진영이의 스프링 웹 개발기초

내용

[컨텐츠에 따른 분류 2](#_Toc54868740)

[MVC 자세히 알아보기 3](#_Toc54868741)

[Annotation 3](#_Toc54868742)

[1. @Controller 3](#_Toc54868743)

[2. @GetMapping(String) 3](#_Toc54868744)

[3. @RequestParam(String) 4](#_Toc54868745)

[Spring 중요개념 5](#_Toc54868746)

[1. 제어의 역행과 컨테이너 5](#_Toc54868747)

[2. 스프링 설정파일 XML 5](#_Toc54868748)

[3. 의존성 관리 5](#_Toc54868749)

[4. 비즈니스 컴포넌트 구조 5](#_Toc54868750)

[5. AOP(Aspect Oriented Programming) 5](#_Toc54868751)

[패턴 6](#_Toc54868752)

[1. Factory 패턴 6](#_Toc54868753)

[2. Singleton 패턴 6](#_Toc54868754)

# 컨텐츠에 따른 분류

1. 정적 컨텐츠

변화가 없는 콘텐츠

HTML, CSS, JS와 같이 미리 서버에 저장해두고 서버가 요청을 받으면 그저 응답만 해주면 되는 것들로 구성되어 있습니다.

2. MVC와 템플릿 엔진

템플릿 엔진

서버에서 프로그래밍 후 HTML을 동적으로 만들어서 제공하는 것

Ex) JSP, PHP

MVC

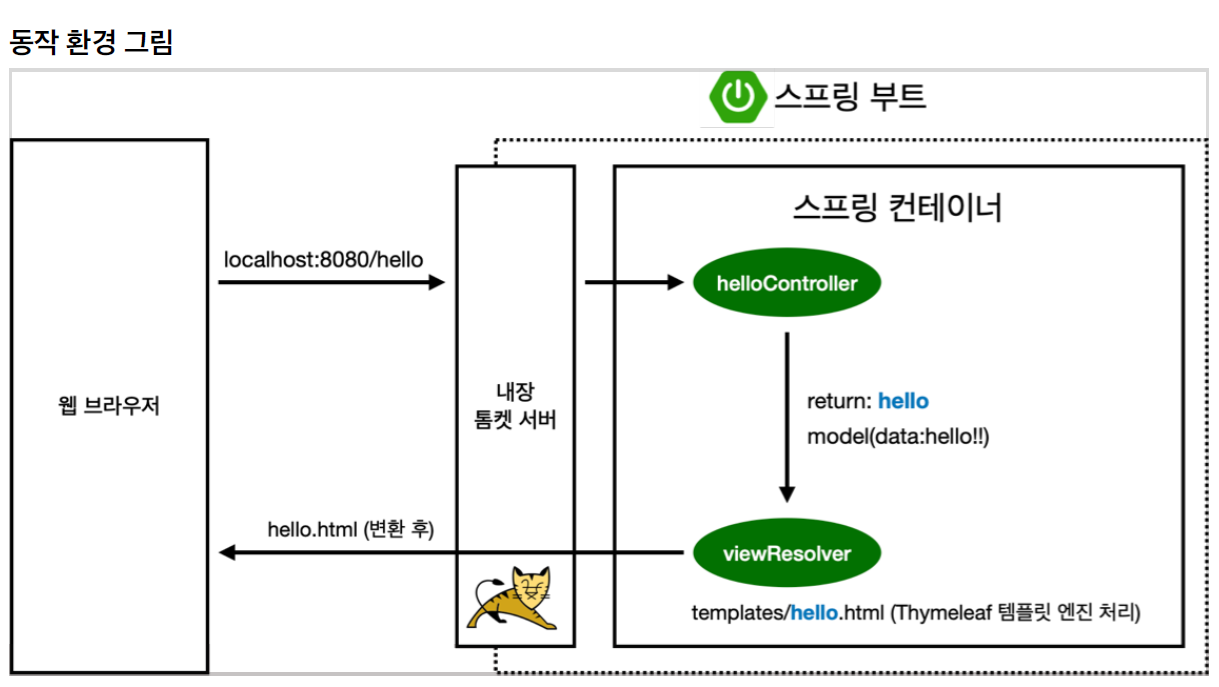
모델, 뷰, 컨트롤러 ----> 요즘 인기 많음.

3. API

JSON 데이터 구조포맷으로 클라이언트에게 데이터 전달하는 것

뷰나 리액트, 서버 간 통신 등에서도 API를 사용하기도 함.

동작환경 그림으로 살펴보기

1. 웹 브라우저 -> 서버(웹페이지 요청)
2. 톰캣 서버에서 ‘hello’ 컨트롤러 검사
3. 스프링은 hello 컨트롤러가 존재함
4. 스프링은 컨트롤러를 반환 ---> 웹페이지가 나타나게 됨
5. 만약에 hello 컨트롤러가 Spring에 없다면????
6. static 폴더에서 파일을 찾아보고 있다면 반환!   
   없다면 Error

# MVC 자세히 알아보기

과거에는 Model, View, Controller을 따로 나누지 않았다. ----> Model1 방식

MVC ----> Model, View, Controller, : 요즘은 이게 기본이다.

View : 화면을 그리는데 집중

Model, Controller : 비즈니스 로직이나 내부적 처리에 집중

# Annotation

## @Controller

클래스 단위로 사용하는 어노테이션.

Controller 어노테이션이 붙은 클래스는 웹 요청을 처리하는 컨트롤러로 사용가능하다.

## @GetMapping(String)

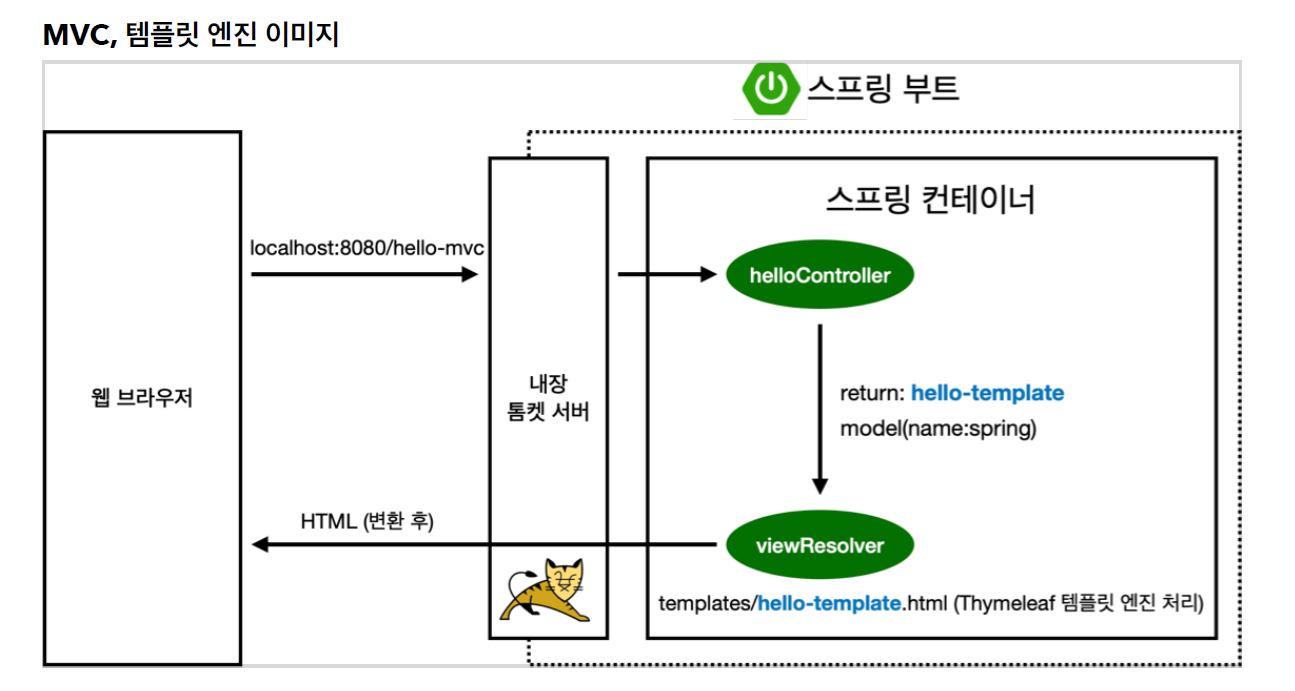
메소드 수준으로 사용하는 어노테이션

GetMapping을 통해 넘어온 String 매개변수는 클라이언트가 요청한 주소

이 노테이션이 붙은 메서드는 클라이언트가 해당 주소로 접근했을 경우 실행된다.

이 어노테이션이 붙은 메서드는 String값을 반환하며, 컴퓨터는 반환된 String값과 이름이 같은 파일을 templates 폴더 안에서 찾아 HTML파일로 변환 후 클라이언트에게 반환해준다.

Ex) @GetMaaping(“hello”)라는 어노테이션이 붙은 메서드의 경우

IP주소:포트번호/hello로 접속할 경우, 실행된다.

## @RequestParam(String)

@GetMapping 어노테이션이 붙은 메서드의 매개변수에 붙이는 어노테이션

해당매개변수에 웹페이지로부터 전달된 파라미터 값을 대입한다.

Ex) public String hello(@RequestParam(“name”) name) { }

위와 같은 메서드의 경우 웹페이지를 통해 전달된 name값이 메서드의 매개변수 name에 대입된다.

# Spring 중요개념

## 제어의 역행과 컨테이너

컨테이너는 특정 객체의 생성과 관리를 담당하며 객체 운용에 필요한 기능 제공

특정 객체 = XML파일에 등록된 <bean>객체

컨테이너는 자신이 관리할 클래스들이 등록된 XML설정 파일을 로딩하여 구동

### bean 객체는 기본적으로 컨테이너가 구동되는 시점에 생성됨.

컨테이너는 애플리케이션 운용에 필요한 객체를 생성하고 객체 간의 의존관계 관리

컨테이너가 객체생성 및 객체들 사이의 의존관계를 처리하는 것 = 제어의 역행

\* 결합도가 높은 프로그래밍은 유지보수가 어렵다.   
이러한 결합도를 낮추기 위한 디자인 패턴 = Factory 패턴

## 스프링 설정파일 XML

아래 태그들이 스프링 설정파일에서 무슨 역할을 하는지 알아보기~~~

beans 네임스페이스

<bean> : 클래스를 등록하면, 스프링컨테이너에서 객체 관리

### 기본적으로 싱글톤 패턴 방식으로 객체 관리

<import> : XML파일을 기능별로 작성한 뒤 하나로 통합하기 위하여 사용

어노테이션 기반 설정

### xml파일의 과도한 설정에 대한 부담이 큼 🡺 때문에 어노테이션을 이용하여 설정

### 어노테이션 기반 설정을 위해서는 xml파일에 context네임스페이스 추가 및 component-scan 설정해야 함

## 의존성 관리

스프링의 IoC 지원 방법

### Dependency Lookup : 컨테이너가 생성한 객체를 클라이언트가 검색하여 사용

### Dependency Injection : 객체들 사이의 의존관계를 컨테이너가 직접 처리 ====> 세터인젝션과 생성자 인젝션으로 나뉨

## 비즈니스 컴포넌트 구조

### DAO - DB와의 연동을 책임지는 녀석이다

### VO - 데이터 전달을 목적으로 사용하는 객체 🡺 DB테이블구조를 확인한 후 작성Service - 서비스 로직 구현 뭔지 모르겠다.

## AOP(Aspect Oriented Programming)

핵심관심 🡺 사용자의 요청에 따라 수행되는 핵심 비즈니스 로직

횡단관심 🡺 메소드마다 공통으로 등장하는 로깅이나 예외, 트랜잭션 처리와 같은 코드

AOP = 횡단관심을 효율적으로 관리하기 위한 기법,   
AOP 관련 라이브러리 - aspectj AOP 관련 네임스페이스 – aop

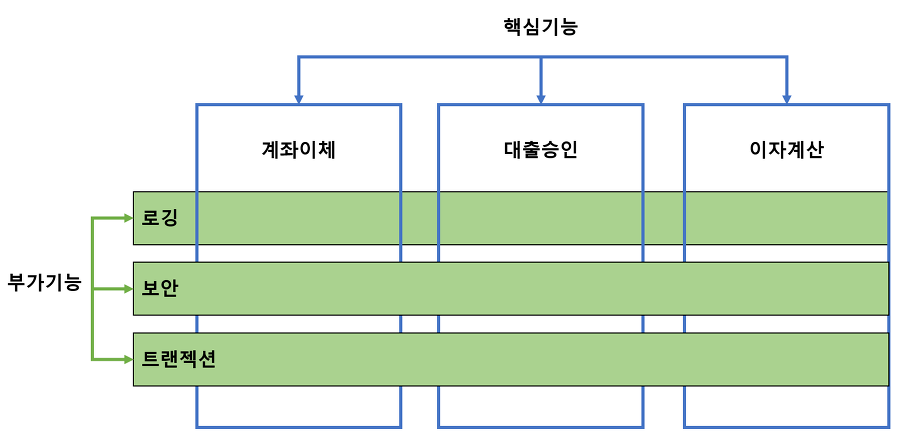


그림 횡단관심과 핵심관심

AOP용어 정리

### 조인포인트 🡺 클라이언트가 호출하는 모든 비즈니스 메서드

### 포인트컷 🡺 필터링된 조인포인트

### 포인트컷 표현식 : 리턴타입 패키지경로 클래스명 메소드명 및 매개변수

### ex) \*

# 패턴

## Factory 패턴

Factoroy 클래스에서 객체 생성을 관리한다.

ex) CarFactory 클래스에서 Bus, Taxi, Truck 등의 클래스 객체 생성

## Singleton 패턴

Singleton패턴을 적용한 클래스의 경우, 그 클래스는 오로지 하나의 인스턴스만을 생성.  
getInstance()와 같은 메서드(혹은 get클래스명())를 이용하여 모든 클라이언트에게   
동일한 인스턴스 반환