

4. Simple Web Application Development using Spring Boot

A decorative graphic consisting of a series of nested, slightly offset rectangles in a light gray color, creating a sense of depth and movement. The rectangles are arranged in a grid-like pattern, with each subsequent rectangle being slightly larger and shifted to the right and down from the previous one.

Spring-boot의 view 지원

- spring boot 에서 Thymeleaf, FreeMarker, Mustache, Groovy Templates ,Velocity 를 view template 으로 제공한다.

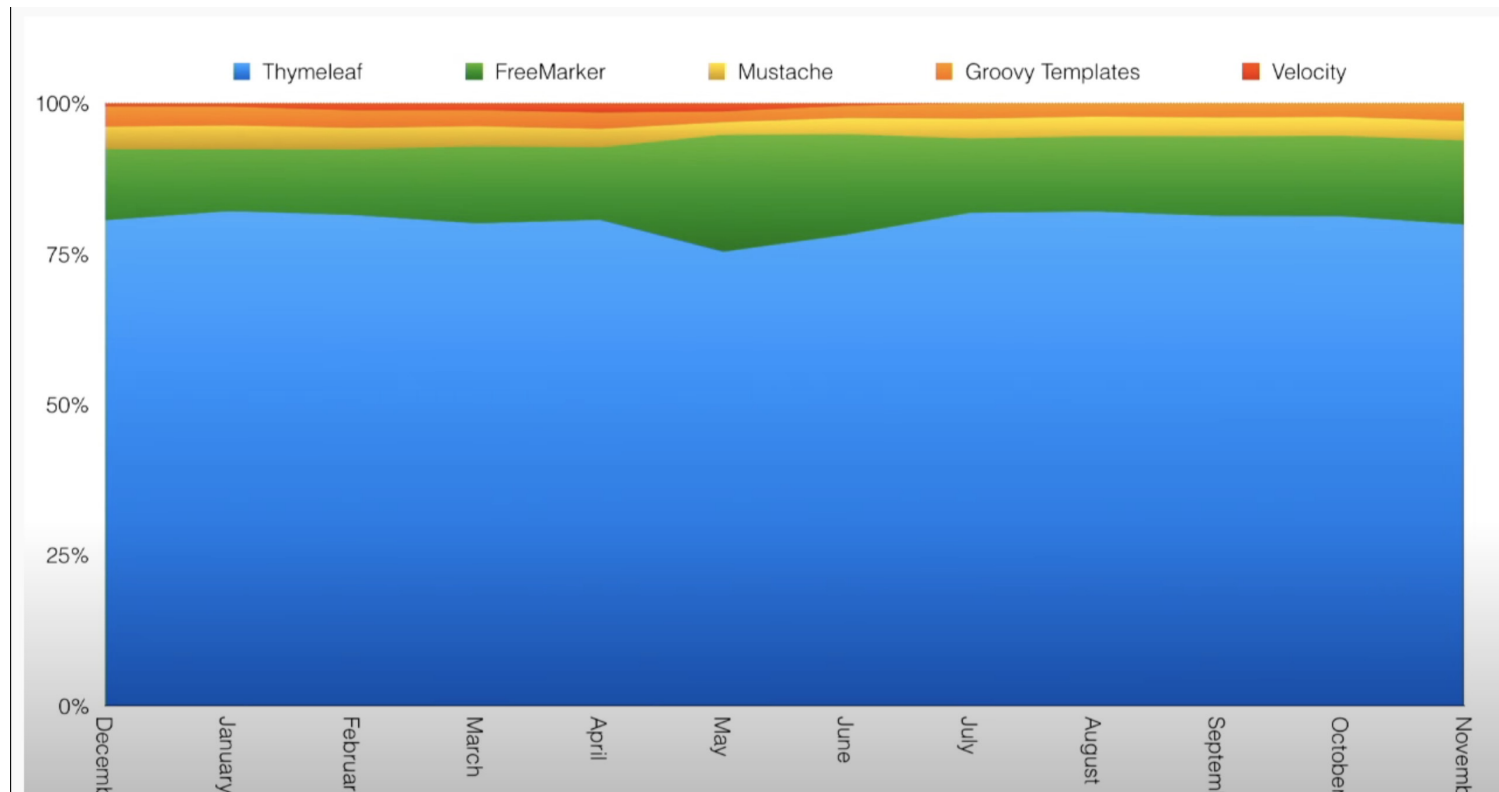
Thymeleaf

- Spring Boot는 Thymeleaf를 기본 지원하여 간단한 설정으로 사용 가능
- html 문법 내에서 view 를 구현할 수 있는 장점

Spring-Boot ㄹ View (Thymeleaf)

Thymeleaf

- Phil Webb (SpringOne 2017)
- <https://youtu.be/MQamx7-bCVI?t=377>



Thymeleaf의 사용

- spring-boot-starter-thymeleaf 의존성 추가

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-thymeleaf</artifactId>  
</dependency>
```

Thymeleaf의 사용

· 프로젝트 루트의 /src/main/resource 경로에 templates 디렉토리를 생성하고 템플릿 작성

Ex) student.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head> <meta charset="utf-8"> <title>welcome</title>
</head>
<body>
  <h1>Name</h1>
  <h2 th:text="${student.name}"></h2>
  <h1>Score</h1>
  <h2 th:text="${student.score}"></h2>
</body>
</html>
```

Thymeleaf의 사용

- view template 을 사용하려면 @Controller를 사용
- /src/main/java 경로의 com.nhnent.edu.springboot 패키지에 StudentWebController 작성
- getStudent 가 반환하는 student 는 view template의 경로 중 일부

```
@Controller
public class StudentWebController {
    @GetMapping("/web/students/{id}")
    public String getStudent(@PathVariable Long id,
                             Model model){
        model.addAttribute("student", new Student("zbum", 100));
        return "student";
    }
}
```

Thymeleaf의 사용

- view template 의 경로변경
- 만약, application.properties 에 아래와 같이 설정되어 있고 getStudent 메서드가 "student"를 반환한다면 템플릿은 "/src/main/resources/templates/main/student.html" 에 위치해야 합니다.

```
spring.thymeleaf.enabled=true  
spring.thymeleaf.prefix=classpath:/templates/main/  
spring.thymeleaf.suffix=.html
```

[과제] thymeleaf 으로 html 렌더링

목표

- Account 시스템의 /account/{id} URL 을 웹브라우저에서 렌더링 하도록 thymeleaf 적용

예상시간

- 10분

JSP의 사용

- spring-boot 에서 JSP를 사용할 수 있지만 권장하지 않습니다.
- legacy의 코드를 재사용하기 위해 학습 필요

Spring-Boot에서 JSP 제약

- war 패키징 된 경우에만 사용이 가능 (실행형 war 또는 tomcat에서 동작)
- 실행형 jar에서는 동작하지 않음
- undertow 는 JSP를 지원하지 않음

패키징 변경 및 라이브러리 의존성 추가

- jar 에는 WEB-INF 를 포함하지 않기 때문에 war로 패키징한다.
- JSTL지원 및 JSP 컴파일을 위해서 두개의 라이브러리를 추가한다.

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

패키징 변경 및 라이브러리 의존성 추가

```
<dependencies>
...
<!-- JSTL for JSP -->
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
</dependency>

<!-- Need this to compile JSP -->
<dependency>
  <groupId>org.apache.tomcat.embed</groupId>
  <artifactId>tomcat-embed-jasper</artifactId>
  <scope>provided</scope>
</dependency>
</dependencies>
```

main 클래스 변경

- war로 패키징 한 경우, main 클래스가 SpringBootServletInitializer을 상속받도록 수정

```
@SpringBootApplication
public class StudentApplication extends SpringBootServletInitializer {

    public static void main(String[] args) {
        SpringApplication.run(StudentApplication.class);
    }

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder builder) {
        return builder.sources(StudentApplication.class);
    }
}
```

JSP 파일 작성

- 프로젝트에 /src/main/webapp/WEB-INF/jsp 디렉토리를 생성하고 jsp file 작성

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="utf-8">
  <title>welcome</title>
</head>
<body>
  <h1>JSP Sample</h1>
  <h2>${message}</h2>
</body>
</html>
```

Controller 클래스 생성

- /src/main/java 경로의 com.nhn.edu.springboot 패키지에 Controller 클래스를 작성
- 화면에 표시할 attribute 는 Map 객체를 사용

```
@Controller
public class WelcomeController {
    @GetMapping("/welcome")
    public String welcome(Map model) {
        model.put("message", "Welcome to the world!");
        return "welcome";
    }
}
```

JSP의 경로 변경

- 만약, application.properties에 아래와 같이 설정되어 있고 welcome 메서드가 "welcome"를 반환한다면 jsp 파일은 "/src/main/webapp/WEB-INF/jsp/welcome.jsp" 에 위치해야 합니다.

```
spring.mvc.view.prefix=WEB-INF/jsp/  
spring.mvc.view.suffix=.jsp
```

JSP 를 사용하는 Spring-Boot 애플리케이션의 실행

- Executable War를 이용한 실행

```
$ mvn clean package  
$ java -jar target/student-1.0.0-SNAPSHOT.war
```

- spring-boot maven plugin 을 이용한 실행

```
$ mvn spring-boot:run
```

- tomcat 등 Web application Server 에 deploy

Q&A



감사합니다