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

2 함수 기반 설계

함수형 프로그래밍?

함수형 프로그래밍(函數型 프로그래밍, 영어: functional programming)은 자료 처리를 수학적 함수의 계산으로 취급하고 상태와 가변 데이터를 멀리하는 프로그래밍 패러다임의 하나이다.

Functional programming is programming without assignment statements.

함수형 프로그래밍?

- 특징
  - 상태가 없음
  - 대입문이 없음
  - 부작용(side effect)이 없는 순수 함수
  - 불변성 (Immutability)
- 역사: 오래되었다!
  - 1930, 람다 대수
  - 1954, IPL
  - 1958, LISP
  - 1990, Haskell

함수형 엔드포인트

Spring Web 의 엔드포인트를 함수형 스타일로 작성하는 방법을 제공

- WebMvc.fn
- routing, request handling
- 불변성을 고려하여 설계됨
- 기존의 DispatcherServlet 위에서 동작
- 애노테이션 스타일과 함께 사용 가능

주요 키워드

- HandlerFunction == @RequestMapping
  - 입력: ServerRequest
  - 출력: ServerResponse
- RouterFunction == @RequestMapping
  - 입력: ServerRequest
  - 출력: Optional<HandlerFunction>

HandlerFunction vs. RouterFunction

- HandlerFunction 의 결과: data
- RouterFunction 의 결과: data + behavior (ex: url mapping)

기타 세부 키워드

- RequestPredicates
- RouterFunctions.route().nest()
- RouterFunctions.route().before()
- RouterFunctions.route().after()
- RouterFunctions.Builder.onError()
- RouterFunctions.Builder.filter()

Reference

- https://ko.wikipedia.org/wiki/함수형_프로그래밍
- https://en.wikipedia.org/wiki/Functional_programming
- https://blog.cleancoder.com/uncle-bob/2012/12/22/FPBE1-Whats-it-all-about.html
- https://docs.spring.io/spring-framework/docs/current/reference/html/web.html#webmvc-fn