

Spring Framework 맛보기|2

EC 23기 양희찬

CONTENTS

01 지난 세미나 복습

Server와 Client
Spring Framework 구조
Spring Boot란?

02 실습 준비

Java List 사용법
HTTP란?
HTTP Method
인자 넘겨주는 법

03 실습 (2)

API 설계하기
도메인 모델
데이터 저장/조회/수정/삭제

04 배운 내용 정리

오늘 배운 내용
다음 실습 내용

01 지난 세미나 복습

#복습 해왔죠? ^o^

01. 지난 세미나 복습

Server

Server

고객(Client)의 요청에 맞는 서비스를 제공하는
컴퓨터 or 소프트웨어

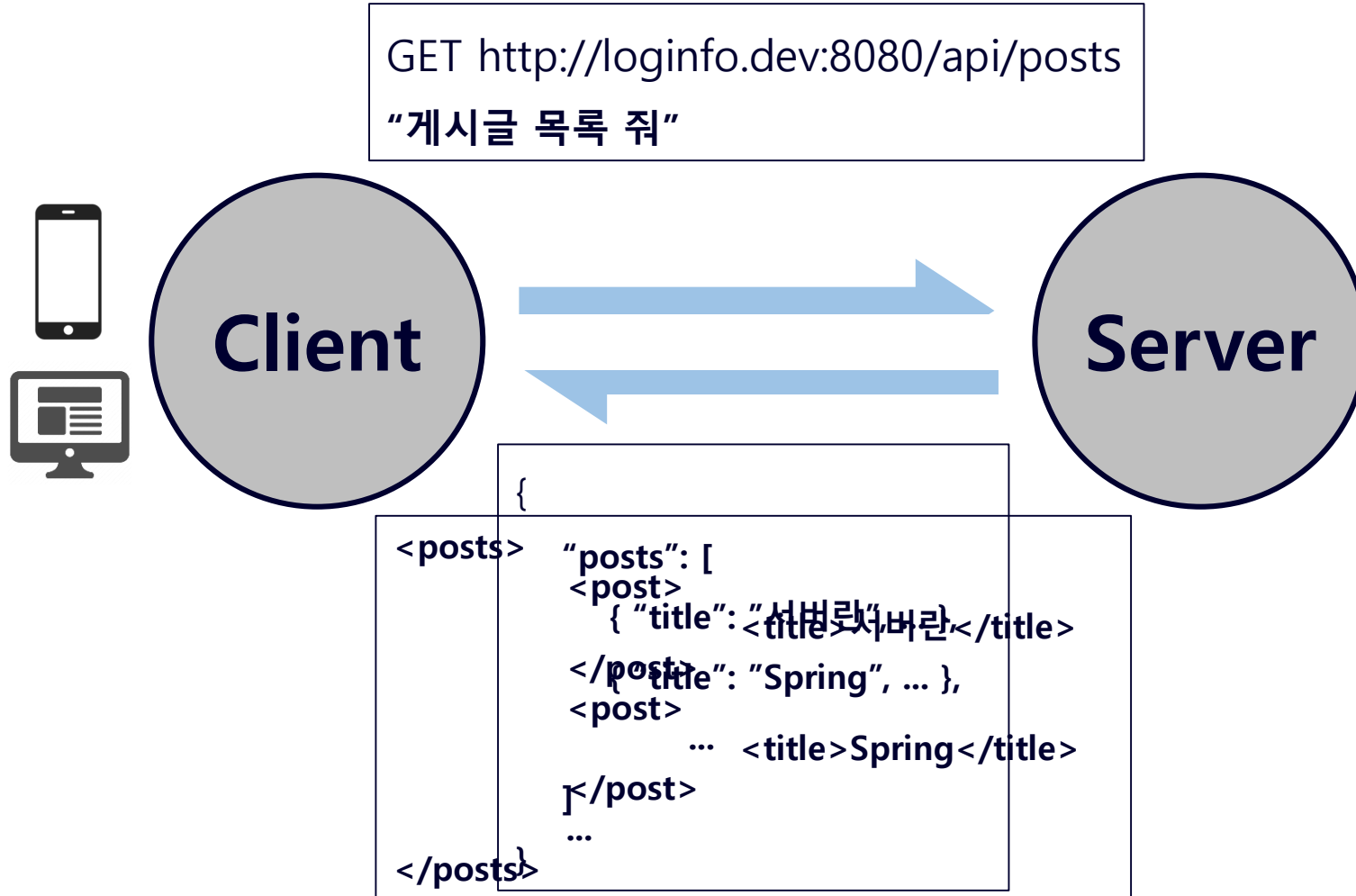
01. 지난 세미나 복습

Server



01. 지난 세미나 복습

Server와 Client



01. 지난 세미나 복습

Server와 Client

Client

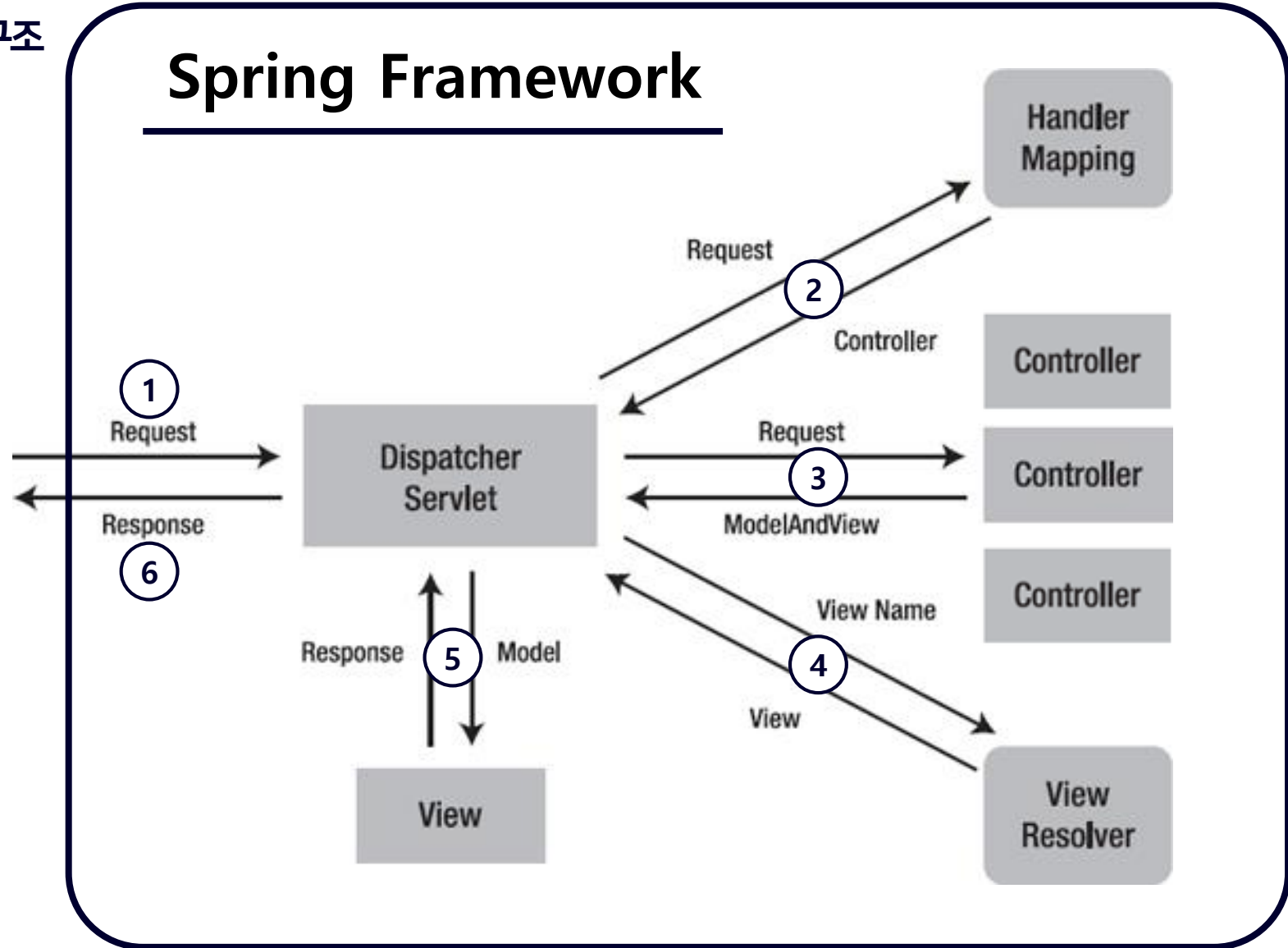
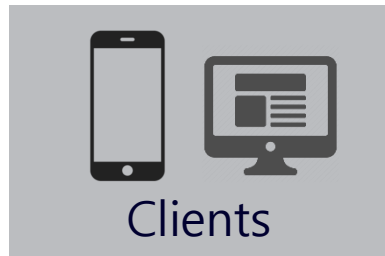
표현계층
Front-End
요청

Server

데이터 접근 계층
Back-End
응답

01. 지난 세미나 복습

Spring Framework 구조



01. 지난 세미나 복습

Spring Boot



초기 설정의 어려움

- 필요 라이브러리 모두 import
- 라이브러리간 dependency 설정
- xml 형식의 설정(Beans 설정 등)
- Tomcat 따로 설치 및 구동 필요

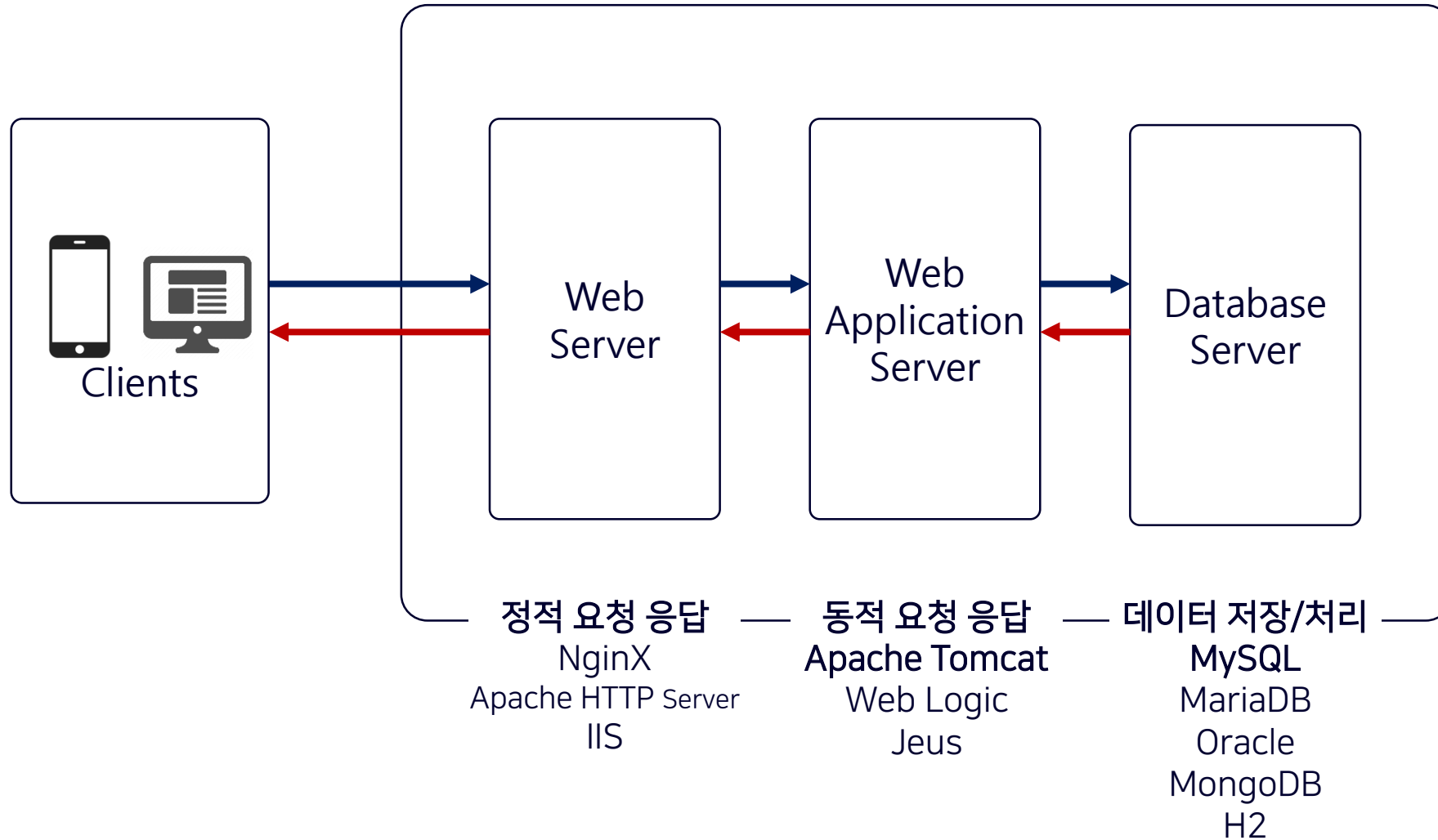


초기 설정 단순화

- 관련 라이브러리를 버전에 맞추어 import할 수 있음
- Java Annotation으로 간단히 Beans 설정
- 내장 Tomcat 사용

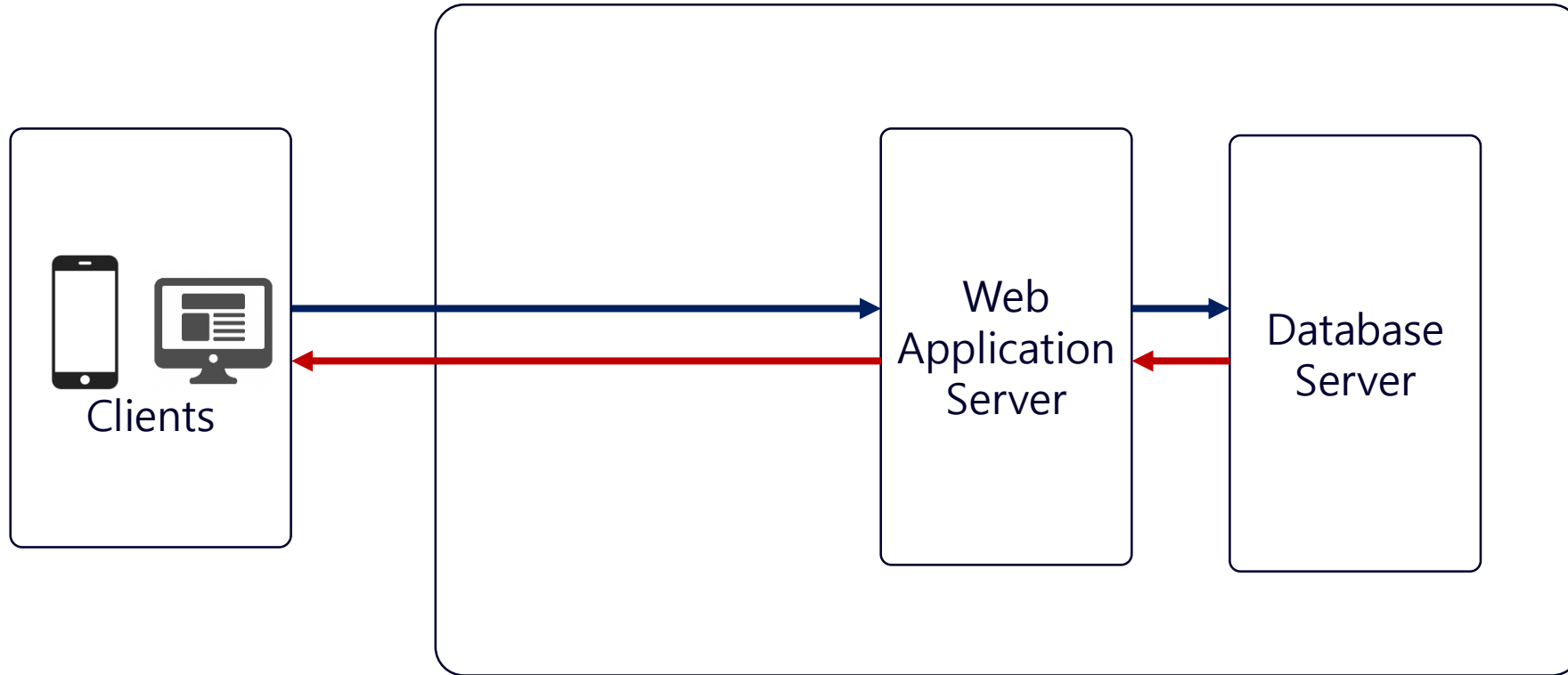
01. 지난 세미나 복습

Server와 Client - 아키텍처



01. 지난 세미나 복습

실습할 아키텍처



01. 지난 세미나 복습

개발 환경

백엔드 개발 프레임워크 : Spring Framework, **Spring Boot**

프로그래밍 언어 : **Java** or Kotlin

빌드툴 : Maven or **Gradle**

Persistence Framework : JPA or **MyBatis**

데이터베이스 : **MySQL**

IDE : **IntelliJ** or Eclipse or STS

DB 구축/운영툴 : **MySQL Workbench**

API 테스트툴 : **Postman** or Insomnia

01. 지난 세미나 복습

프로젝트 생성

Spring Initializr Link

<https://start.spring.io>

01. 지난 세미나 복습

프로젝트 생성

Project

Language

Spring Boot

Project Metadata

Dependencies

Maven Project

Gradle Project

Java

Kotlin

Groovy

2.2.0 M5

2.2.0 (SNAPSHOT)

2.1.8 (SNAPSHOT)

2.1.7

Group

com.endlesscreation

Artifact

spring

> Options

Q

☰

4 selected

Search dependencies to add

Web, Security, JPA, Actuator, Devtools...

Selected dependencies

Spring Web Starter

Build web, including RESTful, applications using Spring MVC.
Uses Apache Tomcat as the default embedded container.

✓

Spring Boot DevTools

Provides fast application restarts, LiveReload, and
configurations for enhanced development experience.

✓

MySQL Driver

MySQL JDBC driver.

✓

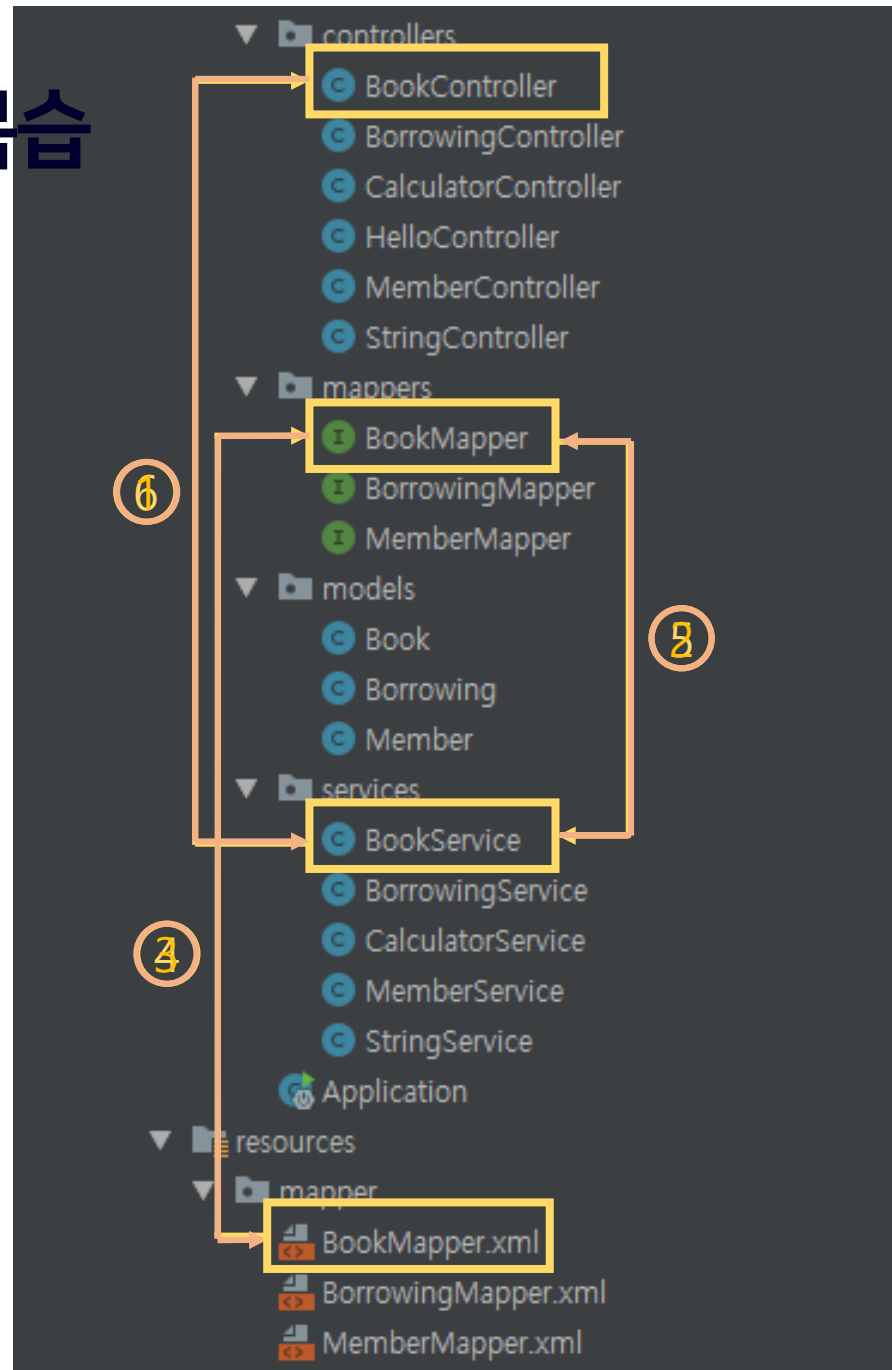
MyBatis Framework

Persistence framework with support for custom SQL, stored
procedures and advanced mappings. MyBatis couples
objects with stored procedures or SQL statements using a
XML descriptor or annotations.

✓

01. 지난 세미나 복습

프로젝트 구조 살펴보기



Controller

- Routing 해주는 역할
- URL과 method를 이어주기만 함
- Service를 호출

Service

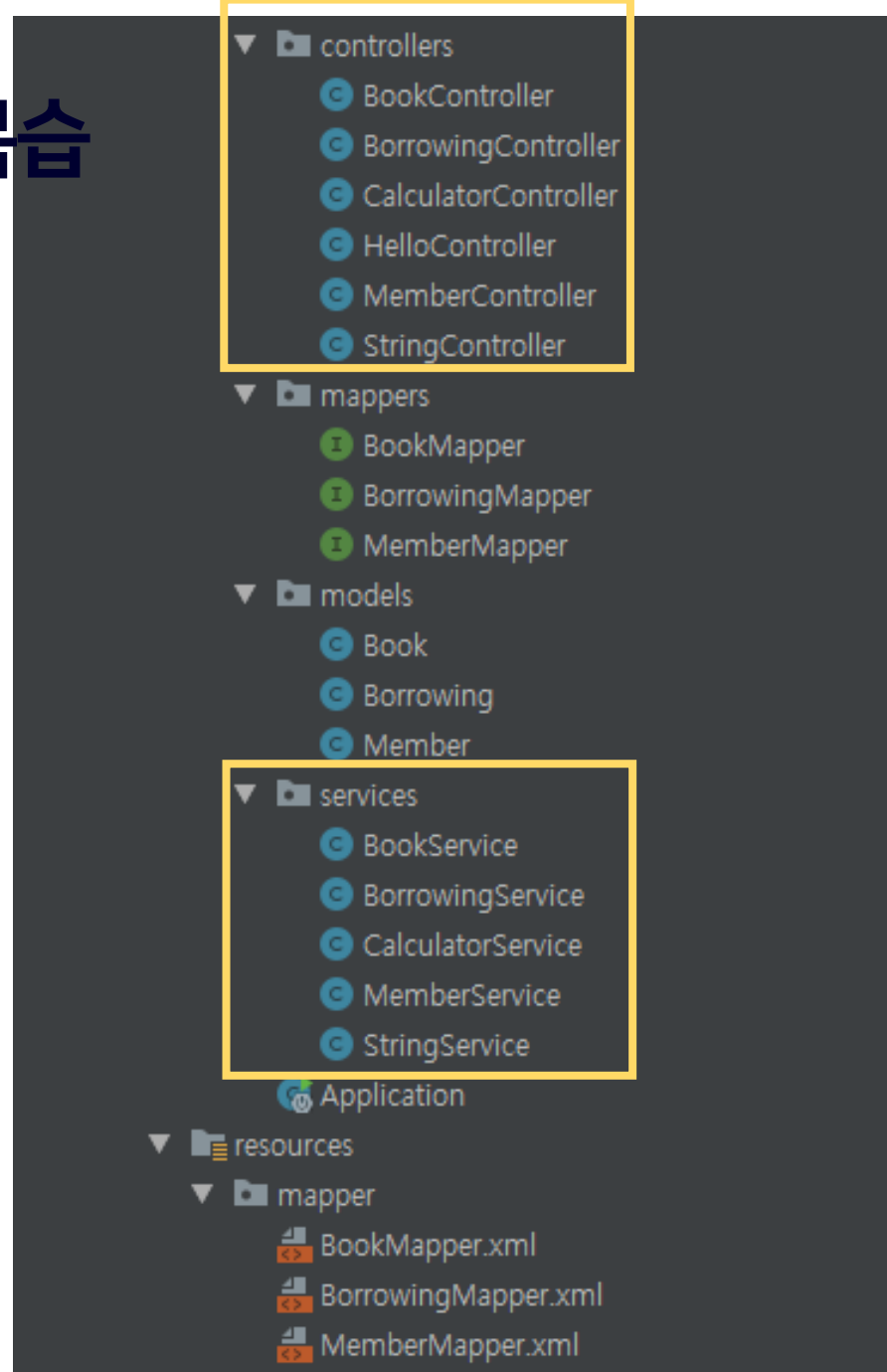
- 비즈니스 로직 수행
- Mapper를 호출

Mapper

- DB에 접근하여 데이터 저장/처리

01. 지난 세미나 복습

프로젝트 구조 살펴보기



Controller

- Routing 해주는 역할
- URL과 method를 이어주기만 함
- Service를 호출

Service

- 비즈니스 로직 수행
- Mapper를 호출

01. 지난 세미나 복습

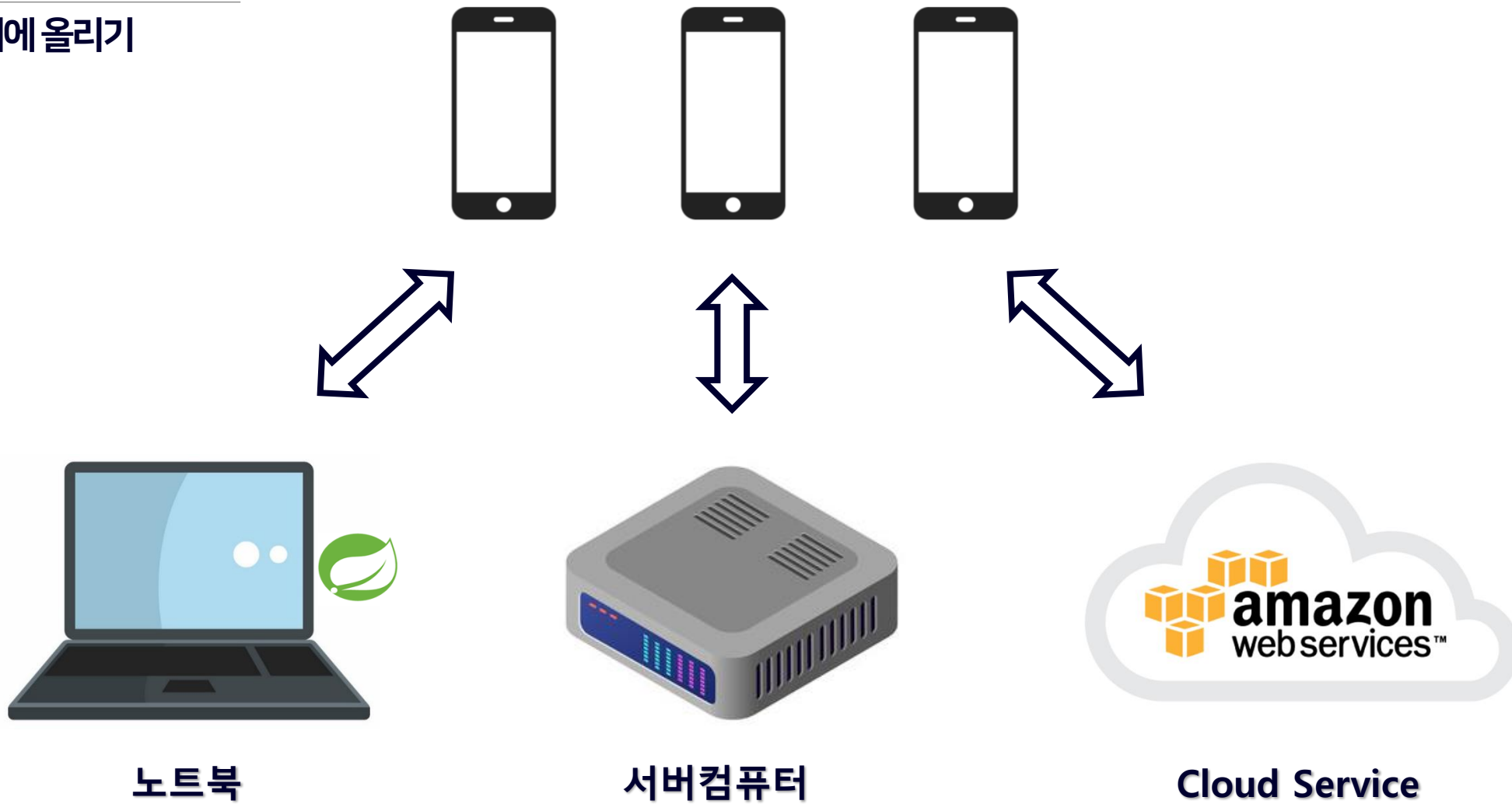
간단한 API 개발

01. 지난 세미나 복습

Controller 나누기

01. 지난 세미나 복습

서버에 올리기



02 실습준비

#List #HTTP #Postman

02. 실습 준비

Java ArrayList

Array를 사용하면서 불편했던 점 [생활코딩 - 배열](#)

- 배열 길이를 초과했을 경우 새로 만든 후 옮겨줘야 함
- 배열 중간에 위치한 데이터를 삭제하면 다른 데이터들을 당겨줘야 함
- 배열 안에 존재하는 데이터의 개수를 일일이 세어야 함

List를 쓴다면? [생활코딩 - 리스트](#)

- 리스트 길이를 초과했을 경우 알아서 추가해 줌
- 리스트 중간에 위치한 데이터 삭제 시 알아서 당겨줌
- 리스트 안에 존재하는 데이터의 개수를 쉽게 구할 수 있음

02. 실습 준비

Java ArrayList

ArrayList 생성

```
ArrayList<Integer> list = new ArrayList<>();
```

ArrayList 데이터 개수

```
list.size();
```

ArrayList 데이터 추가

```
list.add(10);
```

```
list.add(20); list.add(30);
```

ArrayList 삭제

```
list.remove(1);
```

02. 실습 준비

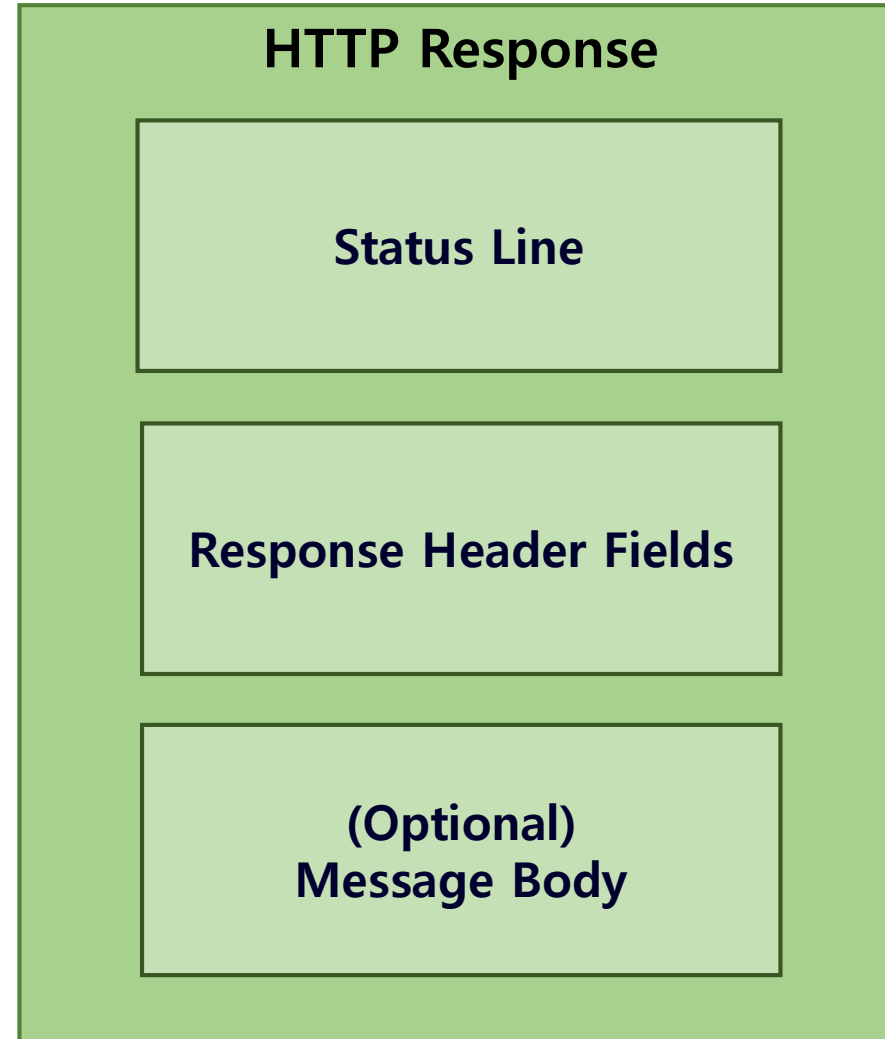
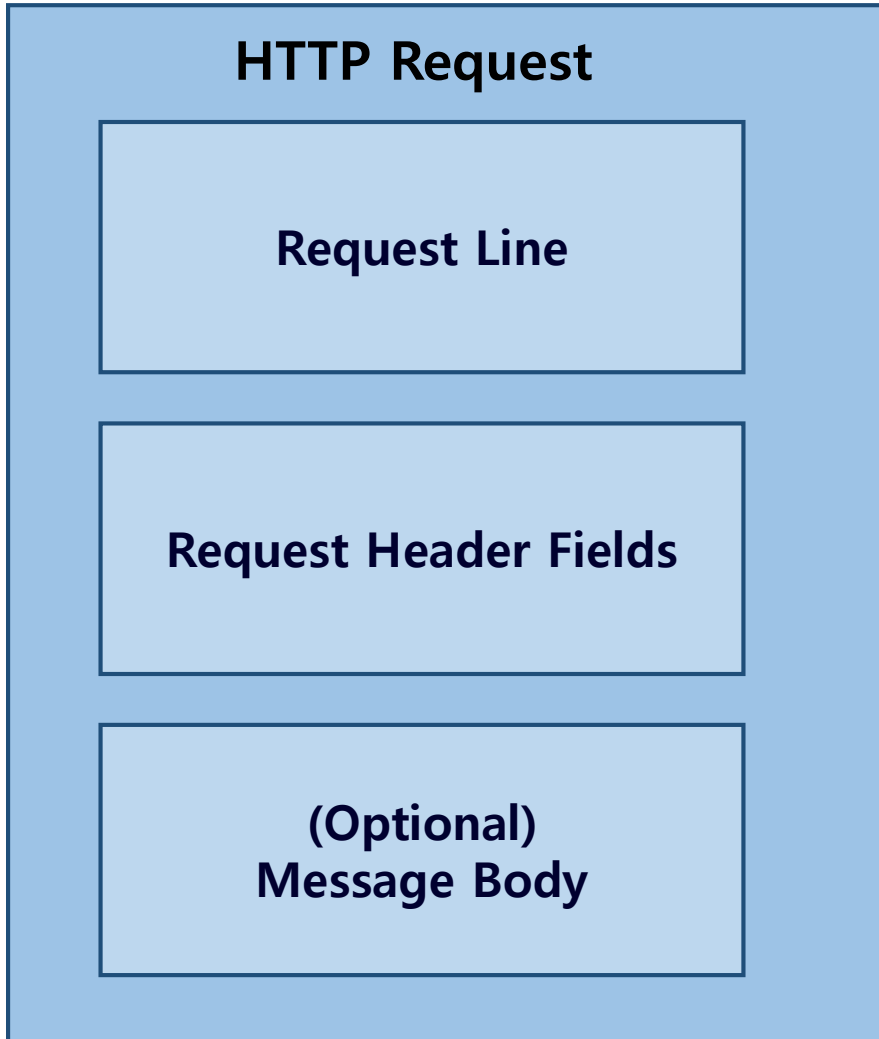
HTTP(HyperText Transfer Protocol)

WWW상에서 정보를 주고받을 수 있는 프로토콜(규약)

클라이언트와 서버 사이에 이루어지는 요청/응답 프로토콜

02. 실습 준비

HTTP 메시지 형식



02. 실습 준비

HTTP 메시지 예시

HTTP Request

GET /images/logo.png HTTP/1.1

Accept-Language: en

Content-Type: application/json

Host: 15.164.106.239:8080

Cache-Control: no-cache

```
{
  "id": "13109346",
  "name": "양희찬",
  "contacts": "01092484270"
}
```

HTTP Response

HTTP/1.1 200 OK or HTTP/1.1 404 Not Found

Accept-Language: en

Content-Type: application/json

Host: 15.164.106.239:8080

Cache-Control: no-cache

```
{
  "success": false,
  "message" "Already existed
memberId":
}
```

02. 실습 준비

HTTP Method

GET

The GET method requests a representation of the specified resource.
Requests using GET should only retrieve data.

POST

The POST method is used to submit an entity to the specified resource, often causing a change in state or side effects on the server.

PUT

The PUT method replaces all current representations of the target resource with the request payload.

DELETE

The DELETE method deletes the specified resource.

⋮

02. 실습 준비

HTTP Method

HTTP method ↕	RFC ↕	Request has Body ↕	Response has Body ↕	Safe ↕	Idempotent ↕	Cacheable ↕
GET	RFC 7231	Optional	Yes	Yes	Yes	Yes
HEAD	RFC 7231	Optional	No	Yes	Yes	Yes
POST	RFC 7231	Yes	Yes	No	No	Yes
PUT	RFC 7231	Yes	Yes	No	Yes	No
DELETE	RFC 7231	Optional	Yes	No	Yes	No
CONNECT	RFC 7231	Optional	Yes	No	No	No
OPTIONS	RFC 7231	Optional	Yes	Yes	Yes	No
TRACE	RFC 7231	No	Yes	Yes	Yes	No
PATCH	RFC 5789	Yes	Yes	No	No	No

02. 실습 준비

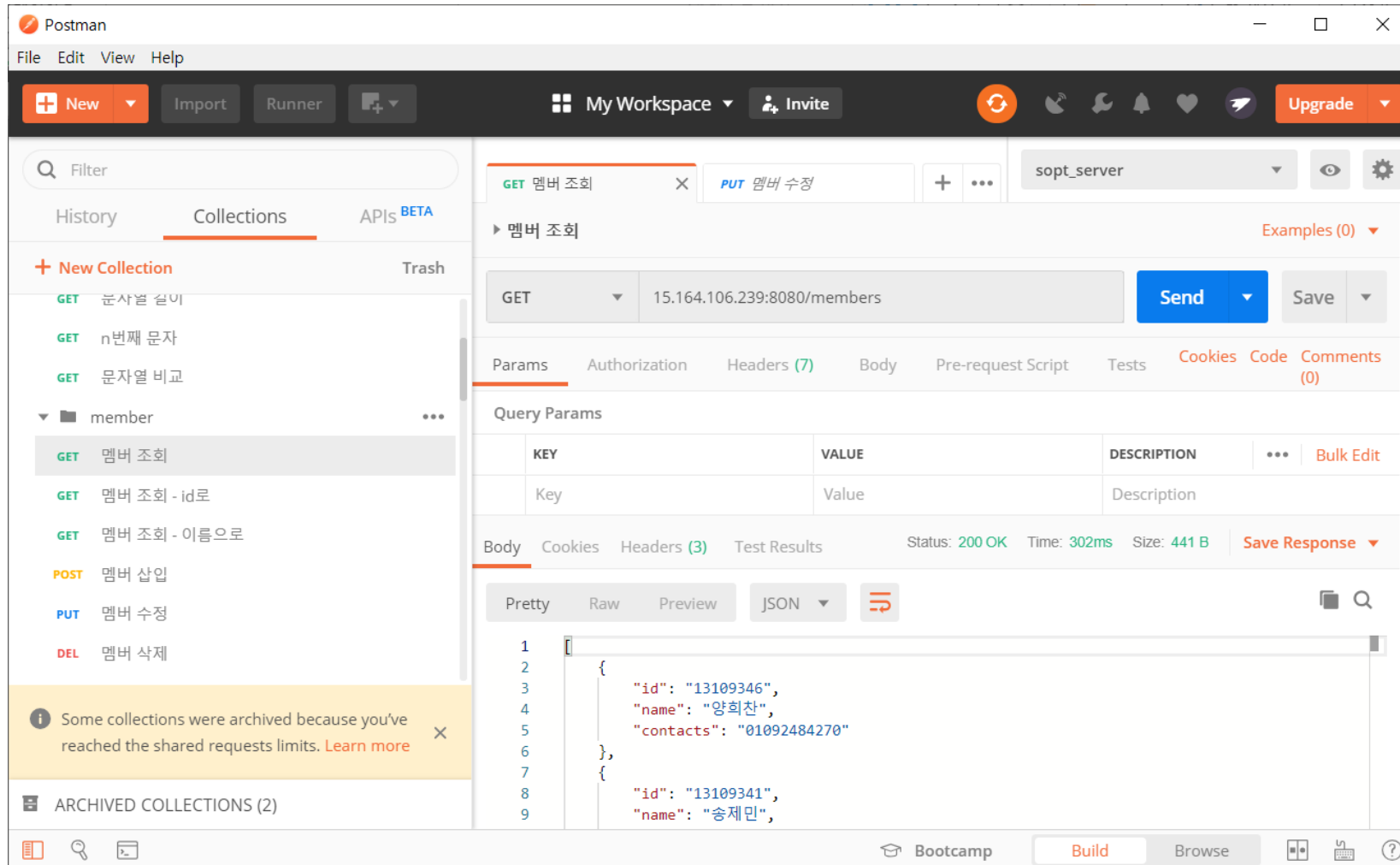
HTTP Method - 한글 자료의 폐해

HTTP 메소드	RFC	요청에 Body가 있음	응답에 Body가 있음	안전	멱등(Idempotent)	캐시 가능
GET	RFC 7231	아니오	예	예	예	예
HEAD	RFC 7231	아니오	아니오	예	예	예
POST	RFC 7231	예	예	아니오	아니오	예
PUT	RFC 7231	예	예	아니오	예	아니오
DELETE	RFC 7231	아니오	예	아니오	예	아니오
CONNECT	RFC 7231	예	예	아니오	아니오	아니오
OPTIONS	RFC 7231	선택 사항	예	예	예	아니오
TRACE	RFC 7231	아니오	예	예	예	아니오
PATCH	RFC 5789	예	예	아니오	아니오	예

02. 실습 준비

Postman

Postman



03 실습 (2)

#백건이 불여일타

02. 실습(2)

API 설계하기

- Member
 - 멤버 추가
 - 멤버 조회
 - 멤버 수정
 - 멤버 삭제
- Book
 - 도서 추가
 - 도서 조회
 - 도서 수정
 - 도서 삭제
- Borrowing
 - 도서 대출
 - 도서 대출 내역 조회
 - 도서 반납
 - 도서 대출 내역 삭제

02. 실습(2)

Domain Model

- Member
 - 학번
 - 이름
 - 연락처
- Book
 - ID
 - 제목
 - 저자명
- Borrowing
 - ID
 - 회원
 - 책
 - 빌린 시각
 - 대출 만료 시각
 - 반납 시각

02. 실습(2)

Domain Model Class 생성

```
public class Member {  
  
    private String id;  
    private String name;  
    private String contacts;  
  
    public String getId() { return id; }  
  
    public void setId(String id) { this.id = id; }  
  
    public String getName() { return name; }  
  
    public void setName(String name) { this.name = name; }  
  
    public String getContacts() { return contacts; }  
  
    public void setContacts(String contacts) { this.contacts = contacts; }  
}
```

```
public class Book {  
  
    private int id;  
    private String name;  
    private String author;  
  
    public int getId() { return id; }  
  
    public void setId(int id) { this.id = id; }  
  
    public String getName() { return name; }  
  
    public void setName(String name) { this.name = name; }  
  
    public String getAuthor() { return author; }  
  
    public void setAuthor(String author) { this.author = author; }  
}
```

```
public class Borrowing {  
  
    private int id;  
    private String memberId;  
    private int bookId;  
    private Date startTime;  
    private Date expiryTime;  
    private Date returnTime;  
    private Member member;  
    private Book book;  
  
    public int getId() { return id; }  
  
    public void setId(int id) { this.id = id; }  
  
    public String getMemberId() { return memberId; }
```

02. 실습(2)

Member Controller 작성

02. 실습(2)

Member Service 작성

02. 실습(2)

Book Controller 작성

02. 실습(2)

Book Service 작성

02. 실습(2)

Borrowing Controller 작성

02. 실습(2)

Borrowing Service 작성

04 배운 내용 정리

#왜 기억을 못하니 $\pi.\pi$

04. 배운 내용 정리

추가로 알아보면 좋을 것들

HTTP (status code, http method, headers, body, etc..)

AOP(Aspect Oriented Programming)

JSON(JavaScript Object Notation) vs XML(eXtensible Markup Language)

WAS(Web Application Server)

IoC(Inversion of Control)

DI(Dependency Injection)

MVC Model(Model-View-Controller)

Bean

기타 디자인 패턴(Singleton, Factory, Proxy, etc..)

RESTful API

Java Annotation

Project Build Tools (Gradle, Maven, Ant, etc...)

AWS (Amazon Web Service)

Persistence Framework (JPA, MyBatis)

RESTful API

04. 배운 내용 정리

어떻게 공부할까?

[백엔드 개발자를 꿈꾸는 학생개발자에게 - NAVER D2 - 네이버](#)

[2018년 웹 개발자가 되기 위한 로드맵 \(번역\) - OKKY](#)

인터넷 강의

- Inflearn / Udemy / Cosera / 생활코딩 / Youtube

Github

- 다른 사람들의 Github 구경

대외활동

- Naver D2 백엔드 밋업 - 현업 개발자들과 자신과 비슷한 사람들을 만날 수 있음
- Spring Camp - Spring을 사용하는 기업들이 와서 관련 경험과 팁을 발표
- SOPT - 대학생 연합 동아리. 개발, 디자인, 기획 파트로 구성되어 실제 창업 프로젝트 진행
- Nexters - SOPT와 비슷한 연합동아리. 개발자, 디자이너들로 구성
- 인턴 실습
- 공모전
- 등



Thank you

감사합니다

