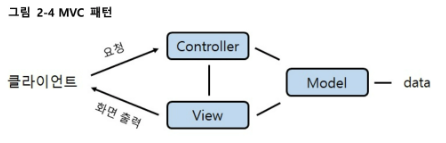
뷰 템플릿이란?

* 계정마다 각기다른 정보로 인해 웹페이지를 하나하나 다 만들 수 없기에 틀을 만들어 두는 작업을 의미
* 웹페이지를 하나의 틀로 만들고 여기에 변수를 삽입하여 서로다른 페이지를 보여주는 형식
* Mustache가 뷰템플릿을 만드는 도구이다

MVC 기법



* Controller : 클라이언트의 요청에 따라 서버에서 처리하는 역할
* Model : 데이터를 관리하는 역할
* View : 웹페이지를 화면에 보여줌

뷰 템플릿은 templates 디렉토리에서 생성

Tip. Doc 작성후 탭키 누르면 기본 html코드가 작성

뷰템플릿을 적용하기 위해서는 컨트롤러를 제작하는 것이 필수

@Controller 를 통해 어노테이션을 입력

어노테이션은 소스코드에 추가하여 사용하는 메타 데이터의 일종. 메타데이터는 프로그램에서 처리해야할 데이터가 아니라 컴파일 및 실행 과정에서 코드를 어떻게 처리해야 할지 알려주는 추가 정보이다. @를 사용하여 나타냄

Controller에서 리턴값은 mustache확장자는 뺀 파일 이름만을 넘겨주면 된다.

GetMapping(“/hi”)

* 페이지를 반환해달라고 하는 url 요청을 도와주는 어노테이션
* 이러면 localhost8080/hi 를 접속시 이 홈페이지에 들어가게 된다.

여기서 한글 깨짐 현상이 발생할 시 application.properties 파일안에 다음과 같은 코드 작성

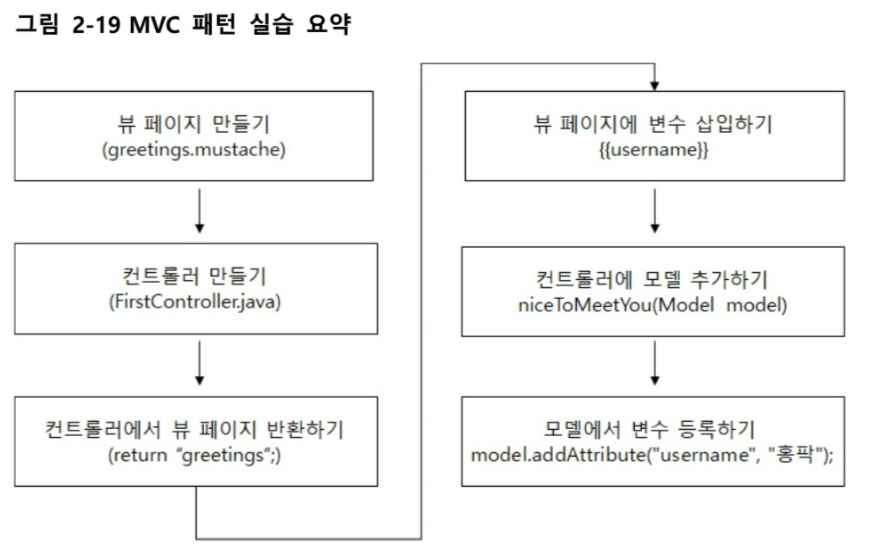
server.servlet.encoding.force=true

머스테치 문법 사용하기

- {{변수명}} -> 변수사용하기

import는 알트 + 엔터를 통해 자동완성이 가능하다

MVC 패턴 요약



1분퀴즈

- !뷰템플릿 이란 웹페이지를 일종의 틀로 만든 것이다.

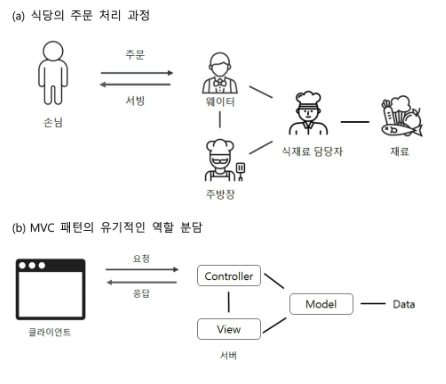
- !컨트롤러 는 클라이언트의 요청을 받아 서버에서 처리하는 역할을 한다

- !모델 은 뷰 템플릿에서 사용되는 데이터를 관리하는 역할을 한다.

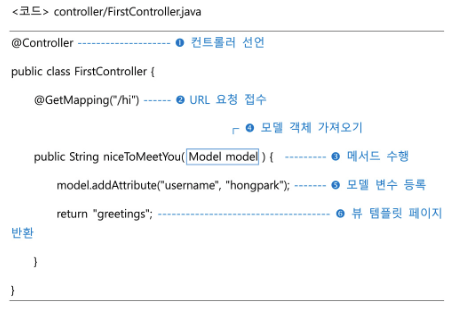
- !@Controller 는 이 클래스가 컨트롤러임을 선언한다.

- !GetMapping 은 클라이언트가 url 요청을 받아 특정 컨트롤러의 메서드가 처리하게 한다1

MVC의 역할과 실행흐름 이해하기



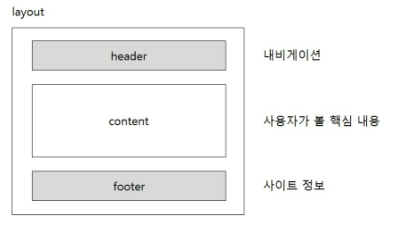
FirstController.java 파일의 구동 원리



1분퀴즈

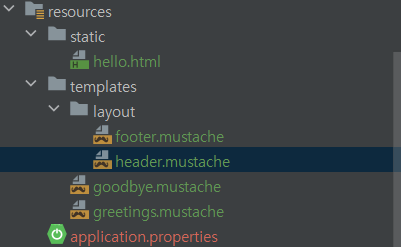
- 컨트롤러의 메서드 반환값은 뷰 템플릿 페이지의 확장자를 포함해 작성한다. -> 확장자를 포함하지 않는다!

뷰 템플릿 페이지에 레이아웃 적용하기  
- 헤더 푸터 레이아웃



템플릿화 하기! 중요

긴웹코드들을 템플릿화하여 깔끔히 정리하기



이런식으로 템플릿안에 레이아웃 디렉토리를 만들어 템플릿을 작성

코드를 일정부분으로 나누어 이를 header.mustache, footer.mustache에 넣기

이후, 원본 템플릿인 greetings.mustache에는 {{>layout/header}}, {{>layout/footer}}로 변수를 설정이 가능! 파일경로를 의미한다

1분퀴즈

- !레이아웃 이란 화면에 요소를 배치하는 것을 의미한다. 웹페이지는 같은 요소를 가지고도 어떻게 배치하느냐에 따라 다른 느낌을 줄 수 있다.

- !부트스트랩 은 웹페이지를 쉽게 만들 수 있도록 도움을 준다 코드 모음짐

- !템플릿화 는 코드를 하나의 틀로 만들어 변수화시키는 것을 의미

정리

1) 뷰 템플릿 : 웹 페이지를 하나의 틀로 만들고, 여기에 변수를 삽입하여 서로 다른 페이지로 보여주는 기술

2) MVC 패턴 : 웹 페이지를 화면에 보여주고(View), 클라이언트의 요청을 받아 처리하고(Controller), 데이터를 관리하는(Model) 역할을 영역별로 나누어 하는 기법을 의미

3) 뷰 템플릿 생성 위치 : 뷰 템플릿은 src > main > resources > templates 디렉토리에서 생성

머스테치 확장자는 mustache

4) 컨트롤러 생성 위치 : 컨트롤러는 src > main > java > 기본 패키지안에 컨트롤러 패키지를 제작 후 자바 클래스 파일을 생성

5) 모델을 통해 변수 등록하는 방법 : 모델은 컨트롤러의 메소드에서 매개변수로 받아온다. 모델에서 변수를 등록할때는 addAttribute() 메소드를 사용한다.

형식 : model.addAttribute(”변수명”, “변수값”);

6) 헤더 - 푸터 레이아웃 : 가장 기본이 되는 레이아웃을 의미. 헤더 영역에는 사이트 안내를 위한 네비게이션, 푸터 영역에는 사이트 정보를 입력. 두 영역 사이의 컨텐츠는 사용자가 볼 핵심 내용을 배치

7) 레이아웃 템플릿 생성과 적용

- 레이아웃 템플릿 파일은 src > main > resources > templates 디렉토리에 만들지만, 이번 실습에서는 파일관리를 편하게 하기 위해 layous 디렉토리를 추가하여 그 안에 넣음

- 레이아웃 템플릿을 삽입할때는 원하는 위치에 {{>파일경로/파일명}} 형식으로 작성하기!