

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

## 3 요청, 응답의 설계

## Handler Methods

### 3. 요청, 응답의 설계

### 핸들러 메소드란

Spring Web 에서 사용자의 요청(request)을 받아 응답(response)을 리턴하는 메소드

1. 매핑 정보
2. 요청
3. 응답

```
@GetMapping(1. "/places/{placeId}")  
public String getPlace2. @PathVariable Integer placeId) {  
    3. return "place " + placeId;  
}
```

## 1. 매핑 정보

## 3. 요청, 응답의 설계

### @RequestMapping

- name: 뷰 템플릿에서 식별할 때 쓰는 이름
- value, path: URI
- method: HTTP method (ex: GET, POST, ...)
- params: 파라미터 검사
- headers: 헤더 검사
- consumes: 헤더의 Content-Type 검사
- produces: 헤더의 Accept 검사

```
@RequestMapping(
    name = "IAmGetPlaces", // 뷰 템플릿에서 나를 부를 수 있는 이름이야!
    value = "/places", // 내 URI 경로
    path = "/places", // 나한테
    method = RequestMethod.GET, // HTTP method: GET
    params = "test=true", // test=true 파라미터 있는 요청만.
    headers = "header-auth=stupidPassword", // header-auth 헤더값 있는 요청만.
    consumes = MediaType.APPLICATION_JSON_VALUE, // json 으로 데이터 주는 요청만.
    produces = MediaType.APPLICATION_JSON_VALUE // json 데이터 받을 수 있어?
)
@GetMapping("/places")
public List<String> getPlaces() {
    return List.of("place1", "place2");
}
```

## 1. 매핑 정보

## 3. 요청, 응답의 설계

### @RequestMapping shortcuts

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

## 2. 요청

## 3. 요청, 응답의 설계

### 핸들러 메소드가 받을 수 있는 요청들

메소드 파라미터로 적어 넣을 수 있는 타입들

- ServletRequest, ServletResponse, HttpSession
- WebRequest, NativeWebRequest
- @RequestParam, @PathVariable
- @RequestBody, HttpEntity<B>
- @ModelAttribute, @SessionAttribute, Model, ModelMap
- @RequestHeader, @CookieValue
- Principal, Locale, TimeZone, InputStream, OutputStream, Reader, Writer, ....
- (넘 많음)

### 3. 응답

### 3. 요청, 응답의 설계

## 핸들러 메소드가 내보낼 수 있는 응답들

메소드가 리턴할 수 있는 타입들

- ModelAndView
- String, View
- @ModelAttribute, Map, Model
- @ResponseBody
- HttpEntity<B>, ResponseEntity<B>
- HttpHeaders
- void
- 등등

## Reference

### 3. 요청, 응답의 설계

- <https://docs.spring.io/spring-framework/docs/current/reference/html/web.html#mvc-ann-methods>