

2021.06.10 HTML/JS/CSS

〈HTML 시작 전 SETTING〉

전제적인 흐름만 가져가고

정확한 개념은 나중에 설명 듣고 다시 정리 (2021.06.10 기본설정.pdf랑 같이 보면 좋을듯)



Project

☐ Maven Project ☒ Gradle Project

Language

☒ Java ☐ Kotlin ☐ Groovy

Spring Boot

☐ 2.5.1 (SNAPSHOT) ☒ 2.5.0 ☐ 2.4.7 (SNAPSHOT) ☐ 2.4.6
☐ 2.3.12 (SNAPSHOT) ☐ 2.3.11

Project Metadata

Group com.example

Artifact demo

Name demo

Description Demo project for Spring Boot

Package name com.example.demo

Packaging ☒ Jar ☐ War

Java ☐ 16 ☒ 11 ☐ 8

Dependencies

ADD DEPENDENCIES... CTRL + B

Lombok DEVELOPER TOOLS

Java annotation library which helps to reduce boilerplate code.

Spring Web WEB

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

Thymeleaf TEMPLATE ENGINES

A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

JDBC API SQL

Database Connectivity API that defines how a client may connect and query a database.

Spring Data JPA SQL

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

MySQL Driver SQL

MySQL JDBC and R2DBC driver.

HtmlTestControl.java × build.gradle (demo) × application.yaml × htmlhi.html × DemoApplication.java ×

```
package com.example.demo.controller;

import lombok.extern.slf4j.Slf4j;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Slf4j
@Controller
public class HtmlTestControl {

    //HtmlTestController가 제어하는 요소를 디버깅하도록 서포트하는 객체체
    // static final Logger log = LoggerFactory.getLogger(HtmlTestControl.class);

    @GetMapping("/")
    public String firstIndex(){
        // 위에 Logger 사용하는 방식은 Lombok이 없을 경우 사용함
        // 그래서 이클립스에서는 상당히 불편하게 구성해야하는데 비해
        // 여기서는 맨 위쪽의 @Slf4j를 추가함으로써 위의
        // static final Logger log = LoggerFactory.getLogger(HtmlTestControl.class); 파트를 적어줄 필요가 없어짐
        log.info("firstIndex() 실행");
        return "htmlHi";
    }
}
```

estControl.java × build.gradle (demo) × application.yaml × htmlhi.html × DemoApplication.java ×

```
plugins {
    id 'org.springframework.boot' version '2.5.0'
    id 'io.spring.dependency-management' version '1.0.11.RELEASE'
    id 'java'
}

group = 'com.example'
version = '0.0.1-SNAPSHOT'
sourceCompatibility = '11'

configurations {
    compileOnly {
        extendsFrom annotationProcessor
    }
}

repositories {
    mavenCentral()
}

dependencies {
    // implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    // implementation 'org.springframework.boot:spring-boot-starter-jdbc'
    implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
    implementation 'org.springframework.boot:spring-boot-starter-web'
    compileOnly 'org.projectlombok:lombok'
    runtimeOnly 'mysql:mysql-connector-java'
    annotationProcessor 'org.projectlombok:lombok'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
}

test {
    useJUnitPlatform()
}
```

나는 java 16 version 쓰는데

spring initializer에서 setting한 11로 그대로 두고 진행.

jpa팔기 전까지 여기 주석 필수

src/main/resources/application.properties >> application.yaml로 바꿔줘야 하는데 이유는 나중에 설명되고 update할 것.

```
TestControl.java × build.gradle (demo) × application.yaml ×  
server:  
  port: 7777
```

```
TestControl.java × build.gradle (demo) × application.yaml × htmlhi.html ×  
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8">  
    <title>첫 번째 테스트 파일 입니다.</title>  
  </head>  
  <body>  
    <h1> HTML5/CSS3</h1>  
    <h2>React/Vue/Svelte</h2>  
    <h3>JPA</h3>  
    <h4>ASP.NET</h4>  
    <h5>Web Assembly</h5>  
  
    <p>GOOD</p>  
  </body>  
</html>
```

HTML 부분.

```
estControl.java × DemoApplication.java × build.gradle (demo) × application.yaml × htmlhi.html ×  
package com.example.demo;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class DemoApplication {  
    public static void main(String[] args) { SpringApplication.run(DemoApplication.class, args); }
```

위에처럼 setting 하고
DemoApplication.java에서
run 하면

서버
구동됨.

```
오후 11:49:35: Executing task 'DemoApplication.main()'...
```

```
Starting Gradle Daemon...
Gradle Daemon started in 1 s 732 ms
> Task :compileJava
> Task :processResources UP-TO-DATE
> Task :classes

> Task :DemoApplication.main()
```

```

      .   _--_       _-_-       -_- \ \ \ \
    / \  _--_   _-_-   _-_-   _-_- \ \ \ \
( C )_--_ | ' | ' | ' | ' V _' \ \ \ \ \
\W/_--_| C)| | | | | | | C_| ) ) ) )
 ' _--_ | _-_-| | | | | _-\_ / / / /
=====|_=====|_/=/././
:: Spring Boot ::                (v2.5.0)

```

(D:\java_work\LectureContents\javascript\JuhyungLee\demo\build\classes\java\main started by Samuel in D:\java_work\LectureContents\javascript\JuhyungLee\demo)

```

2021-06-10 23:49:49.870 INFO 30236 --- [main] com.example.demo.DemoApplication : Starting DemoApplication using Java 16 on DESKTOP-VVE1E8K with PID 36 [D:\java_work
2021-06-10 23:49:49.877 INFO 30236 --- [main] com.example.demo.DemoApplication : No active profile set, falling back to default profiles: default
2021-06-10 23:49:50.759 INFO 30236 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 7777 (http)
2021-06-10 23:49:50.768 INFO 30236 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2021-06-10 23:49:50.769 INFO 30236 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.46]
2021-06-10 23:49:50.860 INFO 30236 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2021-06-10 23:49:50.861 INFO 30236 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 939 ms
2021-06-10 23:49:51.200 INFO 30236 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 7777 (http) with context path ''
2021-06-10 23:49:51.208 INFO 30236 --- [main] com.example.demo.DemoApplication : Started DemoApplication in 1.735 seconds (JVM running for 2.076)
2021-06-10 23:49:51.209 INFO 30236 --- [main] o.s.b.a.ApplicationAvailabilityBean : Application availability state LivenessState changed to CORRECT
2021-06-10 23:49:51.211 INFO 30236 --- [main] o.s.b.a.ApplicationAvailabilityBean : Application availability state ReadinessState changed to ACCEPTING_TRAFFIC
2021-06-10 23:49:59.724 INFO 30236 --- [n 1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2021-06-10 23:49:59.724 INFO 30236 --- [n 1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2021-06-10 23:49:59.725 INFO 30236 --- [n 1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
2021-06-10 23:49:59.746 INFO 30236 --- [n 1] c.e.demo.controller.HtmlTestControl : firstIndex()

```



서버 구동시키면

내가 만들어 놓은 html page를 볼 수 있음