

Building a RESTful Web Application with Spring Boot



Kesha Williams

SOFTWARE ENGINEERING MANAGER & PROFESSOR

@KeshaWillz www.kesha.tech



Overview



Build a RESTful service

REST architecture style

HTTP response codes

Annotations

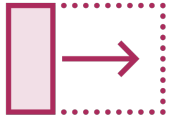
Exception Handling



REST Architecture Style



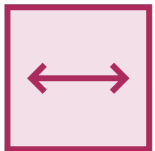
REST Architecture Style



Data and functionality are considered resources



Resources are manipulated using a fixed set of operations



Resources can be represented in multiple formats



Communication between the client and endpoint is stateless

Demo



RESTful APIs

@RestController annotation

@RequestMapping/@GetMapping



Response Formats



```
Class ResponseEntity<T>
```

```
return new  
ResponseEntity<List<Ticket>  
>(list, HttpStatus.OK);
```

```
return new  
ResponseEntity<List<Applica  
tion>>(list, HttpStatus.OK);
```

◀ ResponseEntity is generic

◀ Returning Tickets

◀ Returning Applications



Response Codes

`HttpStatus.OK` - 200

`HttpStatus.BAD_REQUEST` -
400

`HttpStatus.CONFLICT` - 409

`HttpStatus.NOT_FOUND` -
404 Not Found



Exception Handling



Response Codes

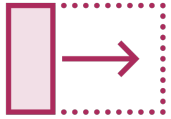


Success Status Code
200



Failure Status Code
404

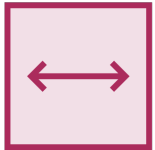
ResponseStatusException



Programmatic alternative to `@ResponseStatus`



Provide `HttpStatus` and a reason and a cause



Exceptions can be created programmatically



Provides a default error mapping



Demo



Exception Handling
ResponseStatusException
ResponseStatusExceptionResolver



Summary



Build RESTful APIs

REST architecture style

@RestController

Automatic Configuration

ResponseStatusException

