



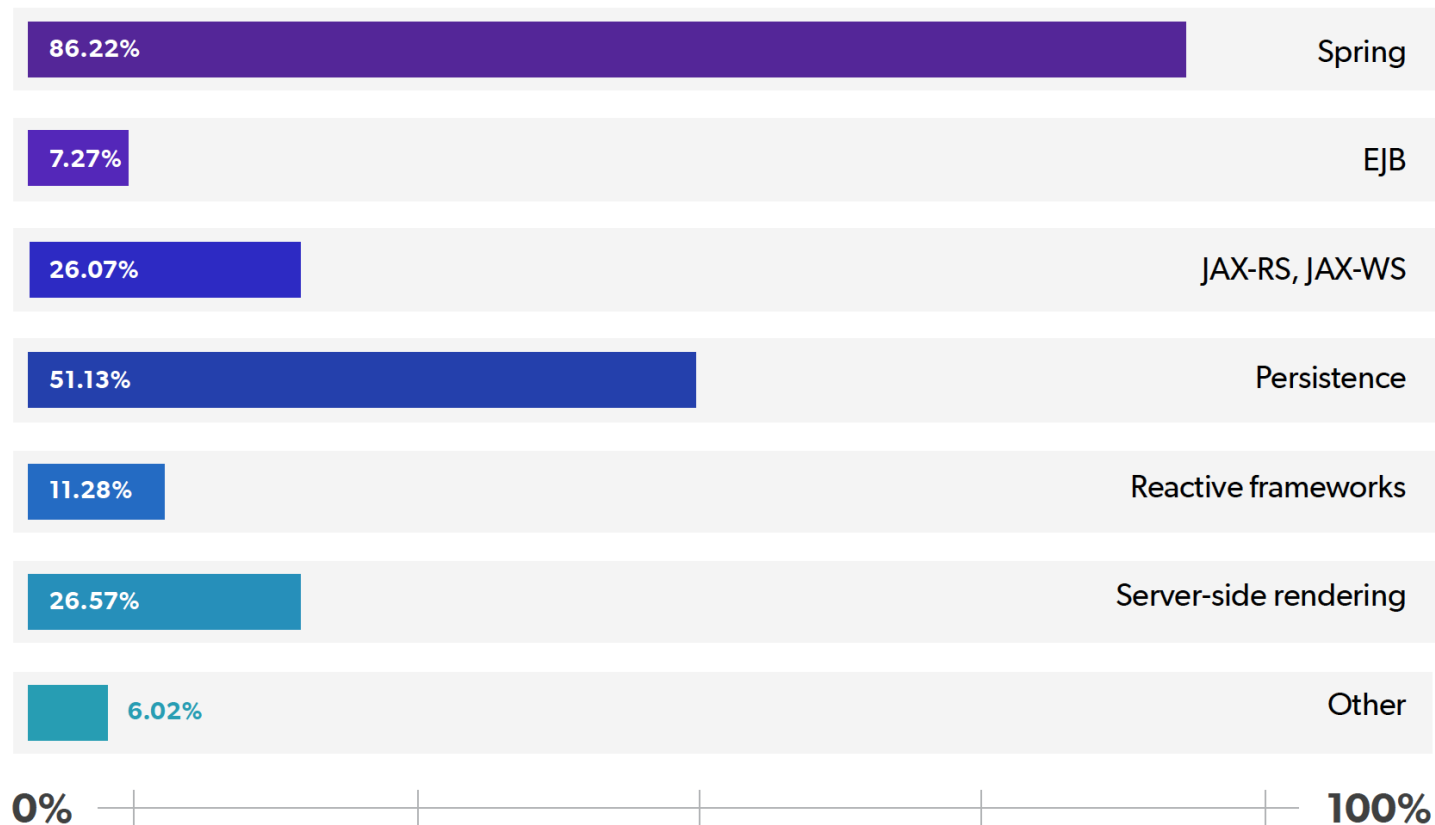
# **Spring Framework & Spring Boot**





# Java Framework Technologies

What Java application frameworks and technologies are you using in your main project?

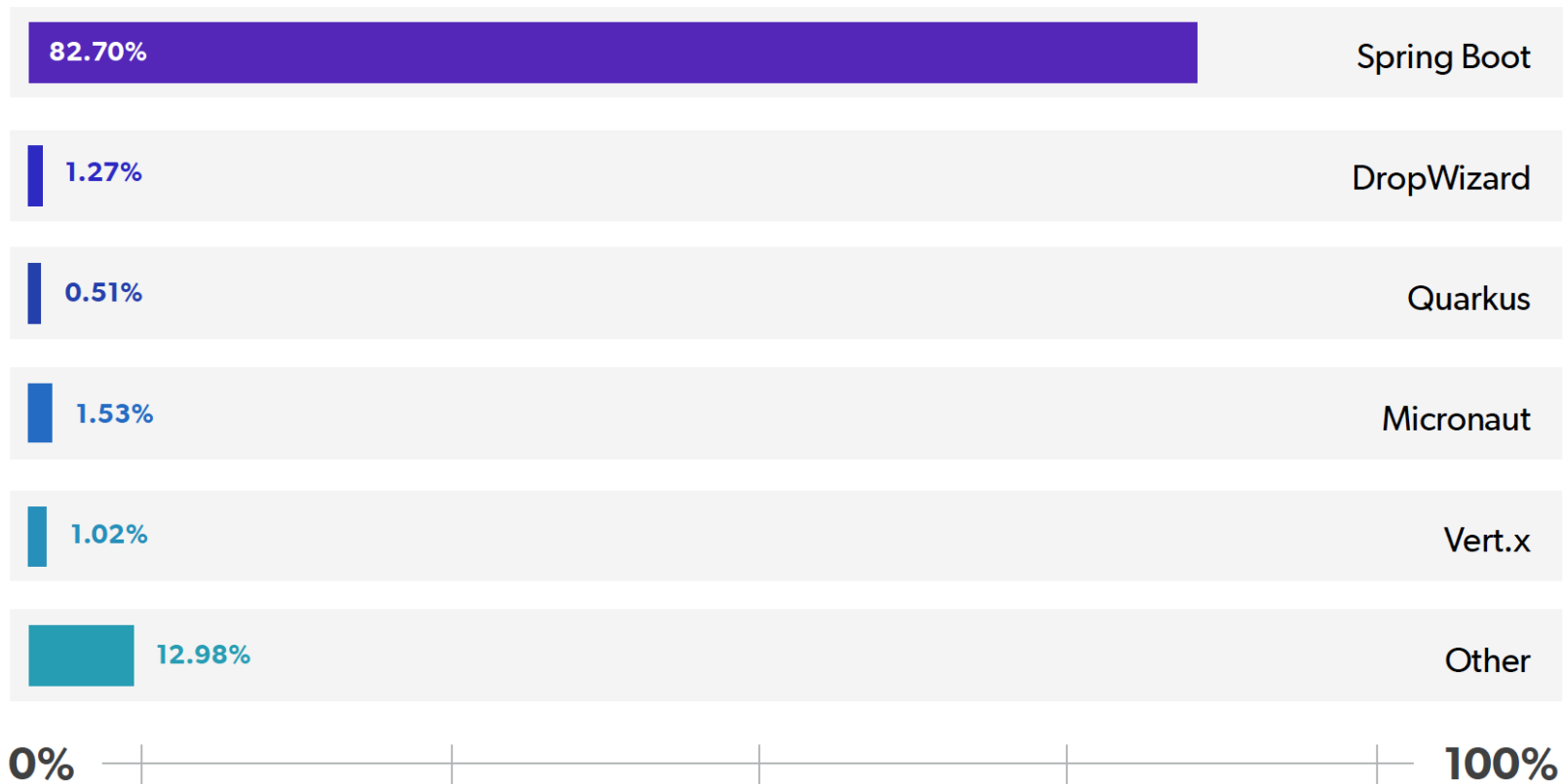




# Java Runtime Platform



What runtime platform does your application use?





# Spring Framework



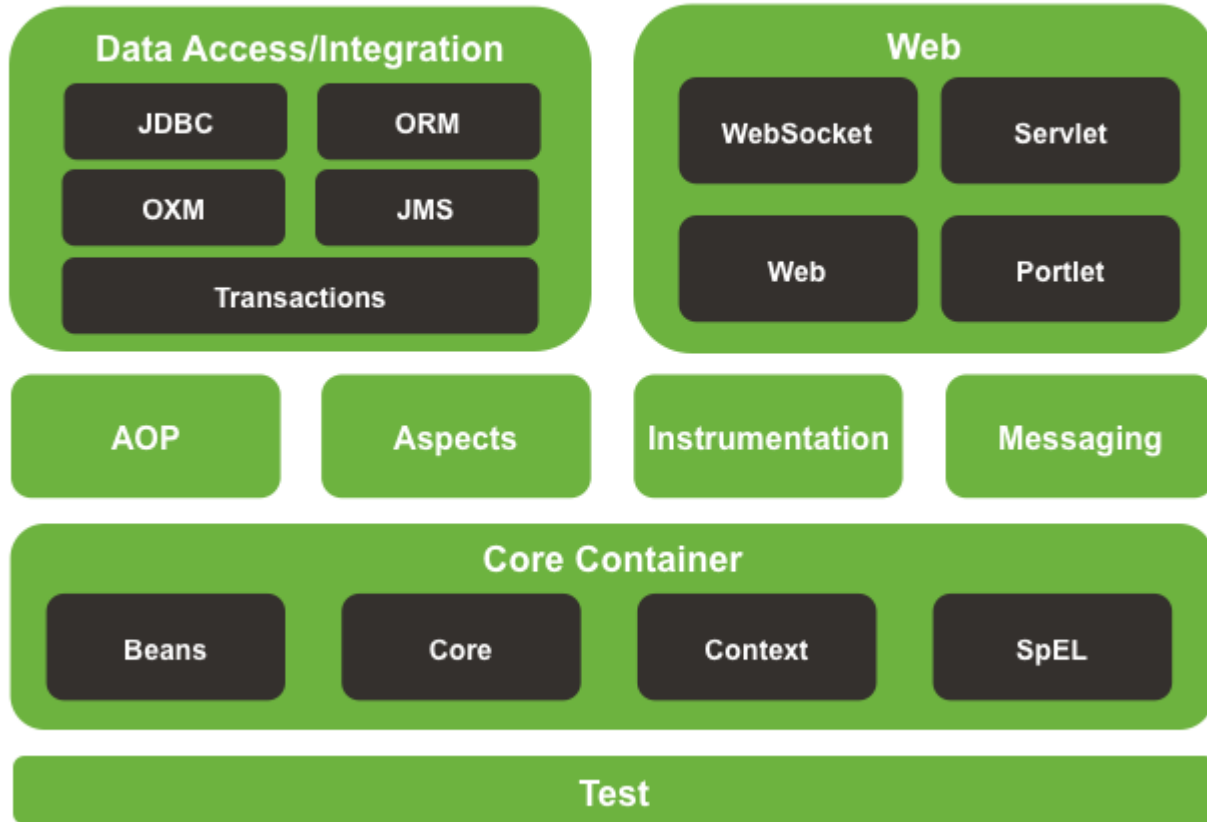
- ❖ Framework สำหรับพัฒนาแอปพลิเคชันสำหรับองค์กรบนแพลตฟอร์มของจาวา
- ❖ เป้าหมายของ Spring คือ ลดเวลาในการพัฒนา และเสนอแนวปฏิบัติที่ดีให้กับนักพัฒนา
- ❖ เป็น Open source ที่มีขนาดเล็ก (Lightweight) และสามารถเพิ่มขยายได้ง่าย



# Spring Framework



## Spring Framework Runtime





# Dependency Injection

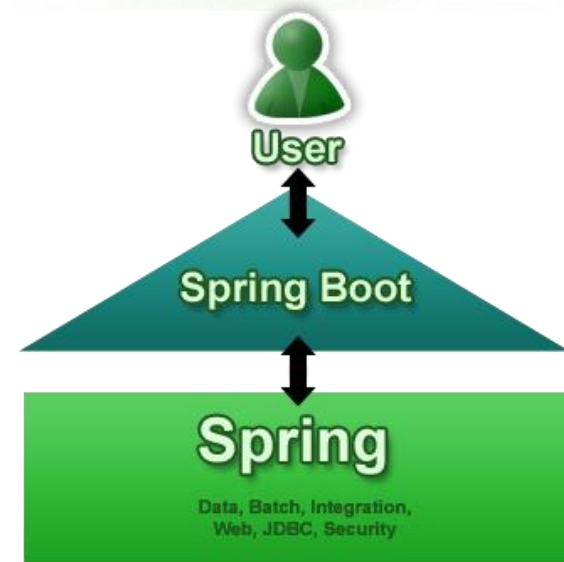


- ❖ Dependency Injection เรียกอีกชื่อหนึ่งว่า **IOC (Inversion of Control Principle)**
- ❖ เป็นวิธีการออกแบบซอฟต์แวร์ตาม pattern ที่มีการแยก component ต่าง ๆ ให้มีความเกี่ยวข้องกันน้อยที่สุด (loosely coupled)
- ❖ Component แต่ละส่วนสามารถนำมาประกอบกัน และสามารถทดสอบได้อย่างง่ายดาย
- ❖ ช่วยให้ reuse code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



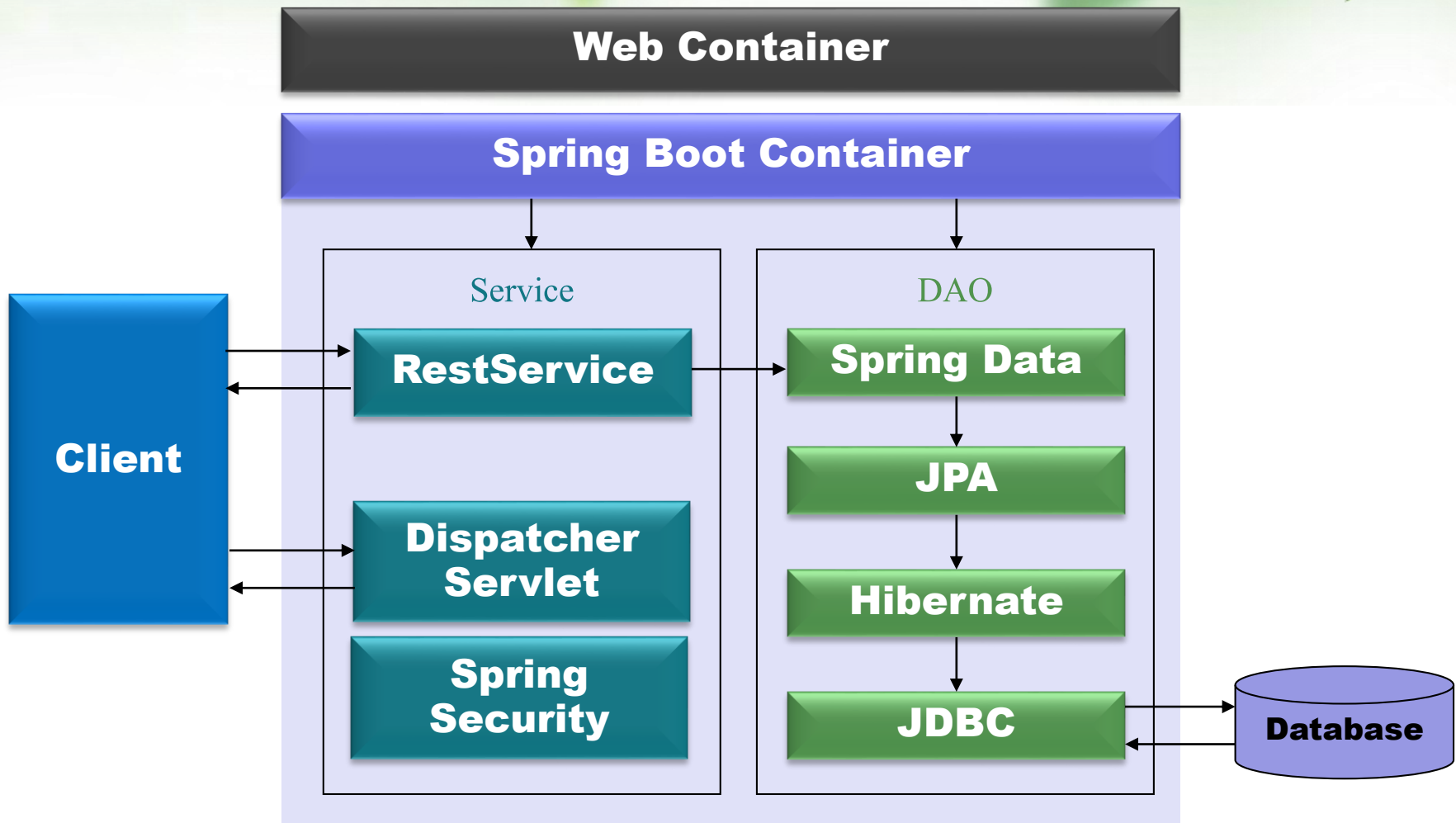
# Spring Boot

- ❖ Spring Boot ไม่ใช่ Framework แต่เป็นเครื่องมือที่ช่วยสร้าง package และรันแอปพลิเคชันที่สร้างด้วย Spring Framework
- ❖ มีชุด Starter Library ให้เรียกใช้งานได้สะดวก ลดขั้นตอนการเพิ่ม library อื่นๆ ที่เกี่ยวข้อง
- ❖ มี Auto Configuration เพื่อให้ลดขั้นตอนการกำหนดค่าต่างๆ
- ❖ มี Built-in web server เช่น Tomcat ช่วยให้อัปโหลดแอปพลิเคชันแบบแยกส่วนการทำงานได้ง่าย





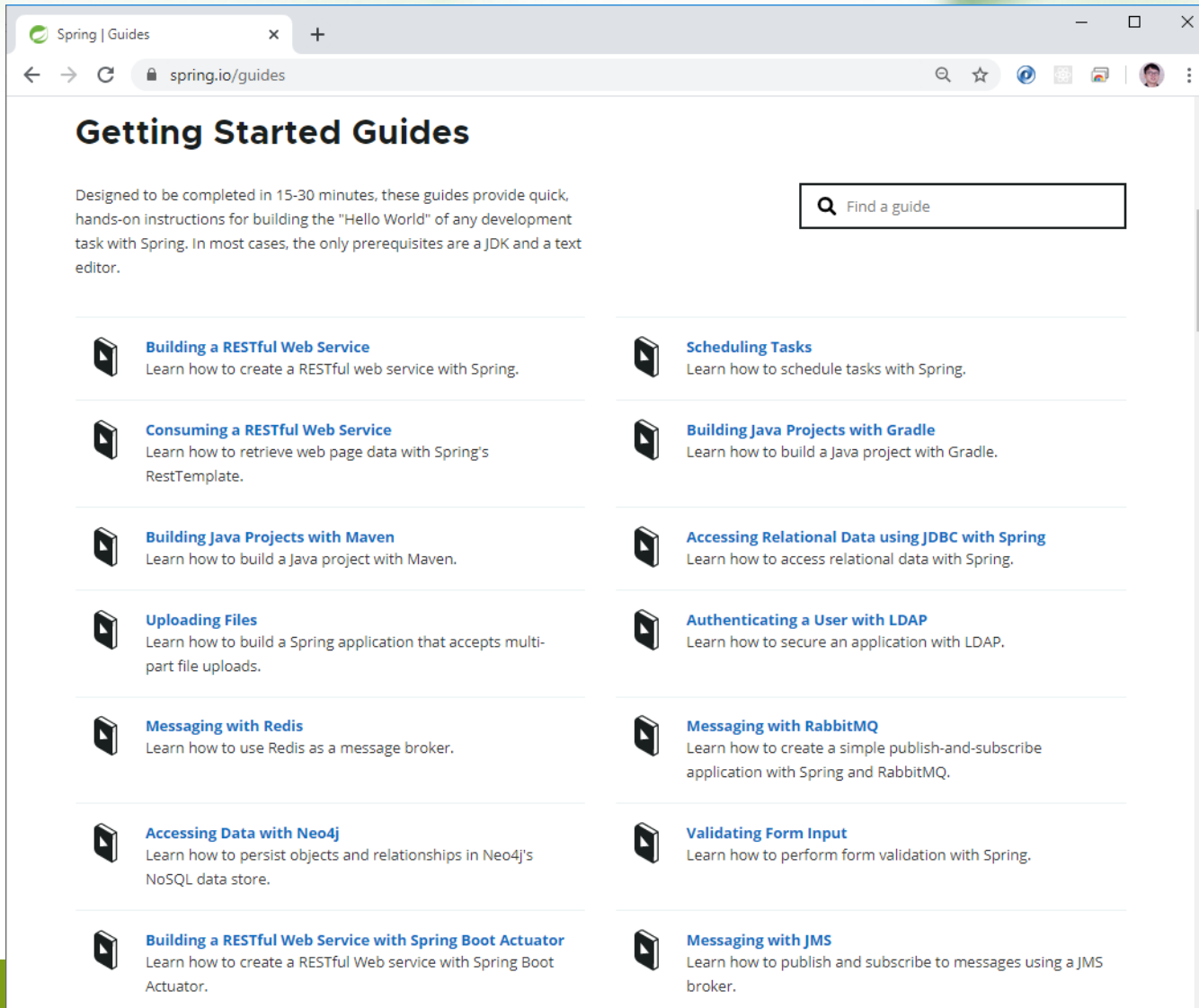
# Spring Boot Architecture





# Spring Boot ทำอะไรได้บ้าง

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



The screenshot shows a web browser window with the URL `spring.io/guides`. The page title is "Getting Started Guides". Below the title, there is a search bar with the placeholder text "Find a guide". The main content area displays a grid of 14 guide cards, each with a play button icon, a title, and a brief description.

Designed to be completed in 15-30 minutes, these guides provide quick, hands-on instructions for building the "Hello World" of any development task with Spring. In most cases, the only prerequisites are a JDK and a text editor.

Find a guide

- Building a RESTful Web Service**  
Learn how to create a RESTful web service with Spring.
- Scheduling Tasks**  
Learn how to schedule tasks with Spring.
- Consuming a RESTful Web Service**  
Learn how to retrieve web page data with Spring's RestTemplate.
- Building Java Projects with Gradle**  
Learn how to build a Java project with Gradle.
- Building Java Projects with Maven**  
Learn how to build a Java project with Maven.
- Accessing Relational Data using JDBC with Spring**  
Learn how to access relational data with Spring.
- Uploading Files**  
Learn how to build a Spring application that accepts multi-part file uploads.
- Authenticating a User with LDAP**  
Learn how to secure an application with LDAP.
- Messaging with Redis**  
Learn how to use Redis as a message broker.
- Messaging with RabbitMQ**  
Learn how to create a simple publish-and-subscribe application with Spring and RabbitMQ.
- Accessing Data with Neo4j**  
Learn how to persist objects and relationships in Neo4j's NoSQL data store.
- Validating Form Input**  
Learn how to perform form validation with Spring.
- Building a RESTful Web Service with Spring Boot Actuator**  
Learn how to create a RESTful Web service with Spring Boot Actuator.
- Messaging with JMS**  
Learn how to publish and subscribe to messages using a JMS broker.



# 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 หน้าแสดงผลโดยไม่ใช้ไค้ภาษาจาวาด้วย FreeMarker templating engine
spring-boot-starter-thymeleaf	ใช้ Render หน้าแสดงผลโดยไม่ใช้ไค้ภาษาจาวาด้วย 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



# การสร้าง Spring Boot Project



- ❖ ใช้ตัวสร้างต้นแบบจากเว็บ Spring Initializr

<https://start.spring.io/>

- ❖ จะได้ไฟล์เริ่มต้นสำหรับนำไปสร้าง Project บนเครื่องมือที่ใช้พัฒนา เช่น Eclipse ในรูปแบบ [ชื่อโปรเจก].zip



# Spring Initializr

The screenshot shows the Spring Initializr web application. The interface includes a sidebar on the left with the Spring Initializr logo and the tagline "Bootstrap your application". The main content area is divided into several sections: "Project" (Maven Project and Gradle Project), "Language" (Java, Kotlin, Groovy), "Spring Boot" (2.3.0 M2, 2.3.0 (SNAPSHOT), 2.2.5 (SNAPSHOT), 2.2.4, 2.1.13 (SNAPSHOT), 2.1.12), and "Project Metadata" (Group, Artifact, Options, Name, Description, Package name, Packaging, Java).

Annotations in Thai:

- A callout box pointing to the "2.2.4" version in the "Spring Boot" section says "เลือกที่ไม่ใช่ SNAPSHOT" (Choose one that is not SNAPSHOT).
- A callout box pointing to the "demo" artifact name in the "Project Metadata" section says "ใส่ชื่อ Project" (Put the Project name).
- A callout box pointing to the "8" version in the "Java" section says "เลือก Version Java ให้ตรงกับที่ติดตั้ง" (Choose the Java version that matches what is installed).



# Spring Initializr



The screenshot shows the Spring Initializr web interface. On the left, there is a sidebar with the text "Dependencies". The main area displays a grid of dependency options under two categories: "Developer Tools" and "Web".

- Developer Tools:**
  - Spring Boot DevTools:** Provides fast application restarts, LiveReload, and configurations for enhanced development experience. (Radio button is empty)
  - Lombok:** Java annotation library which helps to reduce boilerplate code. (Radio button is empty)
  - Spring Configuration Processor:** Generate metadata for developers to offer contextual help and where configurations (e.g. files).
- Web:**
  - Spring Web:** Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container. (Radio button is checked with a green checkmark)
  - Spring Reactive Web:** Build reactive web applications with Spring WebFlux and Netty. (Radio button is empty)
  - Spring Session:** Provides an API and implementations for managing user session information. (Radio button is empty)
  - Rest Repositories HAL Browser:** Browsing Spring Data REST repositories in your browser. (Radio button is empty)
  - Spring HATEOAS:** Eases the creation of RESTful APIs that follow the HATEOAS principle when working with Spring / Spring MVC. (Radio button is empty)

At the bottom of the interface, there are three buttons: "Generate - Ctrl + G", "Explore - Ctrl + Space", and "Share...".

เลือกชุดแม่แบบ

1. Spring Boot DevTools
2. Spring Web
3. Spring Data JPA
4. MySQL Driver

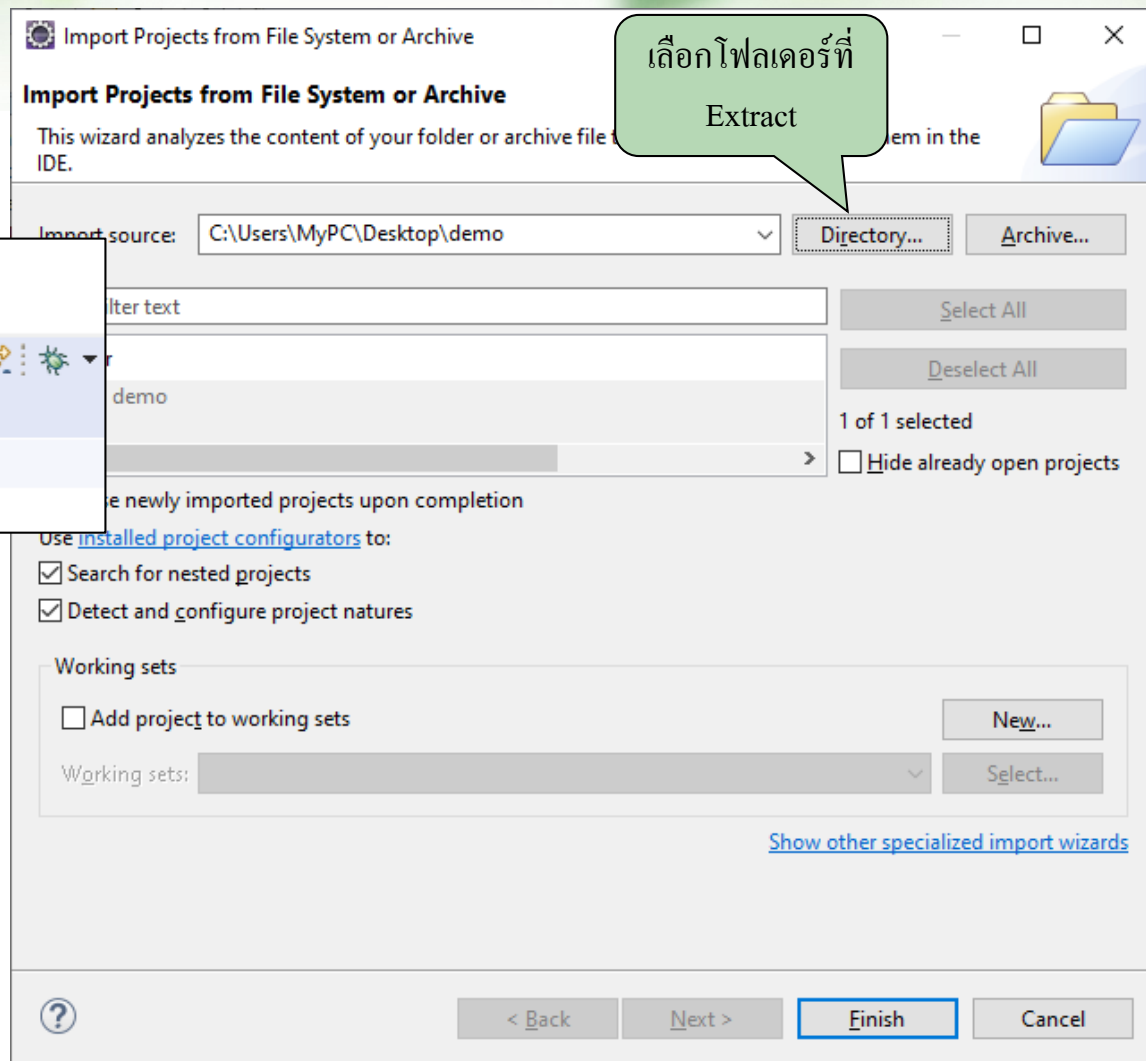
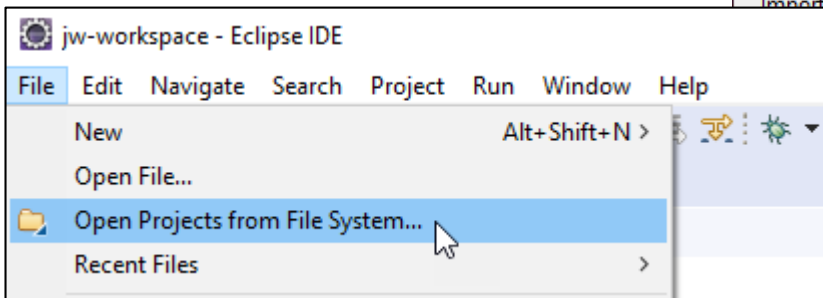
download ไฟล์เริ่มต้น  
จะได้ไฟล์ .zip



# การเปิด Project บน Eclipse



1. Extract ไฟล์ .zip ไว้ที่ใดก็ได้
2. เปิดโปรเจกต์ เลือกเมนู Open Projects from File System...





# โครงสร้างการเก็บไฟล์



## firstspring

- ▼ src/main/java → โฟลเดอร์สำหรับเก็บคลาสจาวา
  - ▼ myapi → package สำหรับเก็บ class file
    - > Customer.java
    - > CustomerRepository.java
    - > FirstController.java
    - > MainApplication.java
- ▼ src/main/resources
  - application.properties

ไฟล์ Configuration เกี่ยวกับ Spring Boot Application เช่น URL ของฐานข้อมูล



# ไฟล์ application.properties

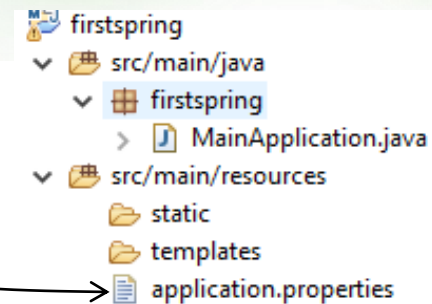


❖ การ Config ค่าเกี่ยวกับ Spring Boot Application จะถูกกำหนดในไฟล์ application.properties

❖ การเก็บข้อมูลจะอยู่ในรูปแบบ

**property-name = value**

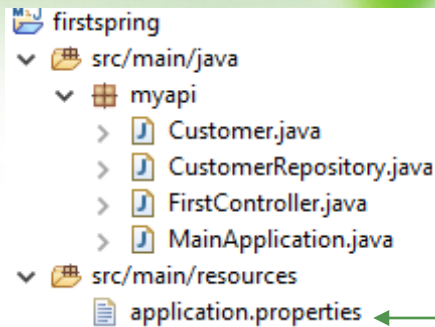
❖ แม้แบบ spring-boot-starter-web มี Web Container Tomcat ฟังมาในตัวแล้ว และ กำหนด port เป็น 8080 ดังนั้นควร Stop Server ที่เคยเปิดอยู่ เพื่อป้องกัน port ซนกัน หรือเปลี่ยน port ของ Spring Boot Application โดยเพิ่ม property ดังนี้



```
*application.properties x
1 server.port=69
2
3
4
```



# การ Config ค่าเกี่ยวกับฐานข้อมูล



กำหนดค่าสำหรับที่ติดต่อฐานข้อมูลในไฟล์ application.properties

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

ชื่อฐานข้อมูล

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

กำหนดให้แสดงคำสั่ง SQL บน console  
true - แสดง, false - ไม่แสดง

```
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
```

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



# ค่า `spring.jpa.hibernate.ddl-auto`

- ❖ 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 World"; ● ส่ง HTTP response กลับไปด้วยข้อความธรรมดา (plain text)
```

```
}
```

```
}
```

# คลาสสำหรับรัน Spring Boot Application

- ❖ Spring Initializr จะสร้างคลาสหลักสำหรับการรัน 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);
```

```
    }
```

```
}
```

↓  
สั่งให้ Spring Boot Application เริ่มทำงาน

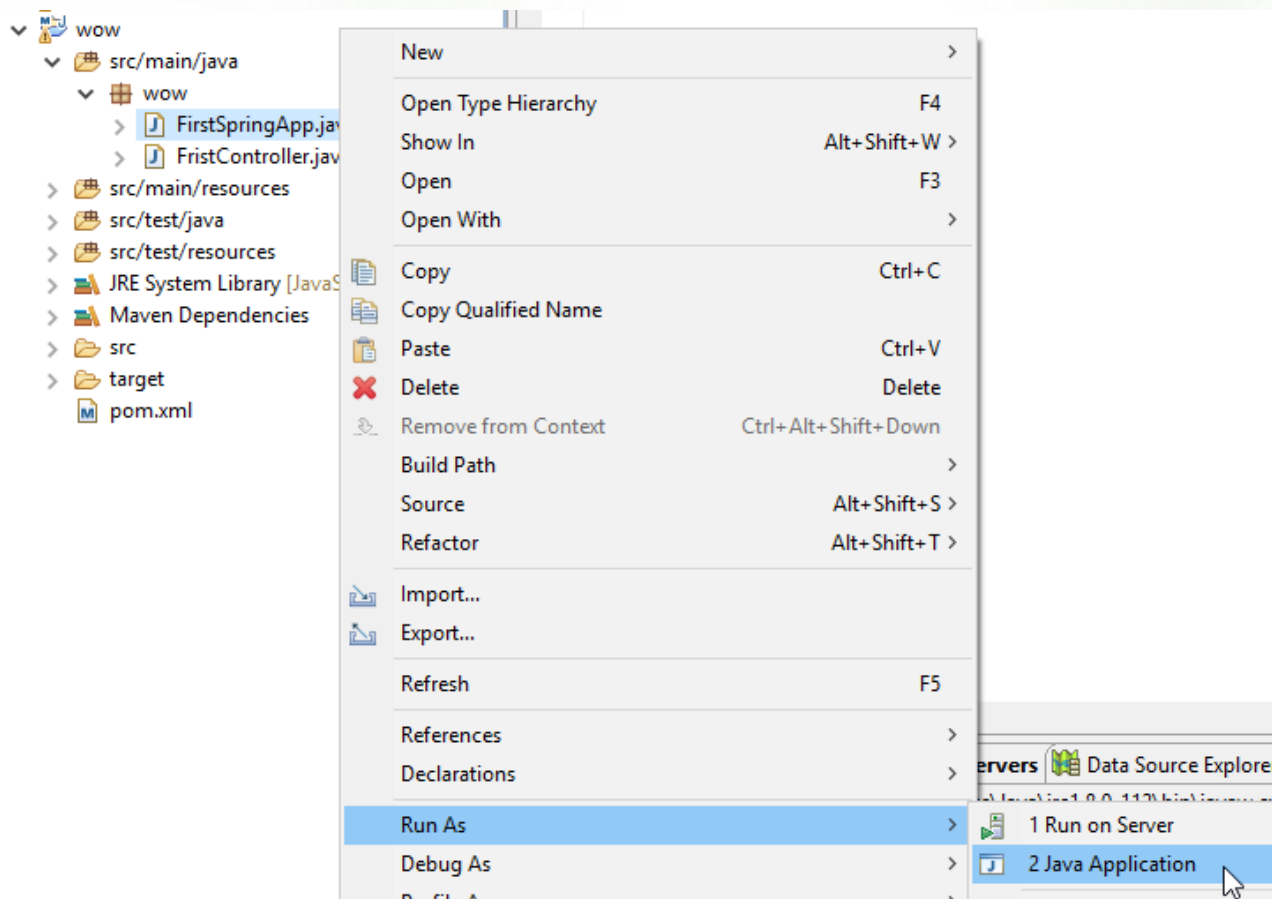
↓  
ใช้รับค่าทาง command line



# การรัน Spring Boot Application



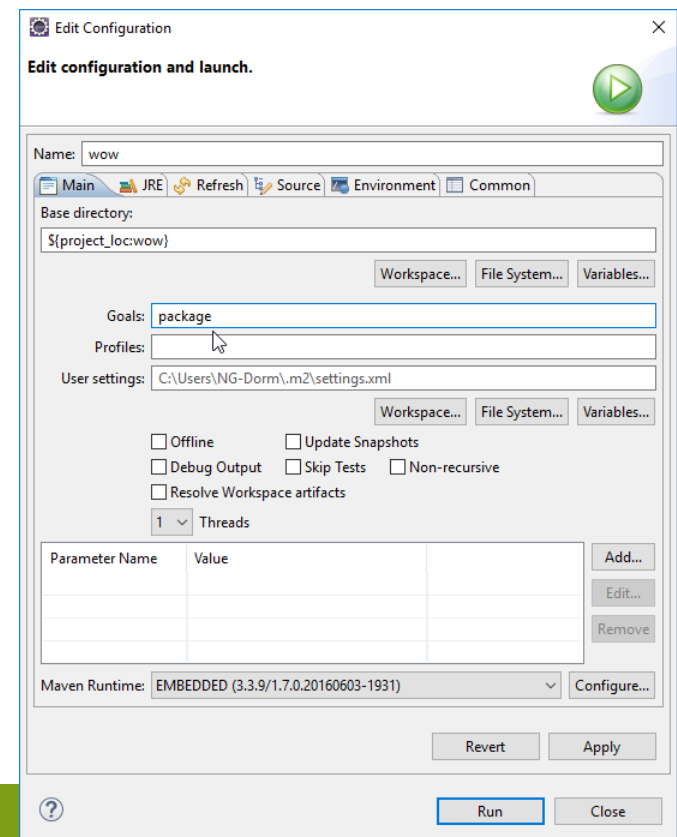
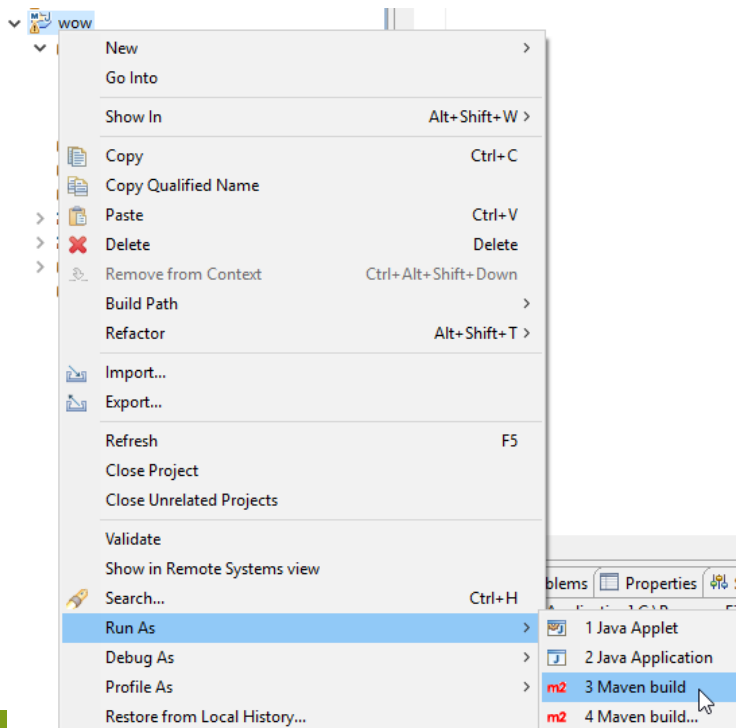
❖ คลิกขวาที่คลาสหลัก เลือก Run As > Java Application





# 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



1. เลือกแท็บ JRE

2. คลิก

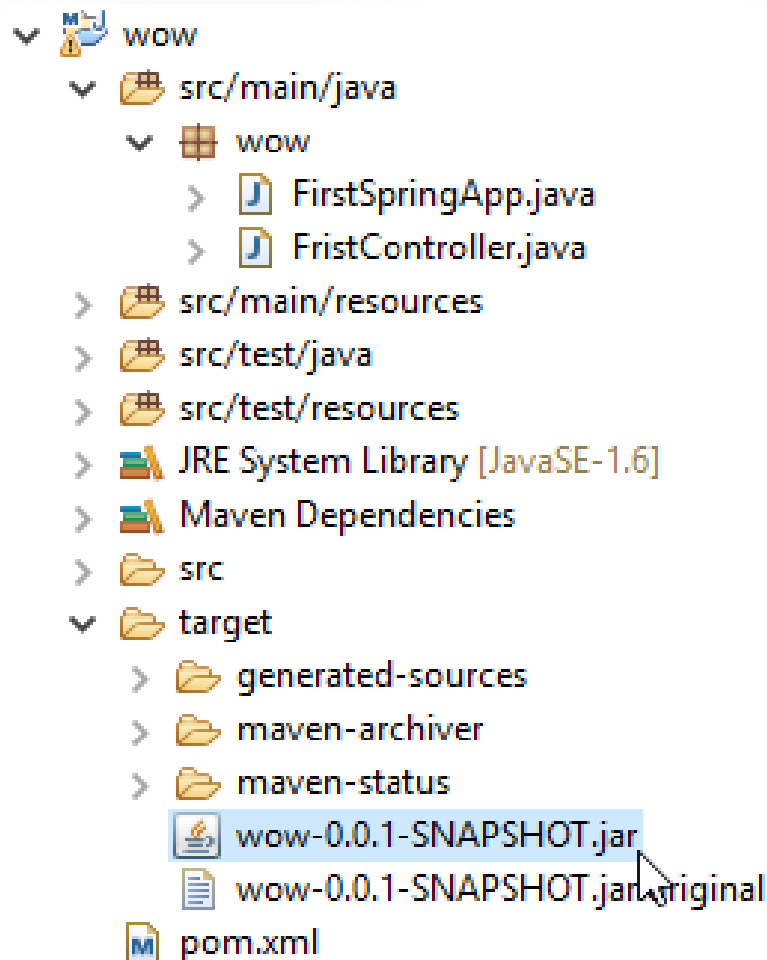
4. เริ่ม package

3. เลือก JavaSE Version ที่ตรงกับ Project

Apply and Close

# jar file ที่ได้

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

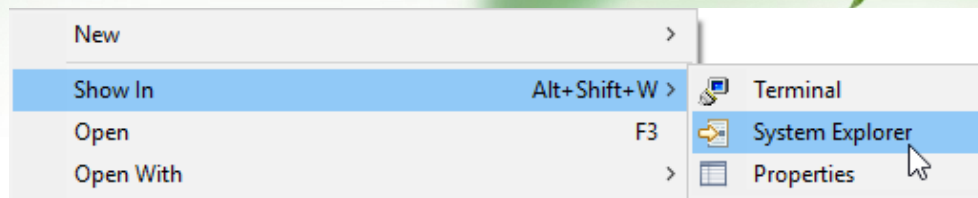




# การรัน Executable jar file



- ❖ คลิกขวาที่โฟลเดอร์ target  
เลือก Show In > System Explorer



- ❖ กดปุ่ม Shift ค้างไว้แล้วคลิกขวาที่โฟลเดอร์ target  
เลือก Open command window here

- ❖ พิมพ์คำสั่งเพื่อเปิดการทำงาน  
ไฟล์ .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)
s\NG-Dorm\workspace\wow\target)
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
```