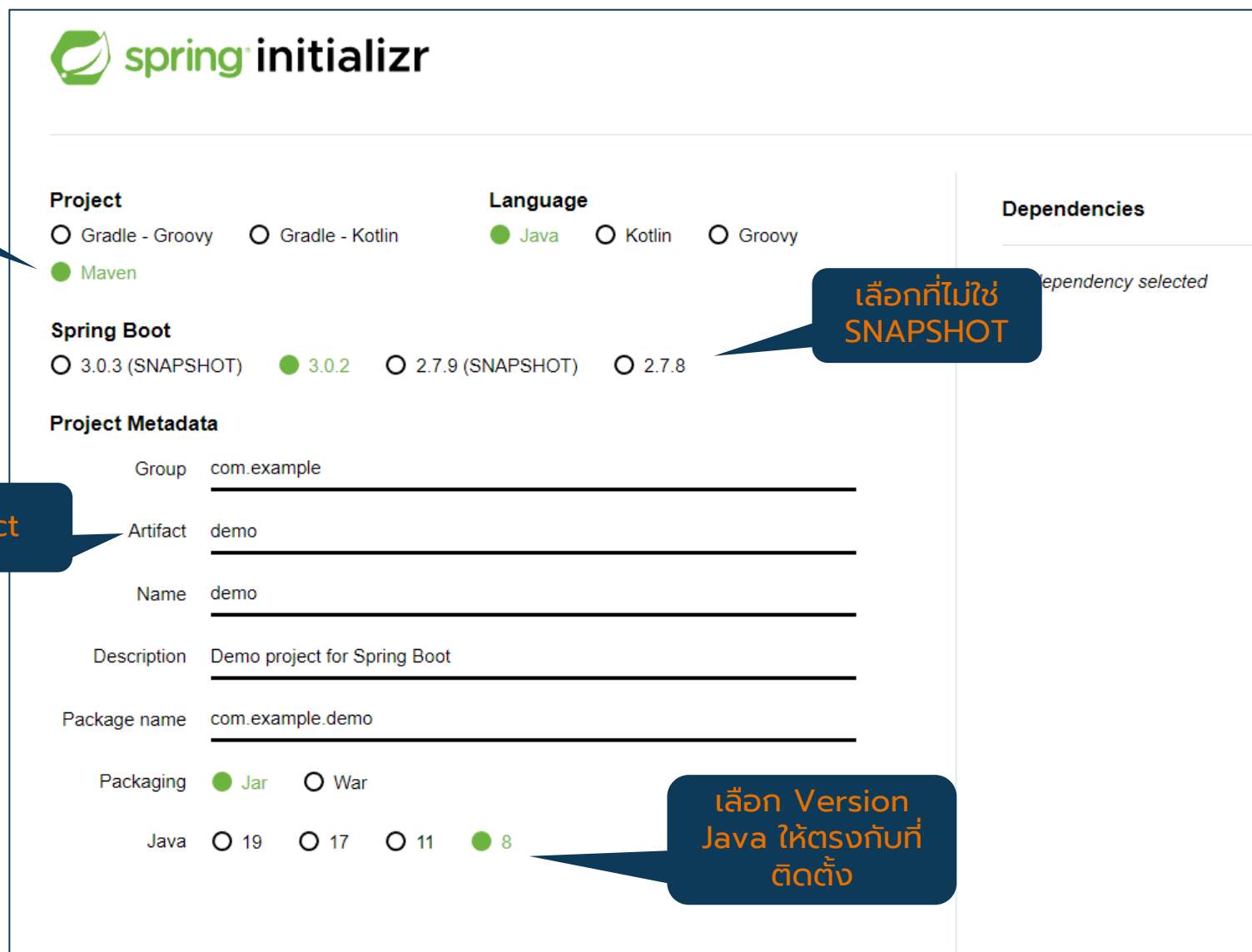
- Spring Boot มีชุดแม่แบบ (Starter พร้อมใช้) แบ่งตามประเภทของแอปพลิเคชัน มีชื่อ artifactId ขึ้นต้นด้วย spring-boot-starter-* สามารถนำชื่อ Dependency ไประบุในไฟล์ POM ของโปรเจกต์ Maven ซึ่งจะได้ชุด Library .jar ต่างๆ ที่เกี่ยวข้องมาเก็บไว้ในโปรเจกต์แบบอัตโนมัติ

ชื่อ Dependency	คำอธิบาย
spring-boot-starter	ใช้กับแอปพลิเคชันที่ใช้เฉพาะส่วนหลักของ Spring Boot ในส่วน auto-configuration ต่างๆ
spring-boot-starter-web	ใช้ในการพัฒนาเว็บแบบ full-stack ประกอบด้วย Tomcat และ spring-webmvc
spring-boot-starter-jdbc	ใช้กับการติดต่อฐานข้อมูลผ่าน JDBC
spring-boot-starter-data-jpa	สนับสนุน JPA (Java Persistence API) ประกอบด้วย spring-data-jpa, spring-orm และ Hibernate
spring-boot-starter-data-mongodb	สนับสนุน MongoDB NoSQL Database
spring-boot-starter-data-rest	ใช้พัฒนา Web API (RESTful API) จาก spring-data-rest-webmvc
spring-boot-starter-freemarker	ใช้ Render หน้าแสดงผลโดยไม่ใช้โค้ดภาษาจาวาด้วย FreeMarker templating engine
spring-boot-starter-thymeleaf	ใช้ Render หน้าแสดงผลโดยไม่ใช้โค้ดภาษาจาวาด้วย Thymeleaf templating engine
spring-boot-starter-ws	ใช้พัฒนา Web Services
spring-boot-starter-cloud-connectors	ใช้ติดต่อกับ "Spring Cloud Connectors" เช่น Cloud Foundry และ Heroku
spring-boot-starter-security	ใช้จัดการความปลอดภัยในการเข้าถึงระบบ
spring-boot-starter-social-facebook	ใช้ติดต่อกับ Facebook API
spring-boot-starter-social-twitter	ใช้ติดต่อกับ Twitter API

Spring Boot

- การสร้าง Spring Boot Project
 - ใช้ตัวสร้างต้นแบบจากเว็บ **Spring Initializr**
 - <https://start.spring.io/>
 - จะได้ไฟล์เริ่มต้นสำหรับนำไปสร้าง Project บนเครื่องมือที่ใช้พัฒนา เช่น Eclipse ในรูปแบบ
 - [ชื่อโปรเจค].zip

Spring Initializr



The image shows the Spring Initializr web form with several Thai annotations in dark blue boxes with orange text. The form is divided into sections: Project, Language, Spring Boot, Project Metadata, and Dependencies. The 'Project' section has radio buttons for Gradle - Groovy, Gradle - Kotlin, Maven (selected), and Groovy. The 'Language' section has radio buttons for Java (selected), Kotlin, and Groovy. The 'Spring Boot' section has radio buttons for 3.0.3 (SNAPSHOT), 3.0.2 (selected), 2.7.9 (SNAPSHOT), and 2.7.8. The 'Project Metadata' section has text input fields for Group (com.example), Artifact (demo), Name (demo), Description (Demo project for Spring Boot), and Package name (com.example.demo). The 'Packaging' section has radio buttons for Jar (selected) and War. The 'Java' section has radio buttons for 19, 17, 11, and 8 (selected). The 'Dependencies' section has a text input field for 'dependency selected'. The annotations point to the Maven, 3.0.2, 8, and Jar options.

spring initializr

Project

☐ Gradle - Groovy ☐ Gradle - Kotlin ☒ **Maven**

Language

☒ **Java** ☐ Kotlin ☐ Groovy

Spring Boot

☐ 3.0.3 (SNAPSHOT) ☒ **3.0.2** ☐ 2.7.9 (SNAPSHOT) ☐ 2.7.8

Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ **Jar** ☐ War

Java ☐ 19 ☐ 17 ☐ 11 ☒ **8**

Dependencies

เลือก Maven

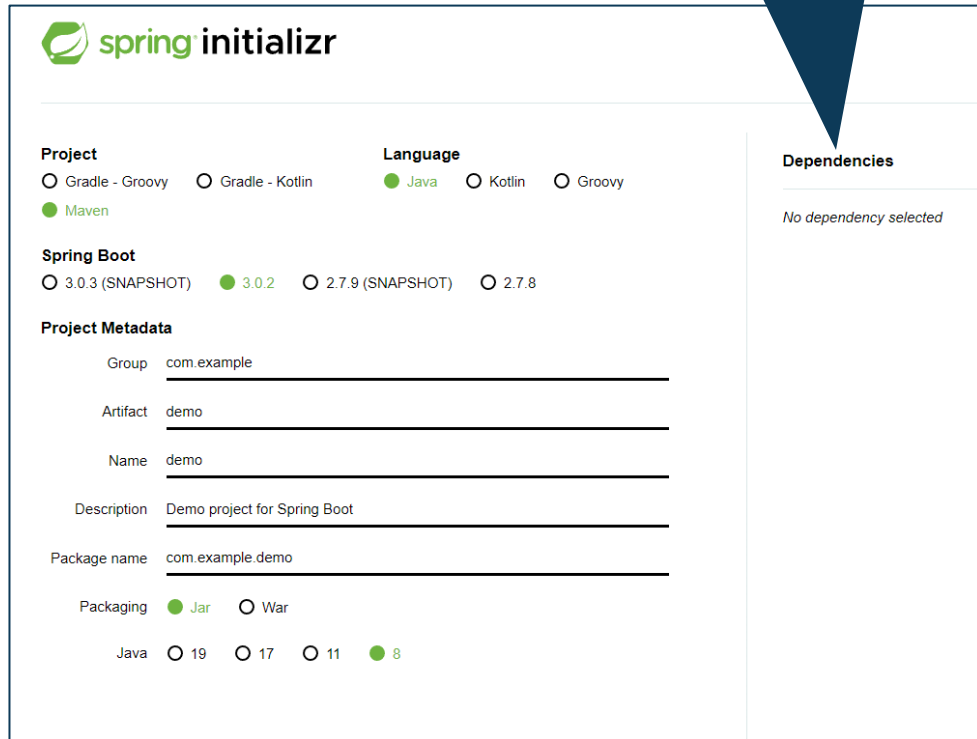
ใส่ชื่อ Project

เลือกที่ไม่ใช่
SNAPSHOT

เลือก Version
Java ให้ตรงกับที่
ติดตั้ง

Spring Initializr

no add dependencies



The Spring Initializr form is divided into several sections. The 'Project' section has radio buttons for 'Gradle - Groovy', 'Gradle - Kotlin', 'Maven' (selected), 'Kotlin', and 'Groovy'. The 'Language' section has radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'. The 'Spring Boot' section has radio buttons for '3.0.3 (SNAPSHOT)', '3.0.2' (selected), '2.7.9 (SNAPSHOT)', and '2.7.8'. The 'Project Metadata' section includes text input fields for 'Group' (com.example), 'Artifact' (demo), 'Name' (demo), 'Description' (Demo project for Spring Boot), and 'Package name' (com.example.demo). The 'Packaging' section has radio buttons for 'Jar' (selected) and 'War'. The 'Java' section has radio buttons for '19', '17', '11', and '8' (selected). The 'Dependencies' section on the right shows 'No dependency selected'.

เลือกชุดแม่แบบ

- Spring Boot DevTools
- Spring Web
- Spring Data JPA
- MySQL Driver

Web, Security, JPA, Actuator, Devtools...

Press Ctrl for multiple adds

DEVELOPER TOOLS

Java annotation library which helps to reduce boilerplate code.

Spring Configuration Processor

Generate metadata for developers to offer contextual help and "code completion" when working with custom configuration keys (ex.application.properties/.yml files).

WEB

Spring Web

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Reactive Web

Build reactive web applications with Spring WebFlux and Netty.

Spring Initializr



Project
☐ Gradle - Groovy ☐ Gradle - Kotlin ☒ **Java** ☐ Kotlin ☐ Groovy
☒ **Maven**

Spring Boot
☐ 3.0.3 (SNAPSHOT) ☒ **3.0.2** ☐ 2.7.9 (SNAPSHOT) ☐ 2.7.8

Project Metadata

Group
Artifact
Name
Description
Package name
Packaging ☒ **Jar** ☐ War
Java ☐ 19 ☐ 17 ☐ 11 ☒ **8**

Dependencies

[ADD DEPENDENCIES... CTRL + B](#)

Spring Web **WEB**
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Boot DevTools **DEVELOPER TOOLS**
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

Spring Data JPA **SQL**
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

MySQL Driver **SQL**
MySQL JDBC driver.

download ไฟล์
เริ่มต้น
จะได้ไฟล์ .zip

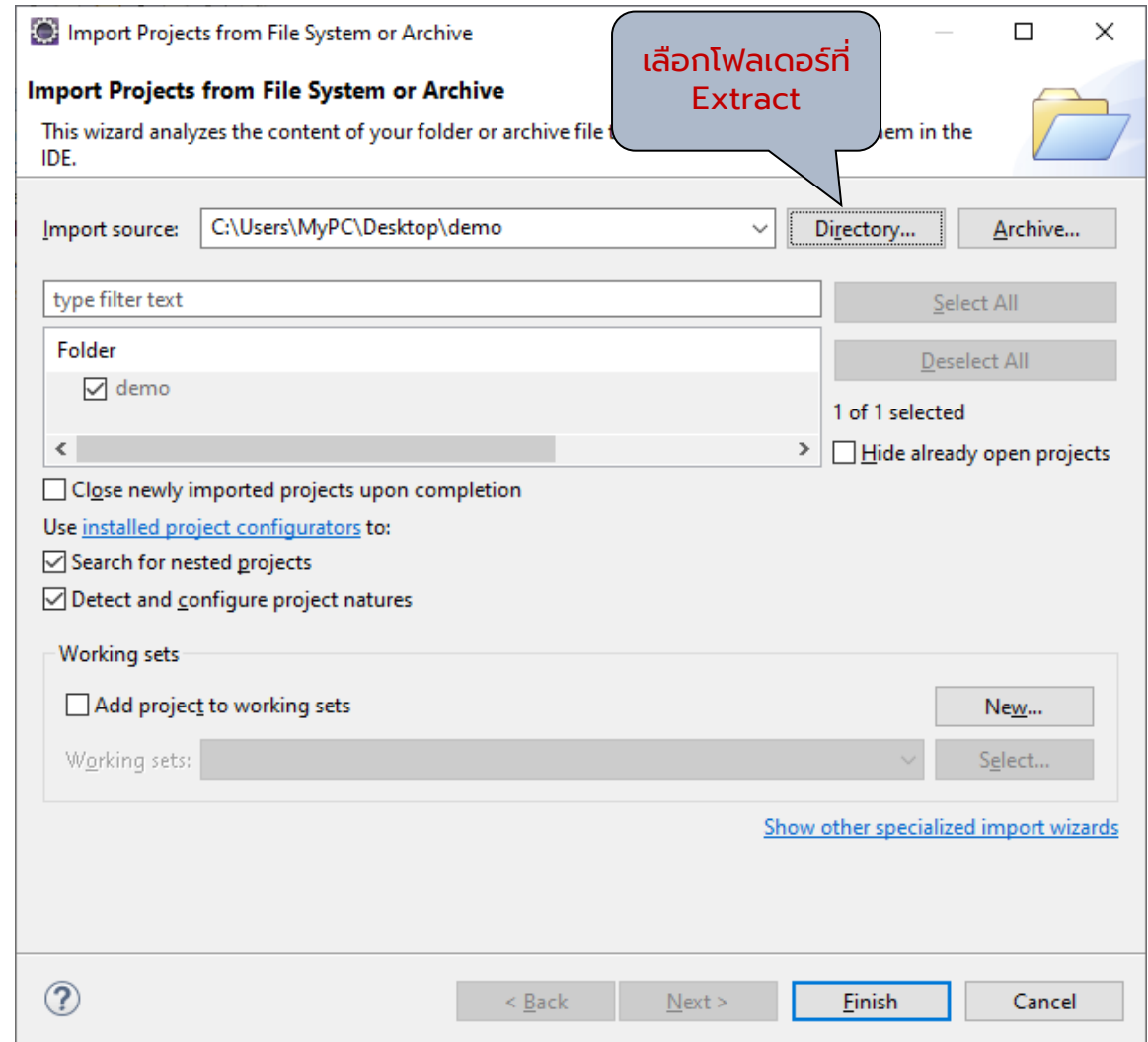
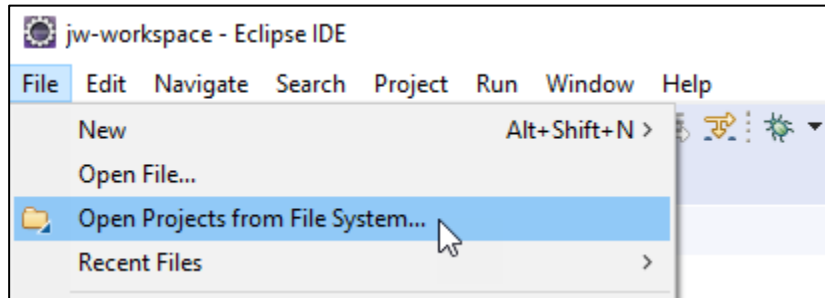
[GENERATE CTRL + G](#)

[EXPLORE CTRL + SPACE](#)

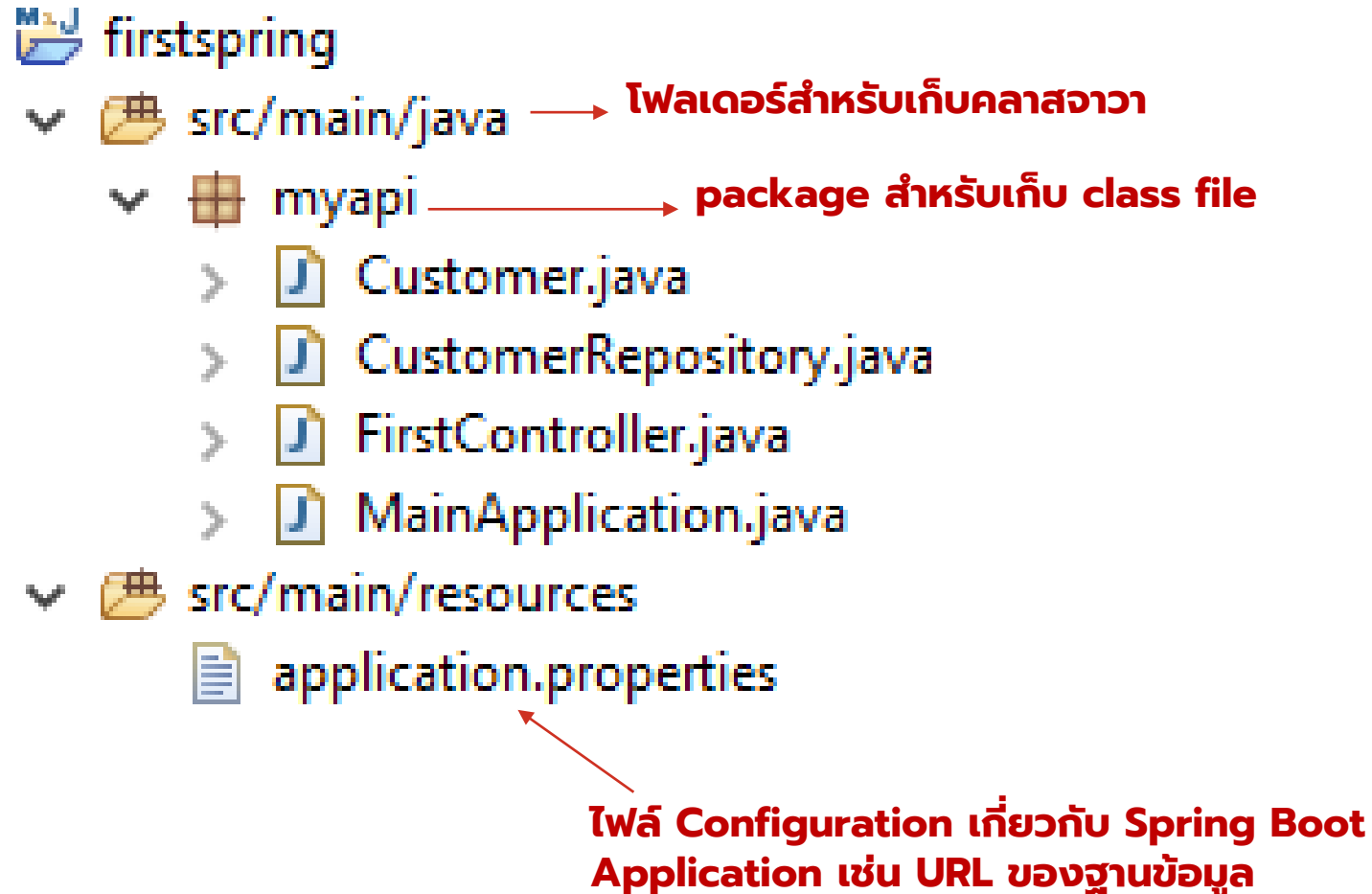
[SHARE...](#)

Eclipse

- การเปิด Project บน Eclipse
 - Extract ไฟล์ .zip ไว้ที่ใดก็ได้
 - เปิดโปรเจกต์ เลือกเมนู Open Projects from File System...

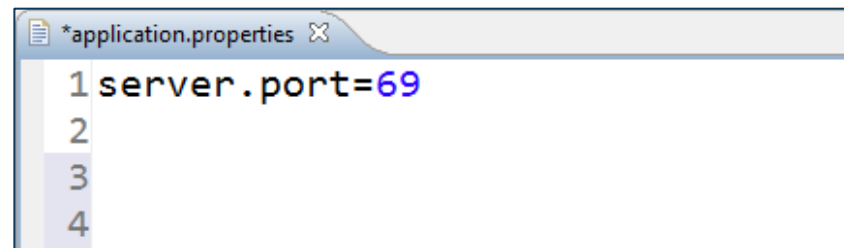
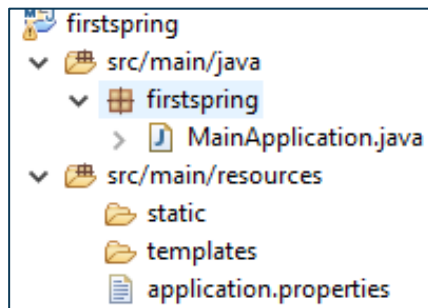


โครงสร้างการเก็บไฟล์

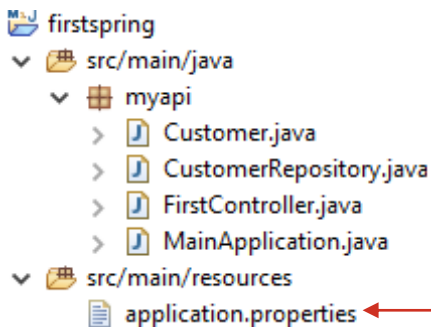


application.properties

- การ Config ค่าเกี่ยวกับ Spring Boot Application จะถูกกำหนดในไฟล์ application.properties
- การเก็บข้อมูลจะอยู่ในรูปแบบ
- `property-name = value`
- แม่แบบ spring-boot-starter-web มี Web Container Tomcat ฝังมาในตัวแล้ว และกำหนด port เป็น 8080 ดังนั้นควรถูก Stop Server ที่เคยเปิดอยู่ เพื่อป้องกัน port ชนกัน หรือเปลี่ยน port ของ Spring Boot Application โดยเพิ่ม property ดังนี้



Config Database



กำหนดค่าสำหรับที่ติดต่อฐานข้อมูลในไฟล์
application.properties

หมายเหตุ ภาษา SQL แบ่งคำสั่งออกเป็น 2 ส่วน

1. ส่วนนิยามข้อมูล (Data Definition Language: DDL) เช่น คำสั่ง create/alter/drop table
2. ส่วนจัดการข้อมูล (Data Manipulation Language: DML) เช่น คำสั่ง select, insert, update, delete

```
spring.mvc.view.prefix=/jsp/  
spring.mvc.view.suffix=.jsp
```

```
spring.datasource.url = jdbc:mysql://localhost/mydb?characterEncoding=utf-8  
spring.datasource.username = root  
spring.datasource.password =
```

```
spring.jpa.show-sql = true  
spring.jpa.properties.hibernate.format_sql=true
```

```
spring.jpa.hibernate.ddl-auto = update
```

```
# spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQLDialect
```

ชื่อฐานข้อมูล

กำหนดให้แสดงคำสั่ง SQL บน console
true - แสดง, **false** - ไม่แสดง

- เป็นการกำหนดให้สร้างคำสั่งสำหรับสร้างโครงสร้างตารางในฐานข้อมูล (DDL) แบบอัตโนมัติ เมื่อเจอ Entity class ขณะเริ่ม start spring boot
- ค่าที่กำหนดได้ประกอบด้วย **none, create-drop, create , update**
- **create-drop** – สร้าง schema ตาม Entity class ที่มีและทำลายเมื่อจบการทำงาน
- **create** – สร้าง schema ตาม Entity class ที่มีแต่ไม่ทำลายเมื่อจบการทำงาน
- **update** – หากยังไม่เคยสร้าง schema จะสร้างก่อน ถ้า start server ใหม่จะตรวจสอบว่า entity มีการปรับปรุงหรือไม่ ถ้ามีจะเพิ่มให้อัตโนมัติ

ตัวอย่าง property อื่นๆ

```
# EMBEDDED SERVER CONFIGURATION (ServerProperties)
server.port=8080
server.address= # bind to a specific NIC
server.session-timeout= # session timeout in seconds

# HTTP encoding (HttpEncodingProperties)

# encoding of HTTP requests/responses
spring.http.encoding.charset=UTF-8

# enable http encoding support
spring.http.encoding.enabled=true

# force the configured encoding
spring.http.encoding.force=true
```

ดู property อื่น ๆ ได้จาก <https://docs.spring.io/spring-boot/docs/current/reference/html/appendix-application-properties.html#json-properties>

ทดสอบสร้าง Controller

`@RestController` •——— ระบุว่าให้คลาสนี้เป็น Controller ของ RESTful API

```
public class FirstController {
```

`@GetMapping("/hello")` •——— กำหนด Path ของ HTTP request ที่ใช้เข้าถึงเมธอดนี้แบบ GET

```
    public String home() {
```

`return "Hello From Spring";` •——— ส่ง HTTP response กลับไปด้วยข้อความธรรมดา (plain text)

```
    }
```

```
}
```

- Spring Initializr จะสร้างคลาสหลักสำหรับการรับ Spring Boot Application มาให้แล้ว

```
@SpringBootApplication • annotation เพื่อให้มีการ config แอปพลิเคชันแบบอัตโนมัติ
public class MainApplication {

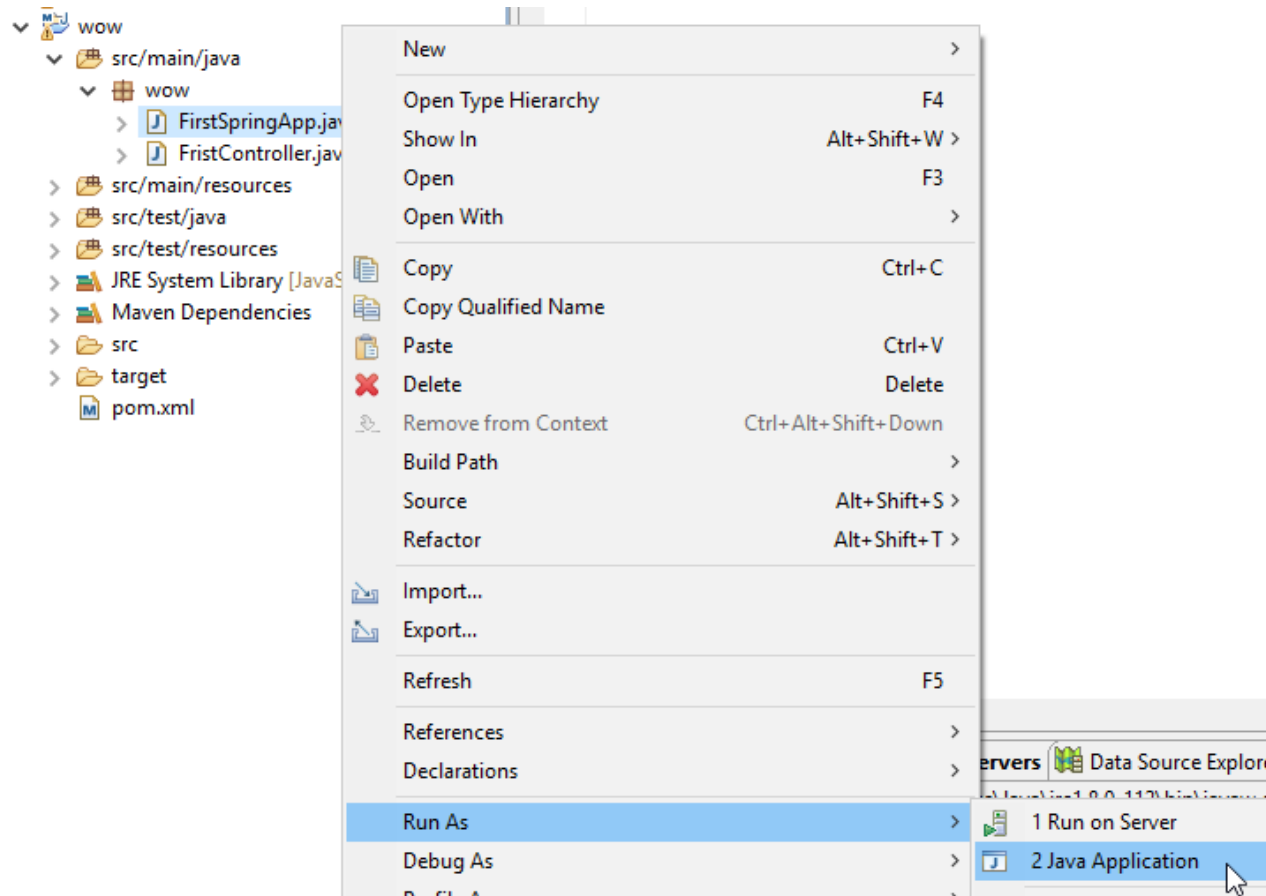
    public static void main(String[] args) { • คลาสเริ่มต้นจะมีเมธอด main( )

        SpringApplication.run(MainApplication.class, args);
        • ส่งให้ Spring Boot Application เริ่มทำงาน      • ใช้นำเข้าทาง command line
    }

}
```

รัน Spring Boot Application

- คลิกขวาที่คลาสหลัก เลือก Run As > Java Application



รับ Spring Boot Application

```

  ____ _
 / ___ \| | | |
/ /   \| |_| |
\ \   /| | | |
 \___/\_|_|_|_|
:: Spring Boot :: (v3.0.2)

```

Version ของ Spring Boot ที่ใช้

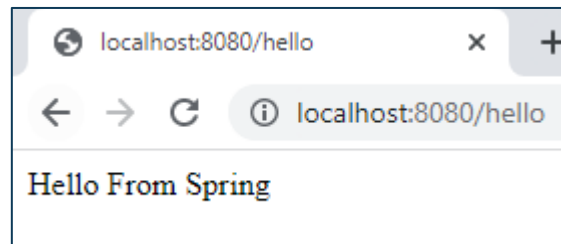
```

2023-02-06T22:29:10.084+07:00 INFO 20324 --- [ restartedMain] com.example.demo.DemoApplication : Starting DemoApplication using Java 17.0.4.1 with PID 20324 (D:\Java Web\demo.zip_expanded\demo\target\classes started by Saitoarm in l
2023-02-06T22:29:10.087+07:00 INFO 20324 --- [ restartedMain] com.example.demo.DemoApplication : No active profile set, falling back to 1 default profile: "default"
2023-02-06T22:29:10.142+07:00 INFO 20324 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable
2023-02-06T22:29:10.142+07:00 INFO 20324 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.web' property to 'DEBUG'
2023-02-06T22:29:10.907+07:00 INFO 20324 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2023-02-06T22:29:10.931+07:00 INFO 20324 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 13 ms. Found 0 JPA repository interfaces.
2023-02-06T22:29:11.654+07:00 INFO 20324 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-02-06T22:29:11.675+07:00 INFO 20324 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-02-06T22:29:11.675+07:00 INFO 20324 --- [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.5]
2023-02-06T22:29:11.763+07:00 INFO 20324 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-02-06T22:29:11.766+07:00 INFO 20324 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1620 ms
2023-02-06T22:29:12.001+07:00 INFO 20324 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2023-02-06T22:29:12.073+07:00 INFO 20324 --- [ restartedMain] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.1.6.Final
2023-02-06T22:29:12.234+07:00 WARN 20324 --- [ restartedMain] org.hibernate.orm.deprecation : HHH90000021: Encountered deprecated setting [javax.persistence.sharedCache.mode], use [jakarta.persistence.sharedCache.mode] instead
2023-02-06T22:29:12.352+07:00 INFO 20324 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2023-02-06T22:29:13.475+07:00 ERROR 20324 --- [ restartedMain] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Exception during pool initialization.

```

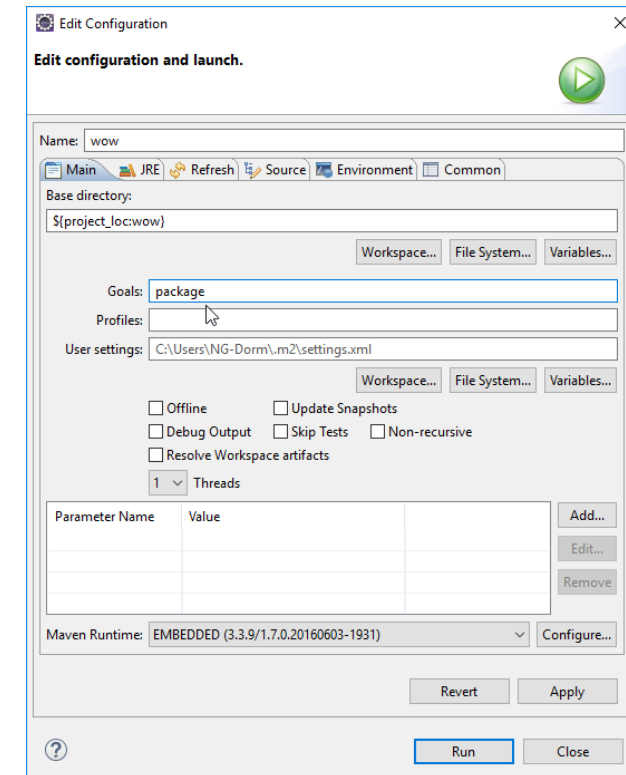
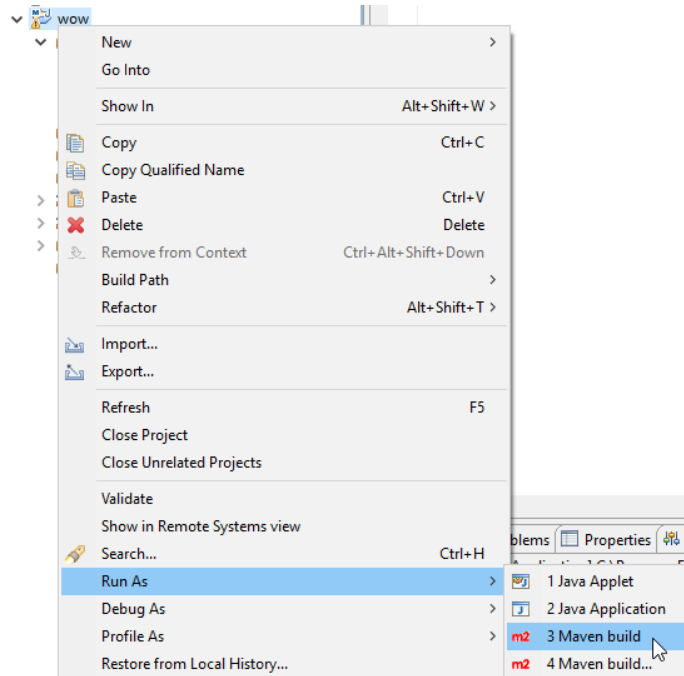
ใช้ port 8080

ผลลัพธ์จากการส่ง HTTP Request ผ่าน Browser



การสร้าง Executable jar file (fat jars)

- Spring Boot Application ที่สร้างเสร็จแล้ว สามารถนำไปรันที่เครื่องใดก็ได้ที่มี JRE แต่ต้องสร้าง Executable jar file ขึ้นมาก่อน
- คลิกขวาที่ชื่อ Project เลือก Run As > Maven build...
- ที่ช่อง Goals ใส่คำว่า "package" และกดปุ่ม Run



กำหนดตำแหน่ง Java SE

The image shows two overlapping Eclipse IDE configuration windows. The 'Edit Configuration' window on the left has tabs for 'Main', 'JRE', 'Refresh', 'Source', and 'Environment'. The 'JRE' tab is active, showing 'Runtime JRE' options. A callout '1. เลือกเก็บ JRE' points to the 'JRE' tab. Another callout '2. คลิก' points to the 'Execution environment' dropdown menu. Below, the 'VM arguments' field is empty, and a callout '4. เริ่ม package' points to the 'Run' button. The 'Execution Environments' window on the right shows a list of 'Execution Environments' and 'Compatible JREs'. A callout '3. เลือก JavaSE Version ที่ตรงกับ Project' points to 'JavaSE-1.8' in the list. The 'Compatible JREs' list shows 'jre1.8.0_241 [perfect match]' selected.

Edit Configuration

Name: firstspring

Runtime JRE:

- ☐ Workspace default JRE (jre1.8.0_241)
- ☒ Execution environment: CDC-1.0/Foundation-1.0 (jre1.8.0_241) **2. คลิก**
- ☐ Alternate JRE: jre1.8.0_241

VM arguments:

1. เลือกเก็บ JRE

4. เริ่ม package

Execution Environments

This page displays the JREs compatible for each execution environment. When multiple JREs are compatible with an environment, a default JRE can be specified

Execution Environments:

- CDC-1.0/Foundation-1.0
- OSGi/Minimum-1.0
- CDC-1.1/Foundation-1.1
- JRE-1.1
- OSGi/Minimum-1.1
- J2SE-1.2
- OSGi/Minimum-1.2
- J2SE-1.3
- J2SE-1.4
- J2SE-1.5
- JavaSE-1.6
- JavaSE-1.7
- JavaSE-1.8** **3. เลือก JavaSE Version ที่ตรงกับ Project**
- JavaSE-9
- JavaSE-10
- JavaSE-11
- JavaSE-12
- JavaSE-13

Compatible JREs:

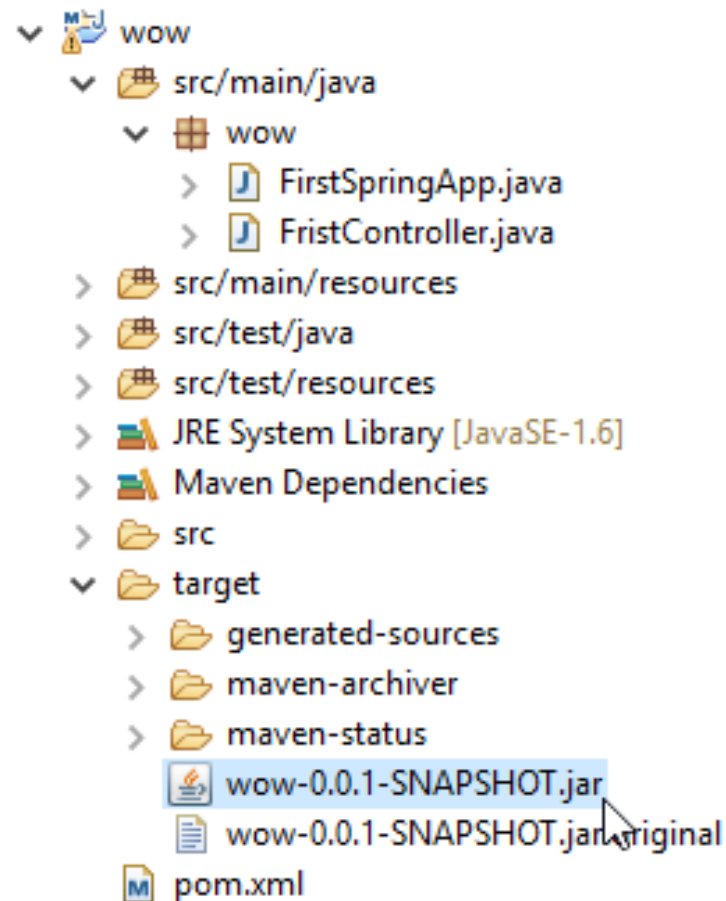
- ☒ jre1.8.0_241 [perfect match]

Buttons: Revert, Apply, Run, Close, Apply and Close, Cancel

jar file ที่ได้

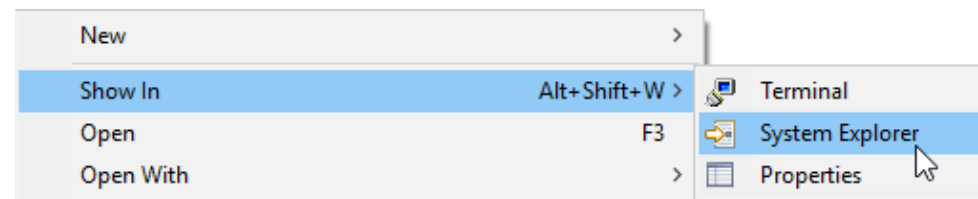
- คลิกขวาที่ชื่อ Project > Refresh จะเห็นไฟล์ Executable jar file (.jar) ภายในโฟลเดอร์

target



การรับ Executable jar file

- คลิกขวาที่โฟลเดอร์ target
 - เลือก Show In > System Explorer
- กดปุ่ม Shift ค้างไว้แล้วคลิกขวาที่โฟลเดอร์ target
 - เลือก Open command window here



- พิมพ์คำสั่งเพื่อเปิดการทำงานไฟล์ .jar ที่สร้างขึ้น รูปแบบดังนี้

java -jar ชื่อไฟล์.jar

```
C:\WINDOWS\system32\cmd.exe - java -jar wow-0.0.1-SNAPSHOT.jar
C:\Users\NG-Dorm\workspace\wow\target>java -jar wow-0.0.1-SNAPSHOT.jar

Spring
:: Spring Boot :: (v1.5.2.RELEASE)

2560-04-08 23:35:38.616 INFO 9556 --- [main] wow.FirstSpringApp
.1-SNAPSHOT on NG-Dorm with PID 9556 (C:\Users\NG-Dorm\workspace\wow\target\wow-0.0.1-SNAPSHOT.jar)
2560-04-08 23:35:38.632 INFO 9556 --- [main] wow.FirstSpringApp
ng back to default profiles: default
2560-04-08 23:35:38.788 INFO 9556 --- [main] ationConfigEmbeddedWebApplicationContext
```

กรณีใช้ JSP เป็นตัว render engine

- แก้ไฟล์ pom.xml

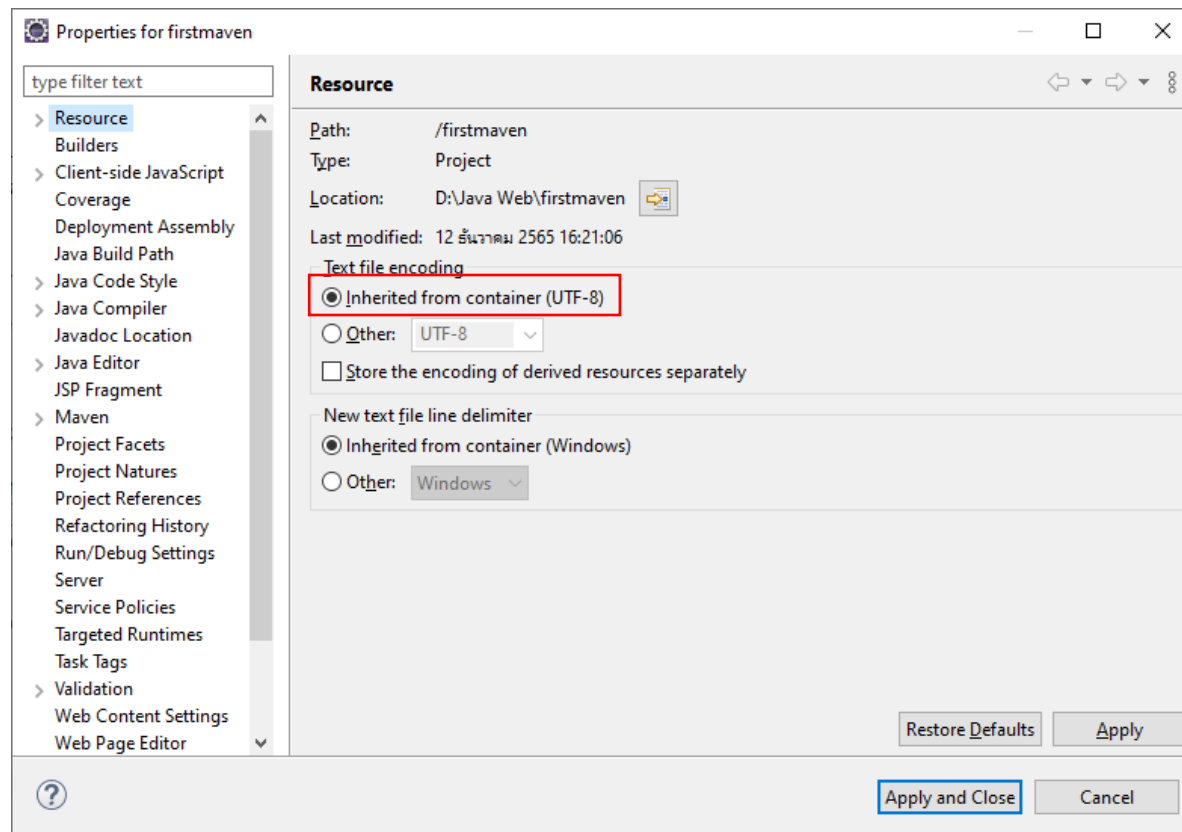
```
<packaging>war</packaging>
```

- แล้ว package ตามปกติ

- เวลารันใส่ชื่อบางสกุลให้ตรง

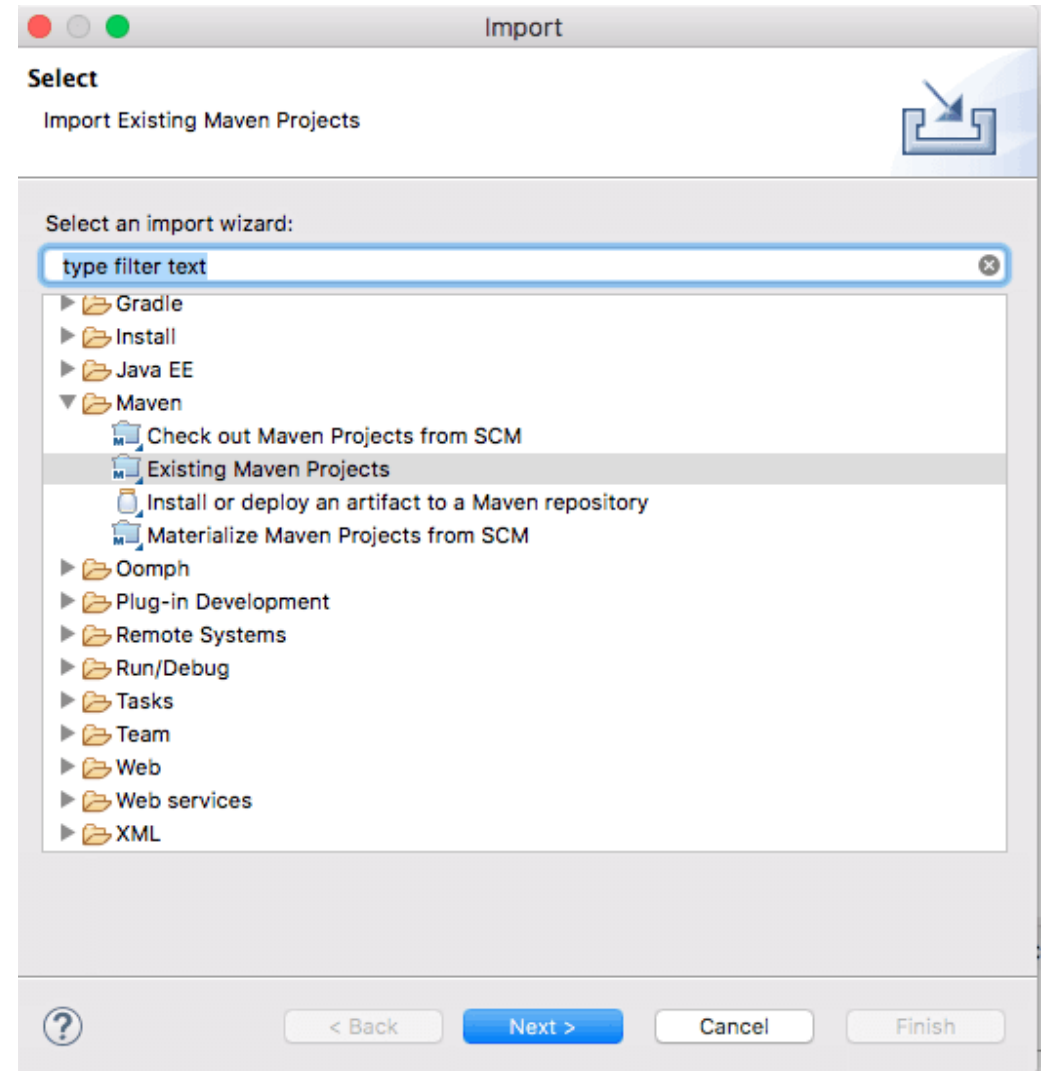
- java -jar ชื่อไฟล์.war

- ให้คลิกขวาที่ชื่อ Project แล้วเลือก Property > Resource > เลือก Inherit from container



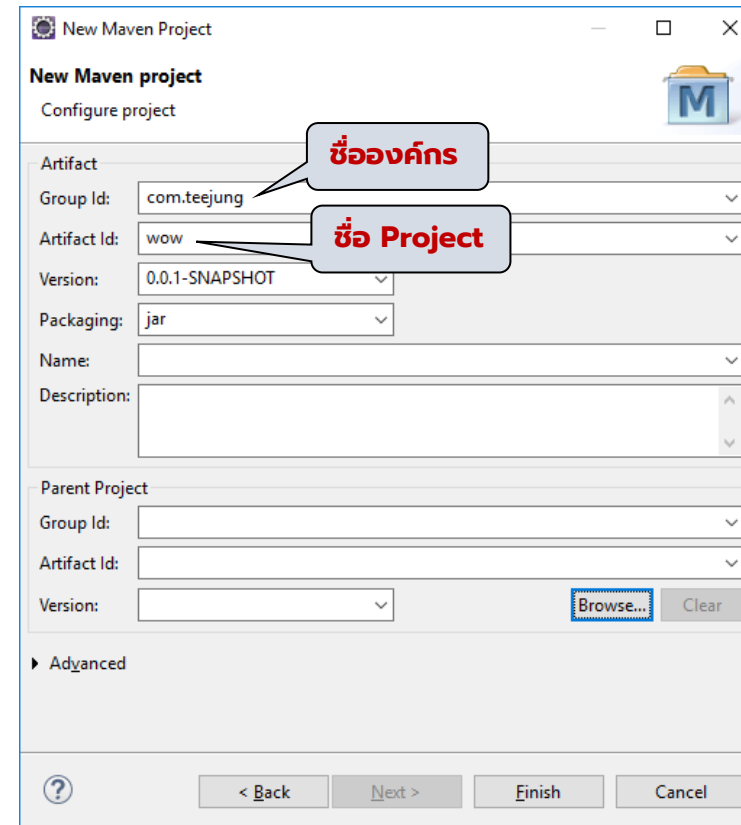
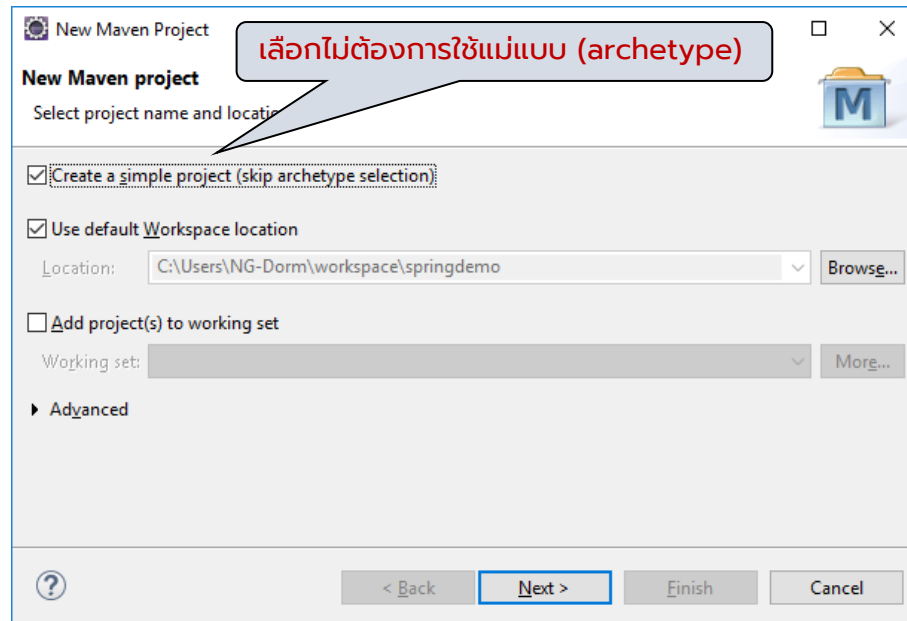
Import ไฟล์จาก Spring Initializr

1. Extract file .zip ไว้ที่ใดก็ได้
2. Import ไฟล์ โดยคลิกขวาที่ชื่อ Project เลือกเมนู
3. Import... > Existing Maven Project



การสร้าง Spring Boot Application

- ที่โปรแกรม Eclipse เลือกเมนู File > New > Maven Project



การกำหนดค่าใน pom.xml

กำหนด dependency
เริ่มต้นของ Spring Boot
Application

สำหรับพัฒนา Web App,
RESTful API

สำหรับ reload server
อัตโนมัติ

JPA: Java Persistence
API

MySQL JDBC Driver

```
<project xmlns="...">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.teejung</groupId>
  <artifactId>wow</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.0.0.RELEASE</version>
  </parent>

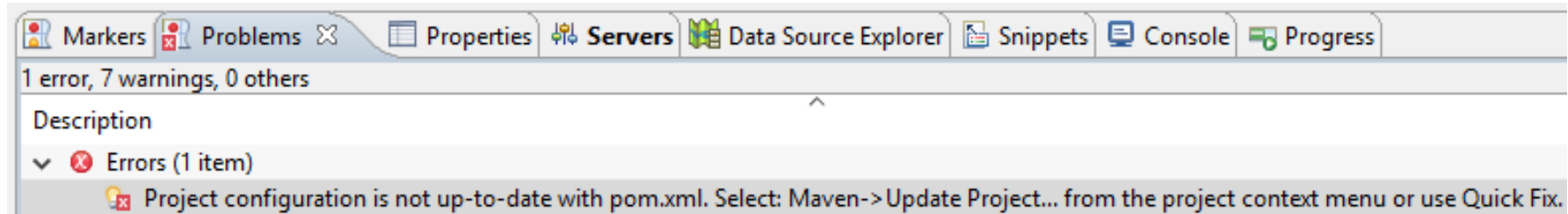
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-devtools</artifactId>
      <optional>true</optional>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>

    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
    </dependency>
  </dependencies>
</project>
```

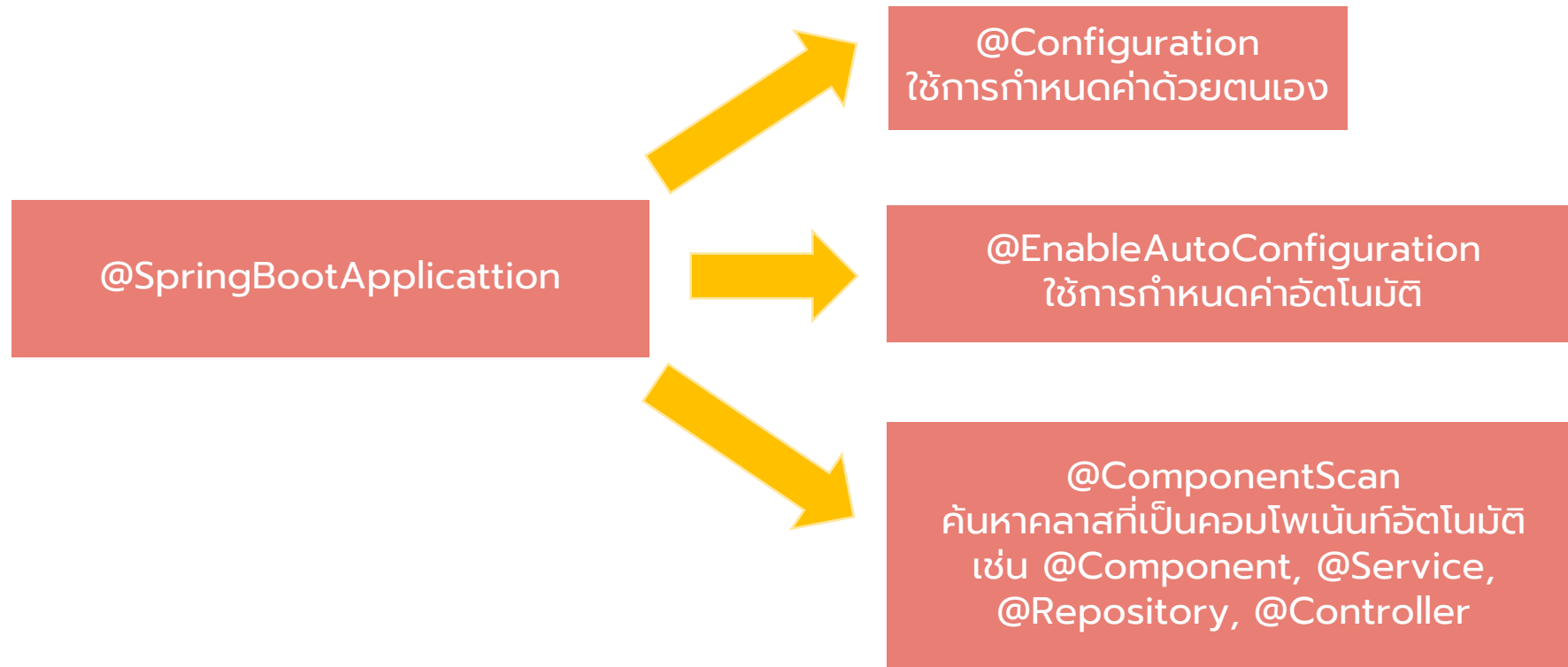

การกำหนดค่าใน pom.xml



- หาก Save ไฟล์ pom.xml แล้ว Project มี Error ดังภาพ ให้คลิกขวาที่ Project เลือกเมนู Maven > Update Project...
- แล้วเลือกที่ "Force Update of Snapshots/Release" และกดปุ่ม OK

- เมื่อมีการระบุ annotation นี้ Spring Boot จะคาดการณ์การ configuration โดยพิจารณาจาก dependencies (.jar) ที่กำหนดไว้ในโปรเจกแบบอัตโนมัติ
- เช่น หากใช้ spring-boot-starter-web จะถูกเพิ่ม Tomcat และ Spring MVC ซึ่งจะถูก setup ค่าให้เป็นเว็บแอปพลิเคชัน และ start Tomcat Web Server แบบอัตโนมัติ
- Auto-configuration จะทำงานได้ดีกับ Dependency ที่เป็น Starter ของ Spring นักพัฒนาอาจไม่ใช้ Starter ก็ได้ แต่ Spring Boot ก็ยังคงทำงานโดยการพยายามกำหนด configuration ที่ดีที่สุดให้กับแอปพลิเคชัน

- สามารถใช้ Annotation @SpringBootApplication ประกาศเหนือคลาส ซึ่งเทียบเท่ากับการกำหนด 3 annotation



การกำหนดค่าใน pom.xml

- ใช้แท็ก <parent> กำหนดค่าเริ่มต้นของ Spring Boot
- แต่ละ Dependency ให้ใส่ groupId เป็น "org.springframework.boot"

กำหนด dependency เริ่มต้น
ของ Spring Boot Application

Dependency สำหรับพัฒนา
เว็บแอปพลิเคชันแบบ MVC

ใช้ระบุให้ Maven สามารถสร้าง executable jar
file
(อาจเรียกว่า fat jars)

```
<project xmlns="...">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.teejung</groupId>
  <artifactId>wow</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.2.RELEASE</version>
  </parent>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>
```

การ Upload File

- กรณีที่ไฟล์ใหม่เข้ามาจะไม่รู้จัก ให้กำหนดค่าใน application.properties ดังนี้

```
project.base-dir= file:///Users/chimdolin/Documents/workspace/projectjw  
spring.resources.static-locations= ${project.base-dir}/src/main/resources/static/
```

Developer tools

- Spring Boot มีเครื่องมือสำหรับนักพัฒนาที่ใช้ในช่วงวงจรการพัฒนา โดยมี dependency ดังนี้

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-devtools</artifactId>  
  <optional>true</optional>  
</dependency>
```

- การกำหนดแท็ก optional เป็น true ช่วยให้ง่ายต่อการสร้าง Production Application โดย devtools จะถูก disable อัตโนมัติ เมื่อรันด้วย Executable jar file
- เมื่อ class ใดมีการแก้ไขโค้ดและ compile ใหม่ จะ restart application อัตโนมัติ
- สามารถกำหนด path ที่ไม่ต้องการ restart ในไฟล์ application.properties ได้ เช่น
spring.devtools.restart.exclude=static/,public/****