



4-3 : SpringBoot 템플릿

들어가기 전에

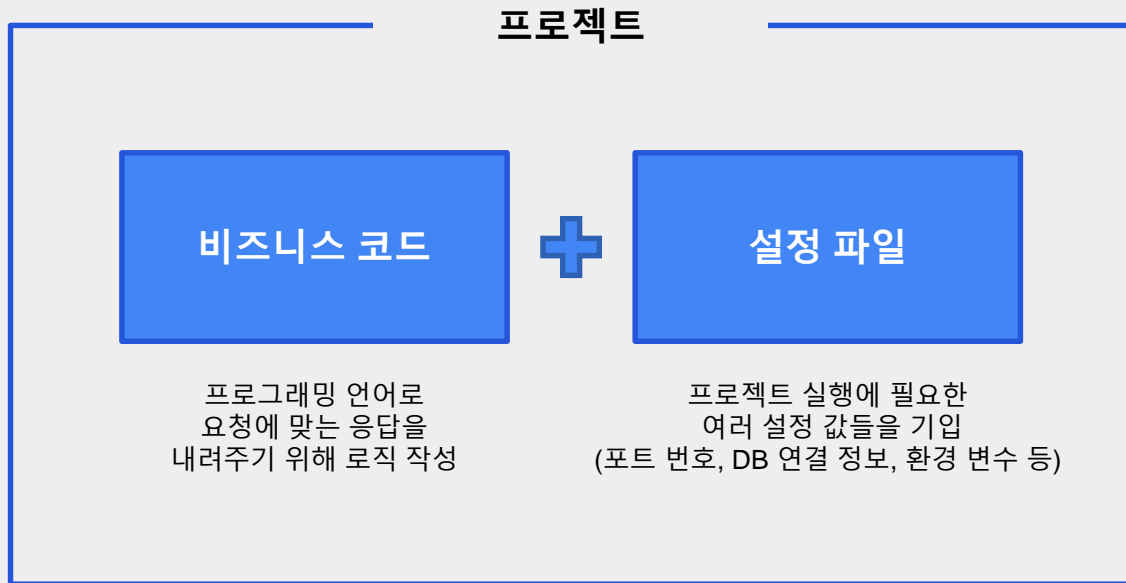


이번 시간에는...

프로젝트 개발 큰 그림
SpringBoot 템플릿 구성 설명
Request 처리 프로세스
Controller, Service, Dao

간단한 API 개발 실습
과제 설명

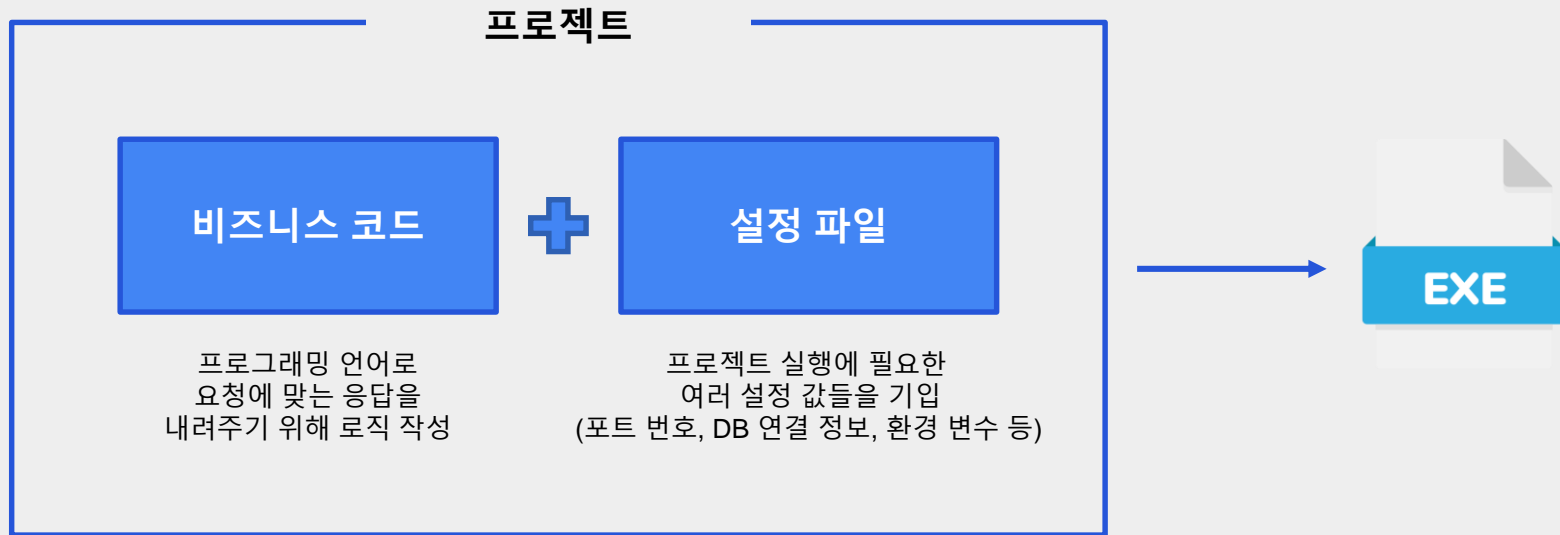
프로젝트 개발 큰 그림



프로젝트 개발

빌드 도구

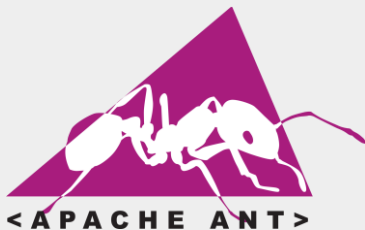
개발된 프로젝트 소스 코드를 실행 가능한 애플리케이션으로 만들어주는 도구



프로젝트 개발

빌드 도구

개발된 프로젝트 소스 코드를 실행 가능한 애플리케이션으로 만들어주는 도구
자바의 대표적인 빌드 도구에는 **Ant**, **Maven**, **Gradle** 등이 있다.



XML 기반 스크립트를 사용
규칙, 표준이 없어 높은 자유도
직접 정의할 값이 많아 사용이 어려움



Ant의 대안으로 개발
정해진 라이프사이클에 의해 동작
pom.xml에서 의존성 설정, 라이브러리 자동 관리

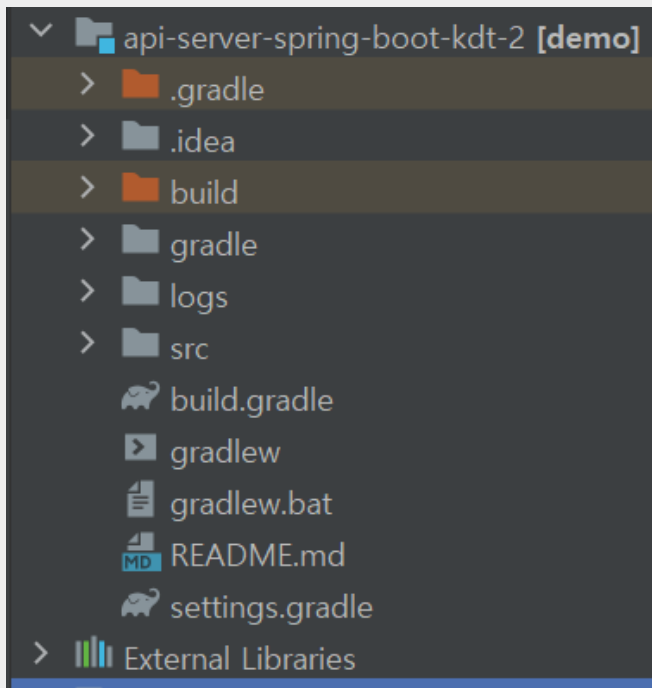


Ant의 유연한 구조 + Maven의 편리한 의존성 관리
Java, C/C++, Python 등 여러 언어 지원
xml이 아닌 별도의 스크립트로 프로젝트 관리
(if, else, for 문법 사용 가능)

SpringBoot 템플릿



SpringBoot 템플릿 구성

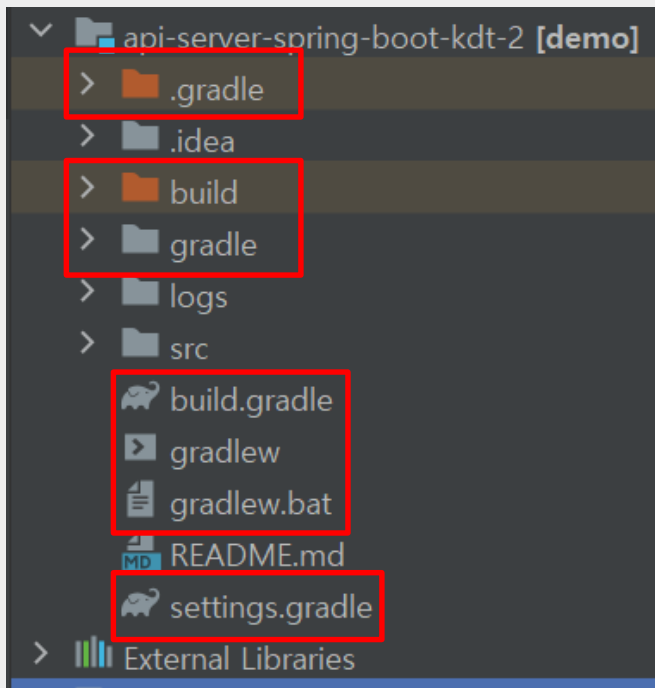


SpringBoot 템플릿



SpringBoot 템플릿 구성

빌드 도구(gradle) 관련 디렉토리, 파일



.gradle: gradle이 사용하는 폴더로 작업으로 생성된 파일이 저장

build: 빌드 파일

gradle: 빌드 시 참조하는 gradle-wrapper 파일 및 해당 파일의 설정 파일

build.gradle: 프로젝트 빌드에 대한 내용 명시

gradlew: Unix용 실행 스크립트

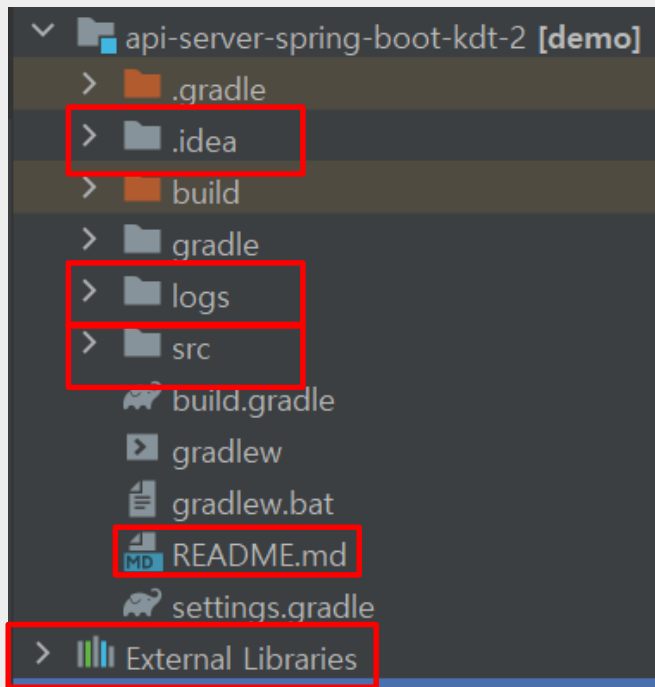
gradlew.bat: Window용 실행 스크립트

settings.gradle: 프로젝트의 설정 정보 파일(멀티 프로젝트 구성 등)

SpringBoot 템플릿

SpringBoot 템플릿 구성

빌드 도구 외 디렉토리, 파일



.idea: IDE 프로젝트 설정 값

logs: 로그 파일

src: 프로젝트의 소스 폴더

README.md: 프로젝트에 대한 설명

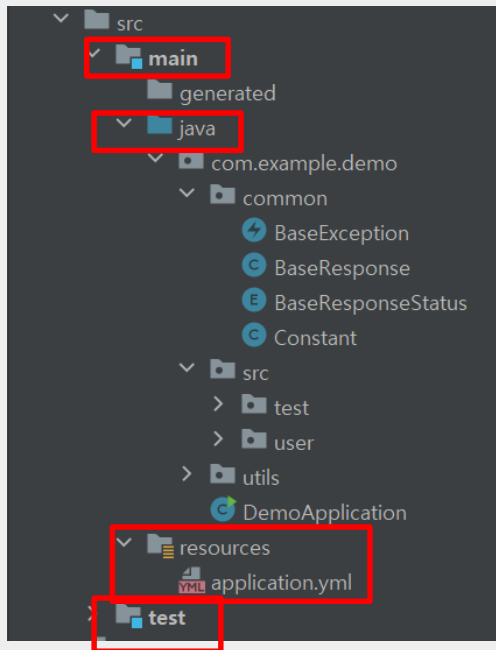
External Libraries: 라이브러리 파일

SpringBoot 템플릿



src 패키지 구성

src 패키지는 크게 main과 test로 구분되며,
main은 다시 java 코드 작성부와 설정 파일 부분으로 나뉜다.



main vs test

main은 프로젝트 코드를 실제로 직접 작성하는 곳

test는 테스트를 목적으로 테스트 관련 코드를 작성하는 곳

java vs resources

java는 java 프로그래밍 언어로 비즈니스 로직을 작성하는 곳

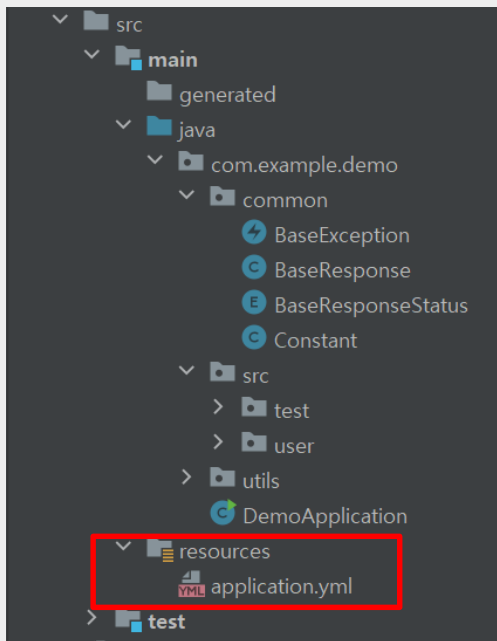
resources는 프로젝트에 필요한 설정 값들을 작성하는 곳

SpringBoot 템플릿



resources 패키지 구성

resources 패키지는 프로젝트에 필요한 설정 파일들이 위치하는 곳
가장 대표적으로 application.yml 파일이 있다.



```
server:
  port: 9000

spring:
  application:
    name: demo

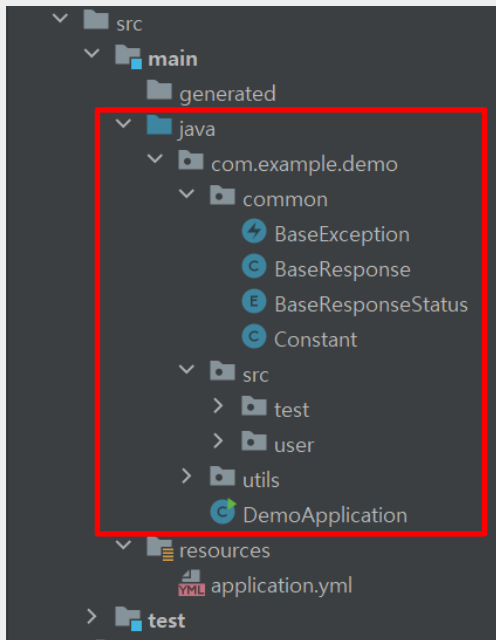
  datasource:
    platform: mysql
    url: jdbc:mysql://database-1.cfbtn77emuhe.ap-northeast-2.rds.
    username: # username
    password: # password
    driver-class-name: com.mysql.cj.jdbc.Driver # mysql 8버전
    # driver-class-name: com.mysql.jdbc.Driver # mysql 5버전
```

port 번호, spring 설정, database 연동 설정,
로그 설정, 환경 변수 등

SpringBoot 템플릿

java 패키지 구성

java 패키지는 직접 비즈니스 로직을 작성하는 패키지
하위 패키지는 정해진 형식이 없어 자유롭게 구성이 가능
여기서는 common, src, utils 패키지와 Application class로 구성



common

비즈니스 로직을 작성하는데 필요한 설정 코드들
상수, 로직 작성에 필요한 기본 양식 등을 작성

src

비즈니스 로직을 직접 작성하는 곳
각 비즈니스 도메인 별로 패키지를 구분
로직에 필요한 객체는 Model 패키지에 작성
로직은 Controller, Service, Dao로 나누어 작성

utils

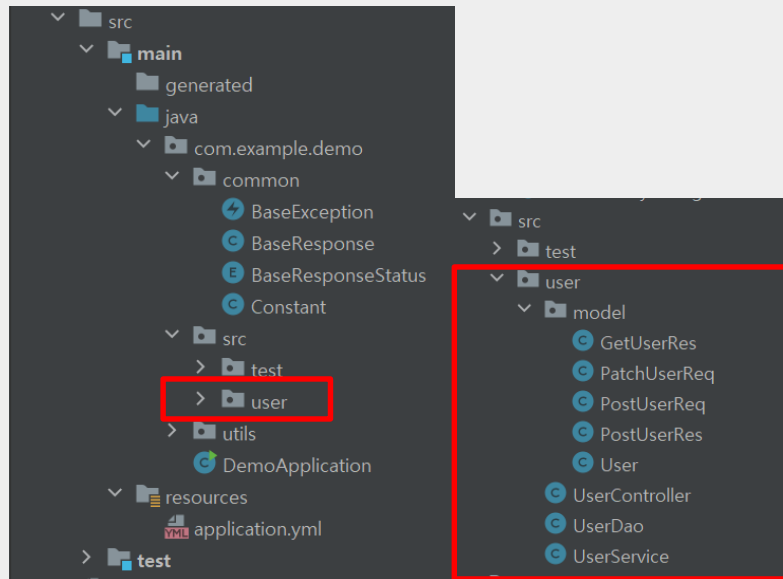
비즈니스 로직 중 반복되는 코드, 메소드 등을 작성
그 외 부가적인 코드 작성 등

SpringBoot 템플릿



도메인 패키지 구성

java 패키지 내에 각 도메인 패키지를 구현
각 도메인 패키지는 크게 model, controller, service, dao로 구성



model

로직에 사용되는 객체 class 정의(DTO)

Controller

사용자의 요청을 처리하기에 적절한 서비스 단으로 연결해주는
다리 역할

Service

사용자의 요청을 본격적으로 처리하는 코드 작성

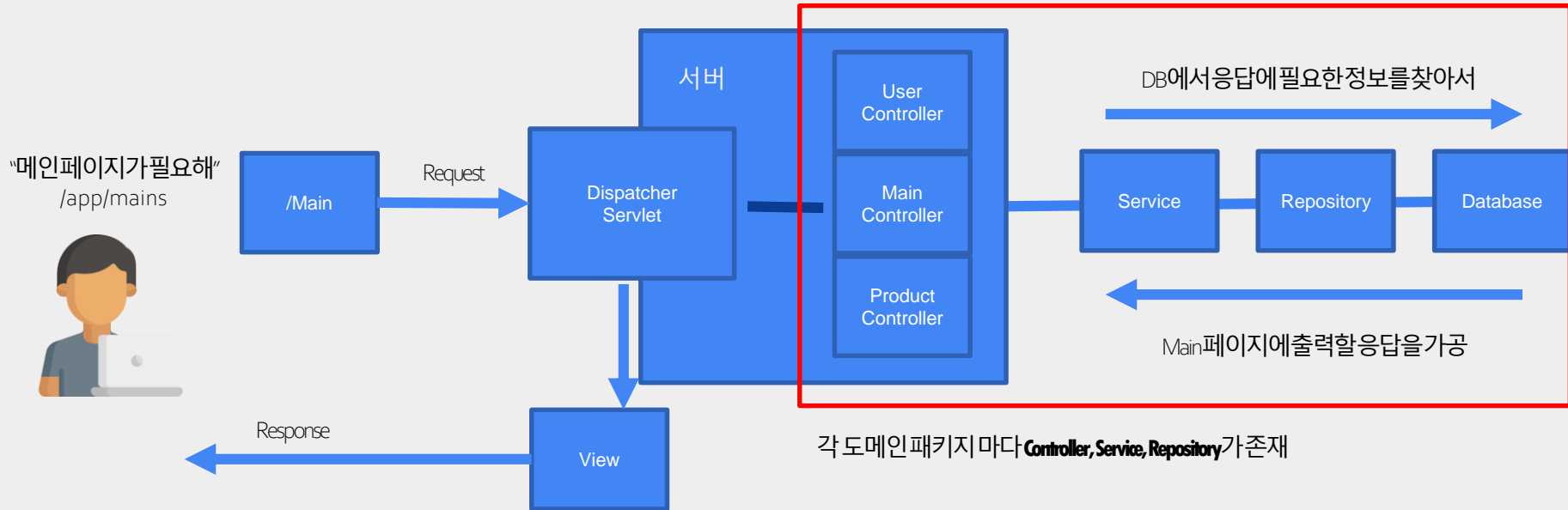
Dao

DB 데이터에 접근하기 위한 코드 작성

WAS 개발 큰 그림



클라이언트가 특정 **API**를 요청하면,
해당 **API**에 맞는 응답을 처리해주는 **Controller, Service, Repository(DAO)**를 개발



각도메인 패키지마다 **Controller, Service, Repository**가 존재

내려줄 페이지 선택 or 곧바로 데이터 Response

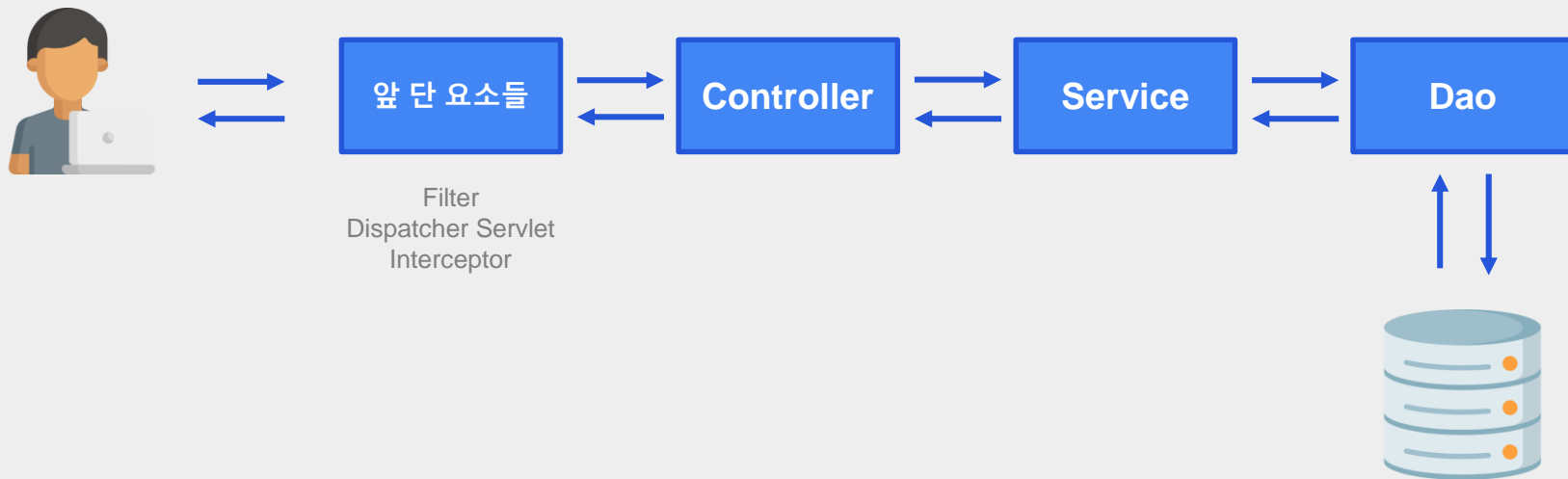
스프링은 API에 해당하는 페이지를 내려줄 수도 있고,
단순히 데이터만 내려줄 수도 있다. ex) CSR, SSR

request 처리



request 처리 프로세스

java 패키지 내에 각 도메인 패키지를 구현
각 도메인 패키지는 크게 model, controller, service, dao로 구성

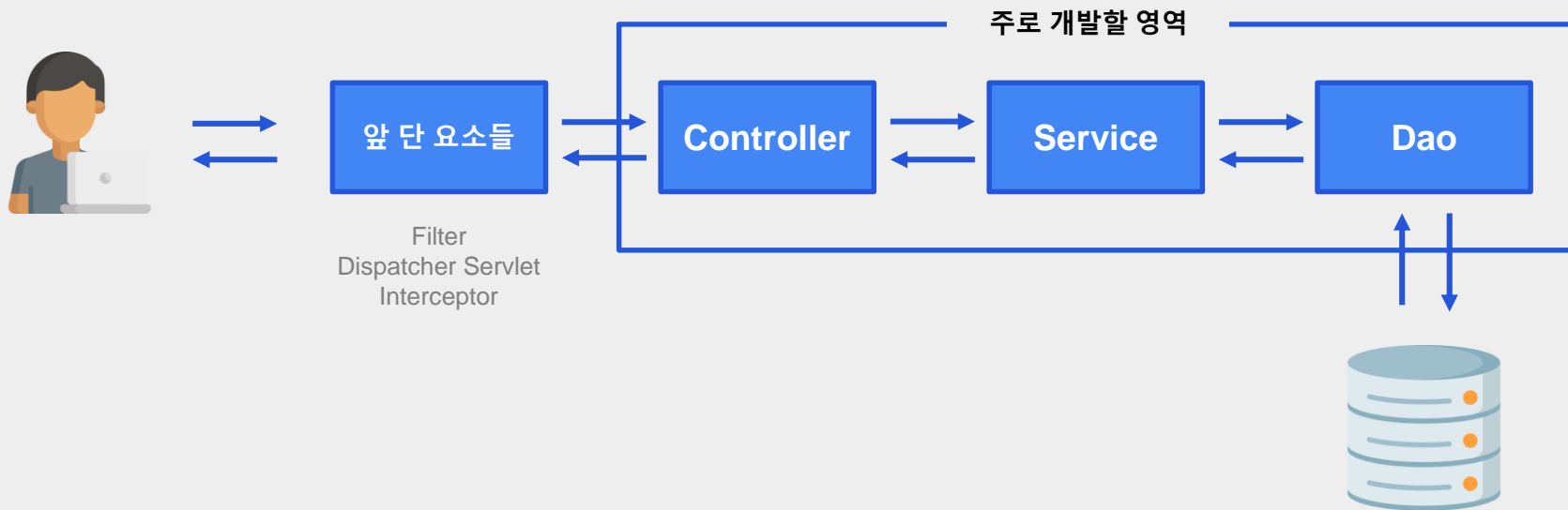


request 처리



request 처리 프로세스

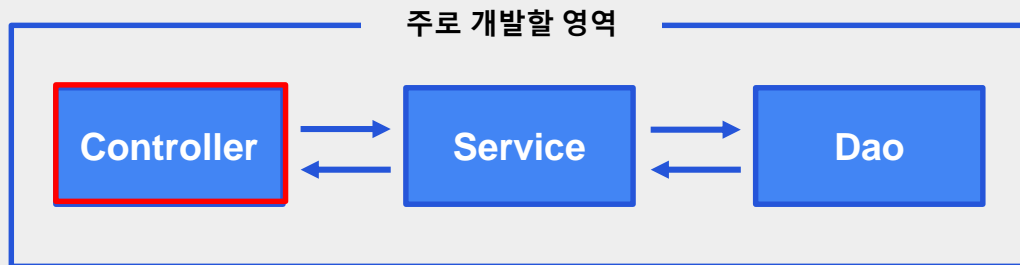
java 패키지 내에 각 도메인 패키지를 구현
각 도메인 패키지는 크게 model, controller, service, dao로 구성



Controller



Controller

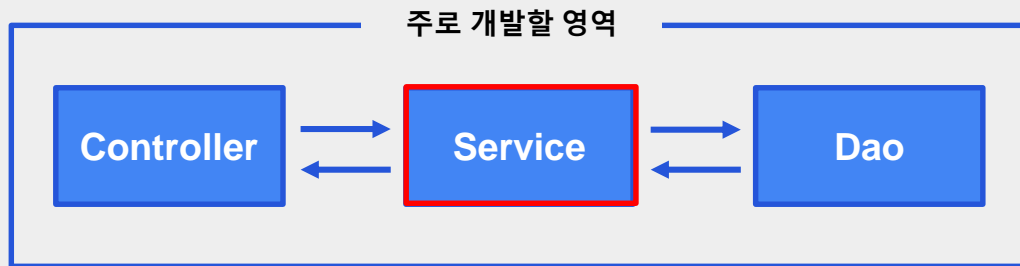


- Controller 관련 어노테이션
- Controller 메소드 구성(CRUD Mapping, 요청 & 응답 객체 등)
 - 형식적 예외처리
- BaseResponse(HTTP 상태 코드)

Service



Service

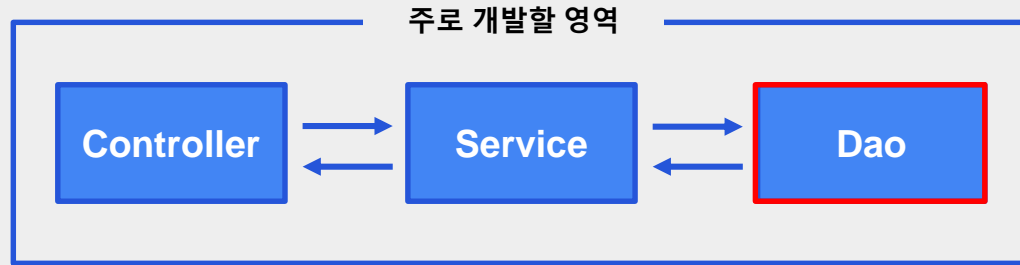


- Service 관련 어노테이션 설명
 - 논리적 예외처리

Dao



Dao



- Dao 관련 어노테이션 설명
- JDBC Template

SpringBoot 구축



SpringBoot 구축

로컬 컴퓨터에서 SpringBoot 템플릿을 실행해보자.

1. SpringBoot 템플릿 다운로드
2. IntelliJ에서 템플릿 Open(build.gradle 클릭)
3. RDS 연결(application.yml 파일 설정)
4. SpringBoot 프로젝트 로컬 실행
5. 로컬에서 <http://127.0.0.1:9000/test/log> 접속해보기

user API 요청 실습

포스트맨을 활용하여 user API를 요청해보자

1. 포스트맨 다운로드
2. RDS DB서버에 Database를 생성하고, USER 테이블을 생성
3. SpringBoot 프로젝트 로컬 실행
4. userController를 참고하여 CRUD API 요청해보기

API 개발 실습

제시된 요구사항에 맞게 RDS에 테이블을 생성하고,
SpringBoot test 패키지에 관련 API를 개발해보자

API 명세서: <https://documenter.getpostman.com/view/16596703/2s8YzQXjFY>

요구 사항

POST /test/memos: test 테이블에 메모 저장

GET /test/memos: test 테이블의 메모들 모두 조회

PATCH /test/memos: test 테이블의 메모 수정

과제 안내



REST API 설계

2주차에 구현한 ERD를 참고하여 **REST API를 설계**하고,
주어진 양식에 맞게 **API 명세서를 작성**하기

과제 유의사항

REST API 설계 규칙에 맞게 설계하기

간단한 CRUD를 기준으로 설계

최소 20개 설계(POST, GET, PATCH, DELET 각 메소드 4개 이상씩 구성)

과제 안내



API 개발

SpringBoot 템플릿을 활용하여 설계한 API를 개발
API 개발 후, 포스트맨 등을 활용하여 정상적으로 동작하는지 확인

과제 유의사항

간단한 CRUD API 위주로 개발
로그인, 로그아웃 기능은 제외
각 API에서 필요한 예외처리 최대한 구현해보기

Hello World!