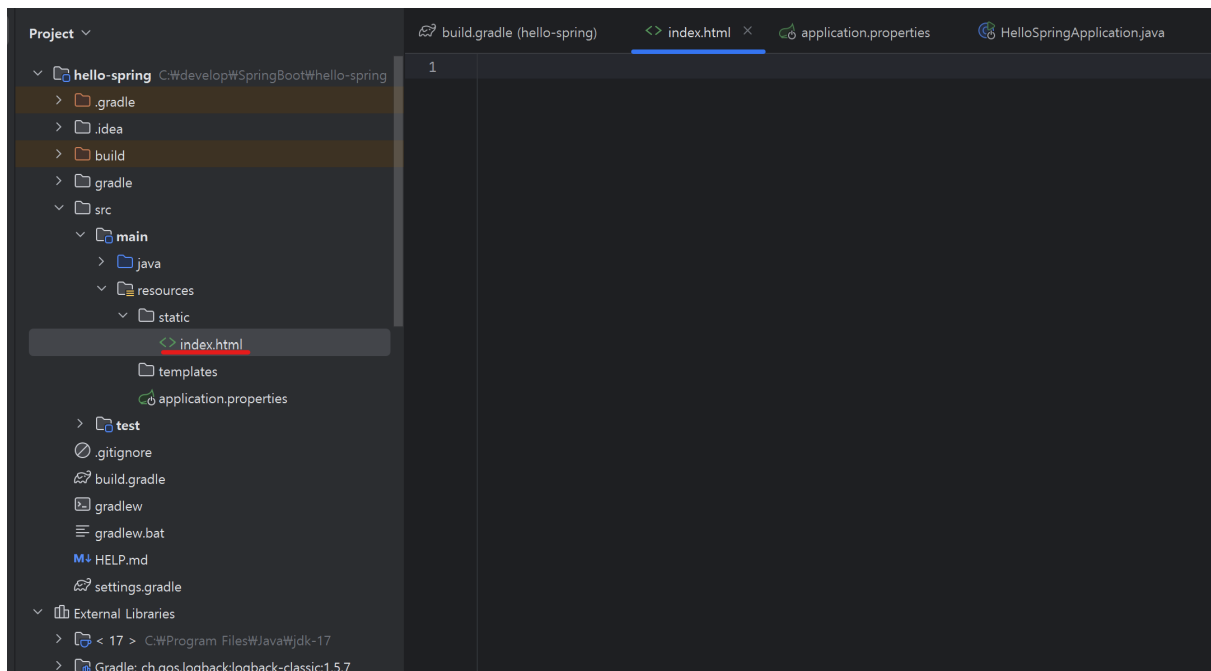


240905 - View 환경설정

- Welcome Page만들기

resources/static/index.html - 스프링 부트는 저 경로에 넣으면 자동으로 웰컴페이지로 지정해준다



- SpringBoot는 너무나 방대해서 필요한 걸 찾는 능력이 중요한데 이 방법을 적어본다

<https://spring.io/> 접속

Documentation

Each **Spring project** has its own; it explains in great details how you can use **project features** and what you can achieve with them.

3.3.3

CURRENT

GA

Reference Doc.

Api Doc.

Spring Boot / Reference / Web / Servlet Web Applications

#Welcome Page

Spring Boot supports both static and templated welcome pages. It first looks for an `index.html` file in the configured static content locations. If one is not found, it then looks for an `index` template. If either is found, it is automatically used as the welcome page of the application.

This only acts as a fallback for actual index routes defined by the application. The ordering is defined by the order of `HandlerMapping` beans which is by default the following:

<code>RouterFunctionMapping</code>	Endpoints declared with <code>RouterFunction</code> beans
<code>RequestMappingHandlerMapping</code>	Endpoints declared in <code>@Controller</code> beans
<code>WelcomePageHandlerMapping</code>	The welcome page support

- thymeleaf 템플릿 엔진 사용.
 - thymeleaf 공식 사이트 : <https://www.thymeleaf.org/>

HelloController

```
package hello.hello_spring.controller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class HelloController {
```

```

    @GetMapping("hello")
    public String hello(Model model) {
        model.addAttribute("data", "Hello!!");
        return "hello";
    }
}

```

data에 hello!! 를 담는다.

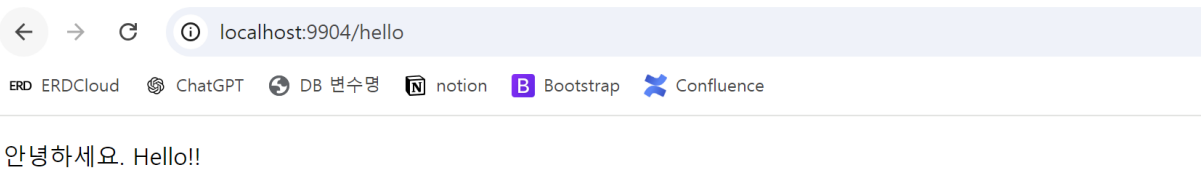
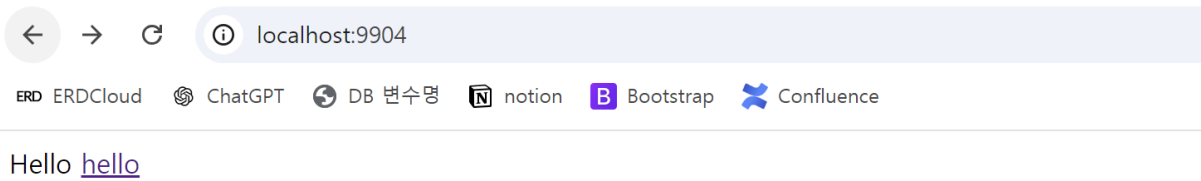
hello.html

```

<!doctype html>
<!-- thymeleaf.org 링크를 걸어서 타임리프문법을 사용 가능. -->
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>Hello</title>
    <meta http-equiv="Content-Type" content="text/html; chars
</head>
<body>
<p th:text="'안녕하세요. ' + ${data}" >안녕하세요. 손님</p>
</body>
</html>

```

아래와 같이 뜬다!!

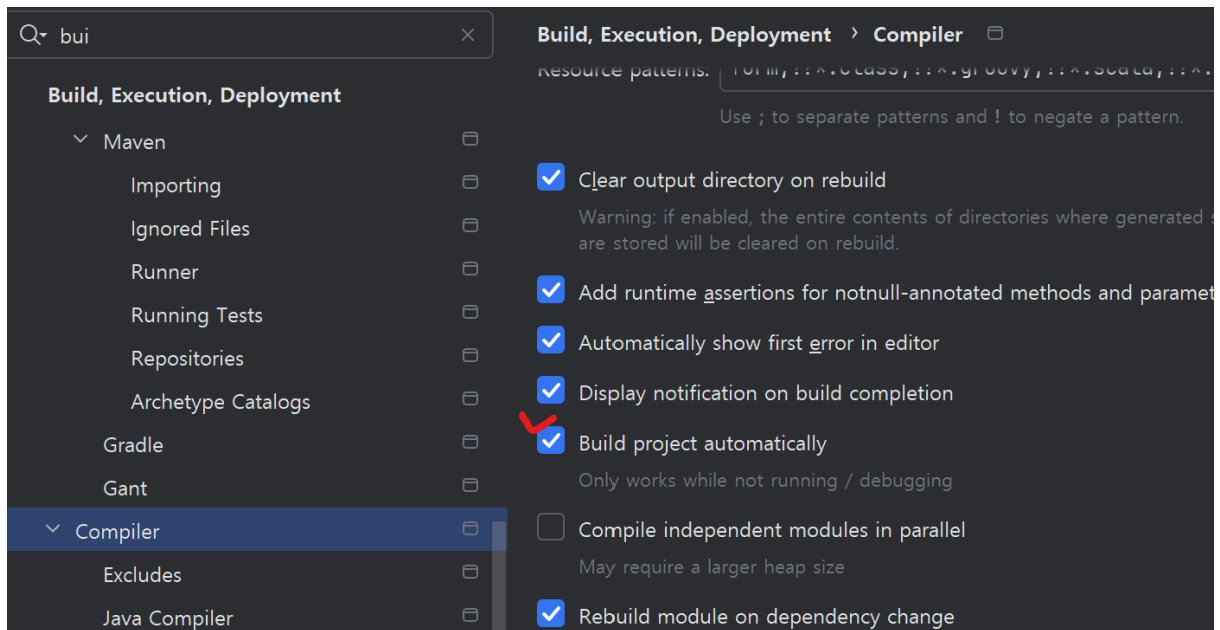


spring-boot-devtools 라이브러리 추가하면, 'html'파일을 컴파일만 해주면 서버 재시작 없이 View파일 변경이 가능하다.

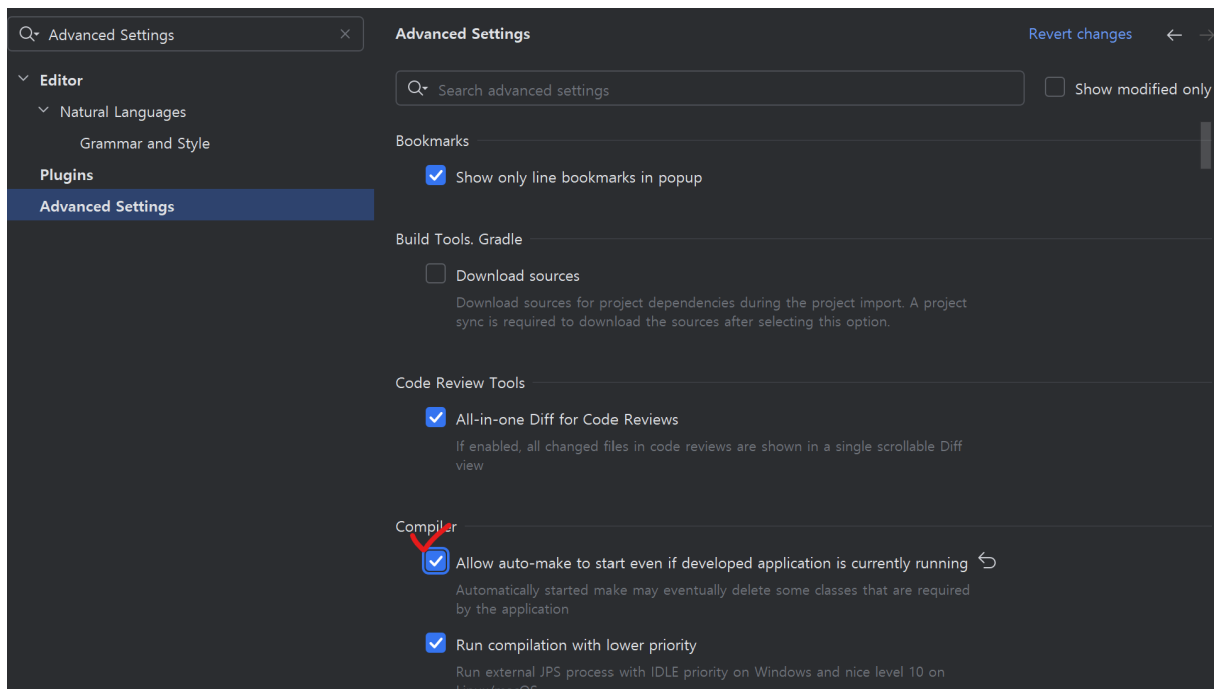
1. dependencies 추가.

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-devtools'  
}
```

2. Build project automatically 체크



3. Allow auto-make to start ~ currently running 체크



4. Gradle 업데이트

5. html수정하고 서버 재시작 안해도 새로고침 되는거 확인하면 끝!