

초격차 패키지 Online.

API 설계

PART1 | 애노테이션 기반 설계

@Controller, @RestController

PART2 | 함수 기반 설계

함수형 프로그래밍으로 만드는 API

PART3 | 요청, 응답의 설계

Handler Methods의 활용

PART4 | @ControllerAdvice

공통 에러 응답을 설계하는 법

PART5 | 컨트롤러 테스트

Spring Web API를 테스트하는 방법

PART6 | TDD 방식으로 복습하기

테스트 주도 개발 과정 맛보기

API 설계

1 애노테이션 기반 설계

작업 내용 파악

1. 애노테이션 기반 설계

종류	uri	method	
web	/	GET	
	/error	GET	
	/login	GET	
	/sign-up	GET	
	/admin/places	GET	
	/admin/places/{place-id}	GET	
	/places	GET	
	/places/{place-id}	GET	
	/admin/events	GET	
	/admin/events/{event-id}	GET	
	/events	GET	
	/events/{event-id}	GET	
api	/api/sign-up	POST	이메일, 닉네임, 비밀번호, 전화번호, 메모
	/api/login	GET	이메일, 비밀번호
	/api/places	GET	필터: [장소 유형, 이름, 전화번호], 정렬: [(id,) 장소 유형, 이름, 주소, 변경날짜]
	/api/places/{place-id}	GET	
	/api/places	POST	장소 유형, 이름, 주소, 전화번호, 최대 수용 인원, 메모
	/api/places/{place-id}	PUT	장소 유형, 이름, 주소, 전화번호, 최대 수용 인원, 메모
	/api/places/{place-id}	DELETE	
	/api/events	GET	필터: [장소, 이름, 상태, 기간], 정렬: [(id,) 장소, 이름, 상태, 기간, 변경날짜]
	/api/events/{event-id}	GET	
	/api/events	POST	장소, 이름, 기간, 최대 참석 인원, 메모
	/api/events/{event-id}	PUT	이름, 상태, 기간, 최대 참석 인원, 메모
	/api/events/{event-id}	DELETE	

애노테이션 기반 설계

1.

애노테이션 기반 설계

컨트롤러 클래스

MVC 패턴 중 핸들러 메소드를 포함하는 컨트롤러 빈을 만드는 과정

- @Controller
- @RestController = @ResponseBody + @Controller

애노테이션 기반 설계

1.

애노테이션 기반 설계

핸들러 메소드 (Handler Method)

스프링 웹서비스가 받는 URI 요청을 컨트롤러 클래스의 특정 메소드에 매핑하는 과정

- @RequestMapping
- @GetMapping
- @PostMapping
- @PutMapping
- @DeleteMapping
- @PatchMapping