

Module 2 - Lecture 15

# Server-Side APIs: Part 2



# REpresentational State Transfer

- Started in the year 2000 by a group of experts to simplify web services.

## Architecture

- Resource-oriented
- Interface uniformity
- Stateless

## Benefits

- Uniformity reduces dependency on documentation
- Stateless reduces complexity



# REpresentational State Transfer

- URLs, HTTP Methods, Response Status Codes, etc. follow

Operation	HTTP Method	URL	Use
Create	POST	/products	Create a new product using data passed in the request
Retrieve	GET	/products	Retrieve all products
Retrieve	GET	/products? name=headphones	Retrieve all products that match the search
Retrieve	GET	/products/5	Retrieve the product with an ID of 5
Update	PUT	/products/5	Update an existing product with an ID of 5 using data passed in the request
Delete	DELETE	/products/5	Delete the product with an ID of 5

# Java Bean Validation

## Two steps

1. Define validation rules in the Model class.
2. Apply `@Valid` annotation to anywhere you want Spring to perform validation.
  - a. Spring will perform validation after Model binding, and before executing the handler method.
  - b. If validation fails, Spring will throw an exception of `MethodArgumentNotValidException`.
  - c. A status code of 400 Bad Request will be returned.



# Java Bean Validation

## Common Rules

Annotation	Applies to	Description
@NotNull	Any	The variable can't be null.
@NotEmpty	array, List, String	The variable can't be null. Also, if it's a list or array, it can't be empty. If it's a String, it can't be an empty string.
@NotBlank	String	The variable can't be empty or only contain white space characters.
@Min	int, long, BigDecimal	The variable must have a value greater than the specified minimum. Null values are skipped.
@DecimalMin	BigDecimal	The variable must have a value greater than the specified minimum decimal. Null values are skipped.
@Size	array, List, String	The variable length, as an array, List, or String, must be within the bounds specified. Null values are skipped.
@Past, @Future	Date, LocalDate	The variable must represent a date in the past/future. Null values are skipped.
@Pattern	String	The variable must hold a value that matches the specified regular expression. Null values are skipped.



# Dependency Injection

Dependency Injection is where instances of classes that are depended on are injected into a new object rather than created by that object.

```
private HotelDao dao;

public HotelController() {
    dao = new MemoryHotelDao();
}
```

```
private HotelDao dao;

public HotelController(HotelDao dao) {
    this.dao = dao
}
```

```
@Component
public class MemoryHotelDao implements HotelDao {
    ...
}
```

QUESTIONS?

