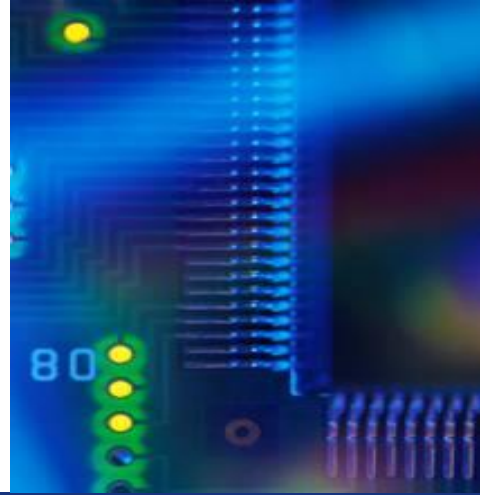
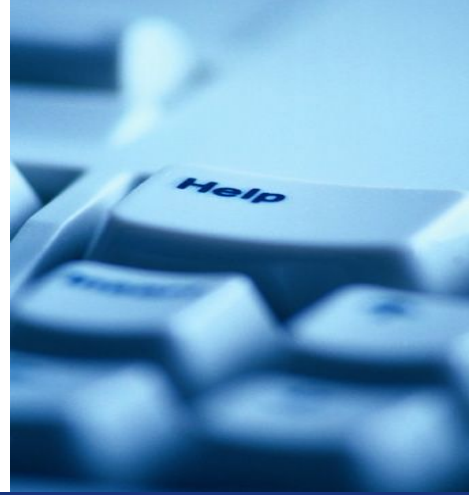




**UNIVERSITAS
BUDI LUHUR**



FAKULTAS TEKNOLOGI INFORMASI

Java Web Programming

[PG119/ 3 SKS]

Riskiana Wulan, MKom

Pertemuan 9

Overview of JAVA Servlet , Spring Framework and Spring Boot

Agenda

- **Class Announcement**
- **Overview of Spring Framework**
- **Hands on Lab**

Class Announcement

1. Kuliah Pengganti

Week	Date	Note	KP
9	2024-11-20		
10	2024-11-27		
11	2024-12-04		
12	2024-12-11		
13	2024-12-18		
14	2024-12-25	CUTI BERSAMA	Senin, 09-12-2024 jam 16.10
15	2025-01-01	CUTI BERSAMA	Senin, 16-12-2024 jam 16.10
16	2025-01-08		

Presensi

1. **Wajib melampirkan bukti kuliah pengganti jika ada kelas yang bentrok**
2. **Wajib melampirkan surat izin jika bentrok dengan partisipasi kegiatan acara atau seminar**

What are Java Servlet?

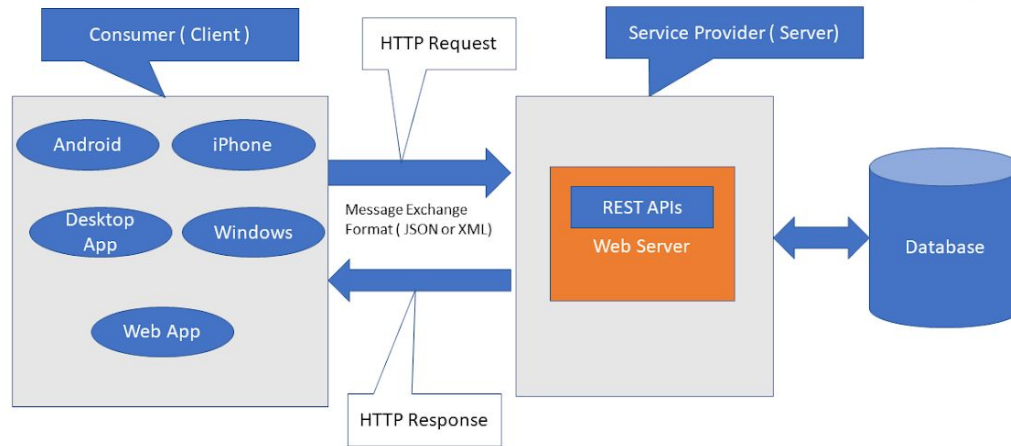
Servlets are Java programs that run on a server to handle HTTP requests and responses.
Part of the Java EE (Jakarta EE) specification.

Key features:

- **Direct control over HTTP methods like `GET`, `POST`, etc.**
- **Requires manual setup for routing, session handling, and dependency management.**

REST API

REST – Architecture



https://www.javaguides.net/p/rest-api-tutorial.html#google_vignette

Download java :

<https://www.oracle.com/id/java/technologies/downloads/>

<https://www.openlogic.com/openjdk-downloads>

Earlier version :

<https://www.oracle.com/java/technologies/javase/javase8u211-later-archive-downloads.html>

Note : gunakan Java version LTS (Long term Support)

Java release history :

https://en.wikipedia.org/wiki/Java_version_history

<https://blogs.oracle.com/javamagazine/post/java-long-term-support-lts>

Build Tools

Maven : <https://maven.apache.org/>

Download : <https://maven.apache.org/download.cgi>

Installation : <https://maven.apache.org/install.html>

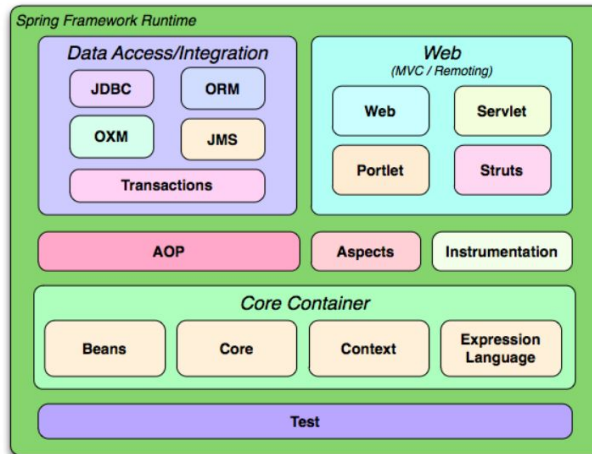
Check maven version : `mvn -version`

Gradle : <https://gradle.org/>

[ant vs maven vs gradle](#)

What is Spring Framework?

- **Spring is a comprehensive framework for building enterprise-level Java applications.**
- **Simplifies web development with Spring MVC and Spring Boot.**
- **Key features:**
 - Built-in dependency injection.
 - Simplified web application development with annotations.
 - Modular architecture with reusable components.



Overview of Spring Framework

<https://docs.spring.io/spring-framework/docs/3.0.x/spring-framework-reference/html/overview.html>

Spring Boot

Merupakan sebuah module dari spring framework, bertujuan untuk menyederhanakan pengembangan aplikasi berbasis Spring.

- Banyak digunakan untuk mendevelop REST APIs
- Autoconfiguration feature memiliki kemampuan untuk secara otomatis
- Standalone application
- Mengurangi boilerplate code. Contoh: Kita tidak perlu lagi membuat code untuk koneksi ke database. Semuanya sudah disediakan oleh spring boot
- Menawarkan embedded server seperti jetty dan tomcat

Spring boot Release Note: <https://endoflife.date/spring-boot>

Java Release history : https://en.wikipedia.org/wiki/Java_version_history



HANDS on LAB

Build My First Spring Boot Project

1. Buka spring initializr melalui link ini
<https://start.spring.io/>

2. Isi Project Metadata
Pilih Project : Gradle-Groovy
Language: Java , pilih versi 21
Spring boot : 3.2.11

Project metadata dibawah ini bisa diisi bebas,
contoh:



Group: com.ubl
Artifact: myapp
Name: myapp

Project

☒ Gradle - Groovy ☐ Gradle - Kotlin ☐ Maven

Language

☒ Java ☐ Kotlin ☐ Groovy

Spring Boot

☐ 3.4.0 (SNAPSHOT) ☐ 3.4.0 (RC1) ☐ 3.3.6 (SNAPSHOT) ☐ 3.3.5 ☐ 3.2.12 (SNAPSHOT) ☒ 3.2.11

Project Metadata

Group

Artifact

Name

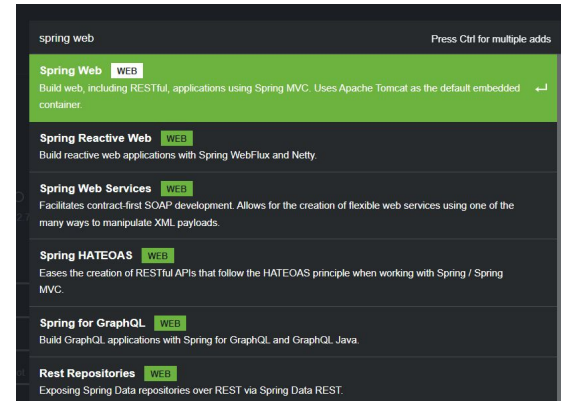
Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 23 ☒ 21 ☐ 17

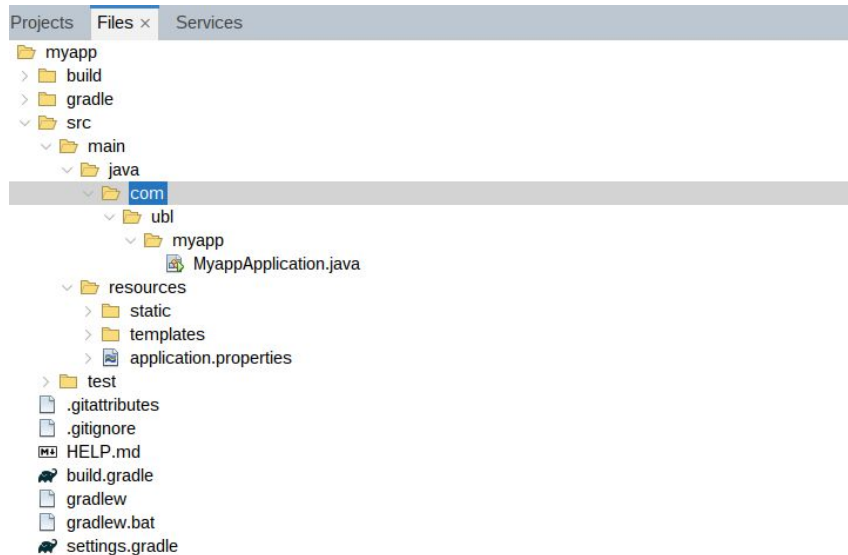
3. Lalu pilihan dependencies, ketik Spring web lalu ok



4. Click generate, maka project akan otomatis terdownload ke local computer

5. Extract zip file ke local directory kamu

6. Lalu buka project melalui IDE (Netbeans)



src/main: seluruh source codes berada pada directory ini
src/main/resources : Directory yang berisi resources yang digunakan untuk konfigurasi spring boot. Biasanya terdiri dari:

- Application.properties atau application.yml: file konfigurasi untuk mendefinisikan properties application seperti database connection details , server port, dll.

Reference:

<https://docs.spring.io/spring-boot/docs/current/reference/html/application-properties.html>

- Static : dapat berupa css files, images, javascript
- Templates: jika menggunakan spring template view engine (tymeleaf atau freemarker) atau jsp, maka file-file template tersebut diletakkan pada directory ini

src/main/java : Directory yang berisi seluruh java files

src/main/test : lokasi source code untuk unit test dan integration test (mirror dari /src/main structure)

Target: directory ini bukan bagian dari source code, namun isinya adalah hasil generated files (compiled java class, packaged jar file, artifact, dll) yang digenerate oleh build tools (Maven/Gradle).

Gradle

What is Gradle?

- **Builds automation system**
- **Manages the build cycle via a series of tasks (for example, compiles Java sources, runs tests, installs app to device)**
- **Determines the proper order of tasks to run**
- **Manages dependencies between projects and third-party libraries**

Gradle build file

- **Declare plugins**
- **Define Android properties**
- **Handle dependencies**
- **Connect to repositories**

build.gradle

is a build configuration for a Java project using Gradle as the build tool.

Plugins

- Adds the Java plugin for Java project support.
- Includes the Spring Boot plugin (v3.2.11) to build Spring Boot applications.
- Adds Spring dependency management (v1.1.6) for version compatibility.

Group and Version

- **group:** Sets the package namespace to `com.ubl`.
- **version:** Marks the project version as `0.0.1-SNAPSHOT` (development stage).

Java Toolchain

- Configures Java 21 as the language version for building the project.

Repositories

- Uses Maven Central to fetch project dependencies.
- <https://mvnrepository.com/>

```

1  plugins {
2      id 'java'
3      id 'org.springframework.boot' version '3.2.11'
4      id 'io.spring.dependency-management' version '1.1.6'
5  }
6
7  group = 'com.ubl'
8  version = '0.0.1-SNAPSHOT'
9
10 java {
11     toolchain {
12         languageVersion = JavaLanguageVersion.of(21)
13     }
14 }
15
16 repositories {
17     mavenCentral()
18 }
19
20 dependencies {
21     implementation 'org.springframework.boot:spring-boot-starter-web'
22     testImplementation 'org.springframework.boot:spring-boot-starter-test'
23     testRuntimeOnly 'org.junit.platform:junit-platform-launcher'
24 }
25
26 tasks.named('test') {
27     useJUnitPlatform()
28 }

```

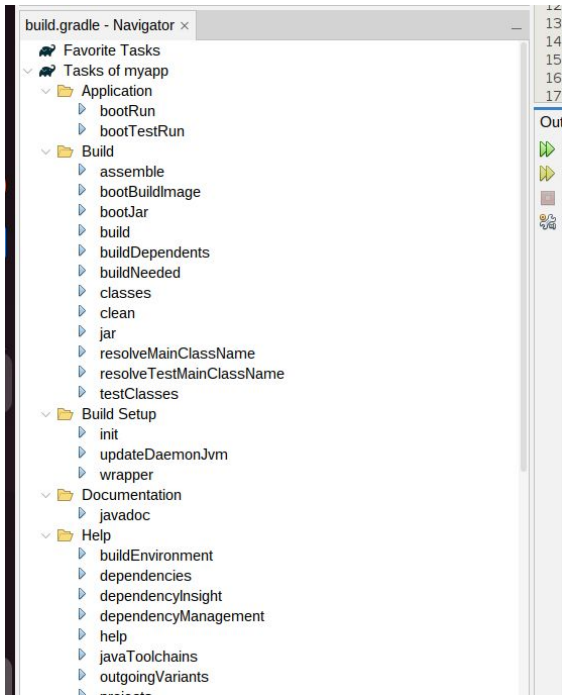
Dependencies

- Adds Spring Boot web starter for building web apps.
- Includes Spring Boot test starter for testing.
- Specifies the JUnit Platform launcher for runtime test execution.

Test Configuration

- Ensures tests are run using the JUnit Platform (JUnit 5).

Build.gradle Navigator



Clean and Build my project

```
Output - Build (myapp) x
Downloading Gradle 8.10.2...
JAVA_HOME="/usr/lib/jvm/java-21-openjdk-amd64"
cd /home/riskiana/data/ubl/myapp; ./gradlew --configure-on-demand clean build
Configuration on demand is an incubating feature.
> Task :clean UP-TO-DATE
> Task :compileJava
> Task :processResources
> Task :classes
> Task :resolveMainClassName
> Task :bootJar
> Task :jar
> Task :assemble
> Task :compileTestJava
> Task :processTestResources NO-SOURCE
> Task :testClasses
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
> Task :test
> Task :check
> Task :build

BUILD SUCCESSFUL in 52s
8 actionable tasks: 7 executed, 1 up-to-date
```

Check gradle version:
Ketik gradle -version pada terminal

```
riskiana@riskiana-workstation:~/data/dev/idea-IU-232.10072.27/bin$ gradle -version
```

Welcome to Gradle 8.4!

Here are the highlights of this release:

- Compiling and testing with Java 21
- Faster Java compilation on Windows
- Role focused dependency configurations creation

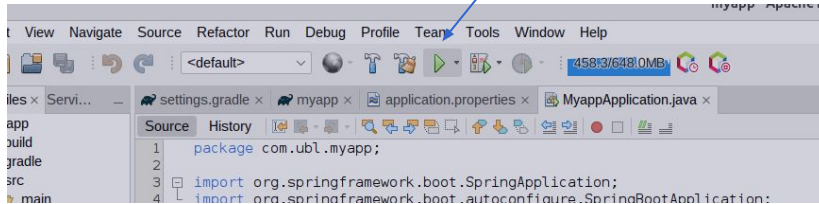
For more details see <https://docs.gradle.org/8.4/release-notes.html>

Gradle 8.4

RUN MY FIRST SPRING BOOT APPLICATION

Run Project dengan klik green arrow

Output



Output

Output - Run (myapp) x

```
> JAVA_HOME="/usr/lib/jvm/java-21-openjdk-amd64"
> cd /home/riskiana/data/ubl/myapp; ./gradlew --configure-on-demand -x check bootRun
Configuration on demand is an incubating feature.
> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :resolveMainClassName UP-TO-DATE

> Task :bootRun
```

```

  ____ _
 / ___ \| | | |
/ /___ \| |_| |
 \___ \| __|_|
  _____
:: Spring Boot ::
                (v3.2.11)

```

```

2024-11-20T09:37:36.778+07:00 INFO 376962 --- [myapp] [main] com.ubl.myapp.MyappApplication : Starting MyappApplication using Java 21.0.5 with PID 376962
2024-11-20T09:37:36.781+07:00 INFO 376962 --- [myapp] [main] com.ubl.myapp.MyappApplication : No active profile set, falling back to 1 default profile
2024-11-20T09:37:37.517+07:00 INFO 376962 --- [myapp] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2024-11-20T09:37:37.532+07:00 INFO 376962 --- [myapp] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-11-20T09:37:37.533+07:00 INFO 376962 --- [myapp] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.31]
2024-11-20T09:37:37.580+07:00 INFO 376962 --- [myapp] [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2024-11-20T09:37:37.581+07:00 INFO 376962 --- [myapp] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1.457 seconds
2024-11-20T09:37:37.870+07:00 INFO 376962 --- [myapp] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path ''
2024-11-20T09:37:37.876+07:00 INFO 376962 --- [myapp] [main] com.ubl.myapp.MyappApplication : Started MyappApplication in 1.457 seconds (process running for 1.457 seconds)
2024-11-20T09:42:19.518+07:00 INFO 376962 --- [myapp] [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2024-11-20T09:42:19.519+07:00 INFO 376962 --- [myapp] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2024-11-20T09:42:19.520+07:00 INFO 376962 --- [myapp] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms

```

Perhatikan port :8080

Access aplikasi dengan <http://localhost:8080>



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Wed Nov 20 09:45:21 WIB 2024

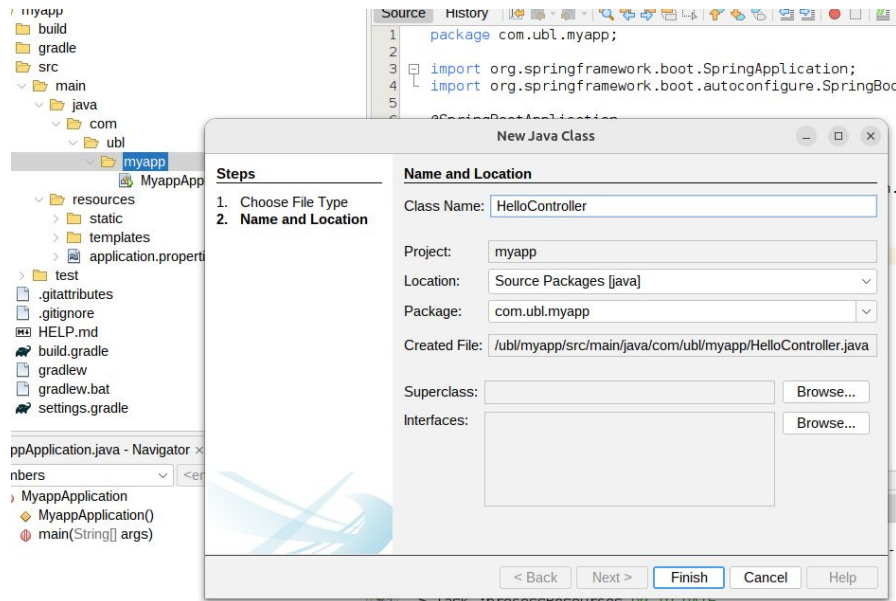
There was an unexpected error (type=Not Found, status=404).

Terlihat belum ada endpoint yang tersedia sehingga muncul Whitelabel Error

Membuat Endpoint pertama dengan REST

1. Buat Controller class dengan nama HelloController.java

Tambahkan endpoint /hello dengan code seperti dibawah ini:



```

6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

/**
 * @author riskiana
 */
@RestController
public class HelloController {

    @GetMapping("/hello")
    public String hello(){
        return "Welcome UBL Student";
    }
}

```

Run kembali application

Akses dengan browser

<http://localhost:8080/hello>

Akan muncul response Welcome UBL Student



← → ↻ ⓘ localhost:8080/hello

Welcome UBL Student

LATIHAN

1. Buat 1 endpoint baru dengan HTTP GET yang jika diakses akan menampilkan String Hello {Nama Kamu}

<http://localhost/{endpoint}>

Output:

Hello Ahmad

2. Buat account pada github.com