



## Special장. 스프링 필수사항

# 0. Web의 기반

## HTTP

Hypertext Transfer **Protocol**

어플리케이션 컨트롤

GET  
POST  
PUT  
DELETE  
OPTIONS  
HEAD  
TRACE  
CONNECT

# 0. Web의 기반



## URI

Uniform **Resource** Identifier

리소스 식별자

특정 사이트  
특정 쇼핑 목록  
동영상 목록

모든 정보에 접근 할 수 있는 정보

# 0. Web의 기반

## HTML

Hyper Text Markup **Language**

하이퍼미디어 포맷

XML을 바탕으로한  
범용 문서 포맷

이를 이용하여  
Chrome,  
Safari,  
Explorer에서

사용자가 알아보기 쉬운 형태로 표현

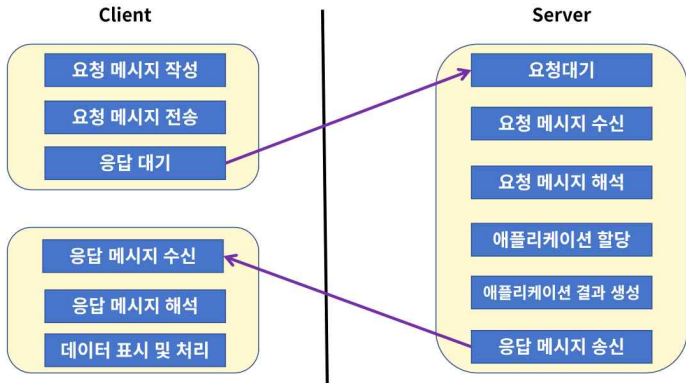


# 1. HTTP Protocol



- HTTP (Hyper Text Transfer Protocol) 로 RFC 2616에서 규정된 Web에서 데이터를 주고 받는 프로토콜.
- 이름에는 하이퍼텍스트 전송용 프로토콜로 정의되어 있지만 실제로는 HTML, XML, JSON, Image, Voice, Video, Javascript, PDF 등 다양한 컴퓨터에서 다룰 수 있는 것은 모두 전송할 수 있습니다.
- HTTP는 TCP를 기반으로 한 REST의 특징을 모두 구현하고 있는 Web기반의 프로토콜

HTTP는 메시지를 주고 (Request) 받는 (Response) 형태의 통신 방법입니다.



# 1. HTTP Protocol

## HTTP Status Code

응답의 상태를 나타내는 코드

	의미	내용
1XX	처리중	처리가 계속 되고 있는 상태. 클라이언트는 요청을 계속 하거나 서버의 지시에 따라서 재요청
2XX	성공	요청의 성공
3XX	리다이렉트	다른 리소스로 리다이렉트. 해당 코드를 받았을 때는 Response의 새로운 주소로 다시 요청
4XX	클라이언트 에러	클라이언트의 요청에 에러가 있는 상태. 재전송 하여도 에러가 해결되지 않는다.
5XX	서버에러	서버 처리중 에러가 발생한 상태. 재 전송시 에러가 해결 되었을 수도 있다.



# 1. HTTP Protocol

## 자주 사용되는 코드

200	성공
201	성공. 리소스를 생성 성공
301	리다이렉트, 리소스가 다른 장소로 변경됨을 알림
303	리다이렉트, Client에서 자동으로 새로운 리소스로 요청 처리
400	요청 오류, 파라미터 에러
401	권한 없음 ( 인증실패 )
404	리소스 없음 ( 페이지를 찾을 수 없음 )
500	서버 내부 에러 ( 서버 동작 처리 에러 )
503	서비스 정지 ( 점검 등등)



## 2. (웹기반의) REST

- Representational State Transfer의 약자로, 하나의 URI가 고유한 리소스를 처리하는 공통 방식
- REST 방식으로 제공되는 API를 REST API(또는 RESTful API)라고 하며, 이는 트위터와 같은 Open API에서 많이 사용됨

## 2. REST 방식으로 URI 표현하기

- REST 방식은 서버에 데이터 조회뿐만 아니라 추가, 수정, 삭제 작업 요청 시 HTTP 메서드를 이용함 (리소스 접근시 URI를 사용함)

`https://foo.co.kr/user/100`

Resource : user

식별자 : 100

## 2. REST 방식으로 URI 표현하기

HTTP 메서드의 기능 : HTTP의 요청을 특정하는 Method  
REST를 구현하기 위한 인터페이스

메서드	설명
POST	추가(Create)
GET	조회(Select)
PUT	수정(Update)
DELETE	삭제>Delete)

## 2. REST 방식으로 URI 표현하기

### REST로 게시판 기능 관련 URI 만들기

메서드	URI	설명
POST	/boards + 데이터	새 글 등록하기
GET	/boards/133	133번 글 조회하기
PUT	/boards/133 + 데이터	133번 글 수정하기
DELETE	/boards/133	133번 글 삭제하기



### 3. URI 설계

- 브라우저에서 웹 애플리케이션 요청

http://IP주소:포트번호/컨텍스트이름/요청파일이름

http://127.0.0.1:8080/webStore/main.do

프로젝트 이름 = 컨텍스트 이름



# 3. URI 설계

- 브라우저에서 웹 애플리케이션 요청

URI (Uniform Resource Identifier)

URL (Uniform Resource Locator)

URL은 URI 의 하위 개념

# 3. URI 설계

## URI 설계 원칙 ( RFC-3986 )

- 슬래시 구분자 ( / )는 계층 관계를 나타내는 데 사용한다.
- URI 마지막 문자로 ( / ) 는 포함하지 않는다.
- 하이픈(-)은 URI 가독성 을 높이는데 사용한다
- 밑줄(\_)은 사용하지 않는다.

# 3. URI 설계

## URI 설계 원칙 ( RFC-3986 )

- URI 경로에는 소문자가 적합하다.
- 파일 확장자는 URI에 포함하지 않는다.
- 프로그래밍 언어에 의존적인 확장자를 사용하지 않는다.
- 구현에 의존적인 경로를 사용하지 않는다.

# 3. URI 설계

## URI 설계 원칙 ( RFC-3986 )

- 세션 ID를 포함하지 않는다.
- 프로그래밍 언어의 Method명을 이용하지 않는다.
- 명사에 단수형 보다는 복수형을 사용해야 한다. 컬렉션에 대한 표현은 복수로 사용
- 컨트롤러 이름으로는 동사나 동사구 를 사용한다.

# 3. URI 설계

## URI 설계 원칙 ( RFC-3986 )

- 경로 부분 중 변하는 부분은 유일한 값으로 대체 한다.
- CRUD 기능을 나타내는 것은 URI에 사용하지 않는다.



## 4. IOC / DI

### IoC (Inversion of Control)

스프링에서는 일반적인 Java 객체를 new로 생성하여 개발자가 관리 하는 것이 아닌

Spring Container 에 모두 맡긴다.

즉, 개발자에서 -> 프레임워크로 **제어**의 객체 관리의 **권한이**

**넘어 갔음** 으로 “**제어의 역전**” 이라고 합니다.

### DI (Dependency Injection)

## 4. IOC / DI

➤(추가 내용은 강의 자료 참조)





# 5. AOP



AOP ( Aspect Oriented Programming )

관점지향 프로그램

스프링 어플리케이션은 대부분 특별한 경우를 제외 하고는  
Web Layer , Business Layer, Data Layer 로 정의.

## 5. AOP

**Web Layer** : REST API를 제공하며, Client 중심의 로직 적용

**Business Layer** : 내부 정책에 따른 logic를 개발하며, 주로 해당 부분을 개발

**Data Layer** : 데이터 베이스 및 외부와의 연동을 처리

## 주요 Annotation

Annotation	의미
@Aspect	자바에서 널리 사용하는 AOP 프레임워크에 포함되며, AOP를 정의하는 Class에 할당
@Pointcut	기능을 어디에 적용시킬지, 메소드? Annotation? 등 AOP를 적용 시킬 지점을 설정
@Before	메소드 실행하기 이전
@After	메소드가 성공적으로 실행 후, 예외가 발생 되더라도 실행
@AfterReturning	메소드 호출 성공 실행 시 (Not Throws)
@AfterThrowing	메소드 호출 실패 예외 발생 (Throws)
@Around	Before / after 모두 제어

## 5. AOP

➤(추가 내용은 강의 자료 참조)

**감사합니다.**

