# Designing RESTful Applications

Sang Shin
JPassion.com
"Code with Passion!"

# Things involved in JAX-RS Application

1. Create Object models
2. Design URIs
3. Determine Data formats
4. Determine HTTP methods to use

# 1. Create Object Models

# Object models, Object relationship

- Object model gets created from use cases
- UML class diagram represent classes and their relationships
- Object models typically results in resources
- Example Object models
  - Customer
  - Order
  - Line item
  - Product

# Use case scenarios

- Retrieve all customers
- Create, read, update, delete a customer
- Retrieve all orders of a customer
- Create, read, update, delete an order for a customer
- Compute the average price of orders for a customer
- ...

# 2. Design URI's

# Design URI's for the resources

- Define the endpoints representing resources
  - http://jpassion.com/customers/{id}
  - http://jpassion.com/orders/{id}
  - http://jpassion.com/products/{id}
  - http://jpassion.com/customers/{id}/orders/average-price

# 3. Determine Data Formats

# Format of the data exchanged

- Data can be represented in multiple formats
- Examples
  - XML
  - JSON

# 4. Determine HTTP Methods

# Determine HTTP methods

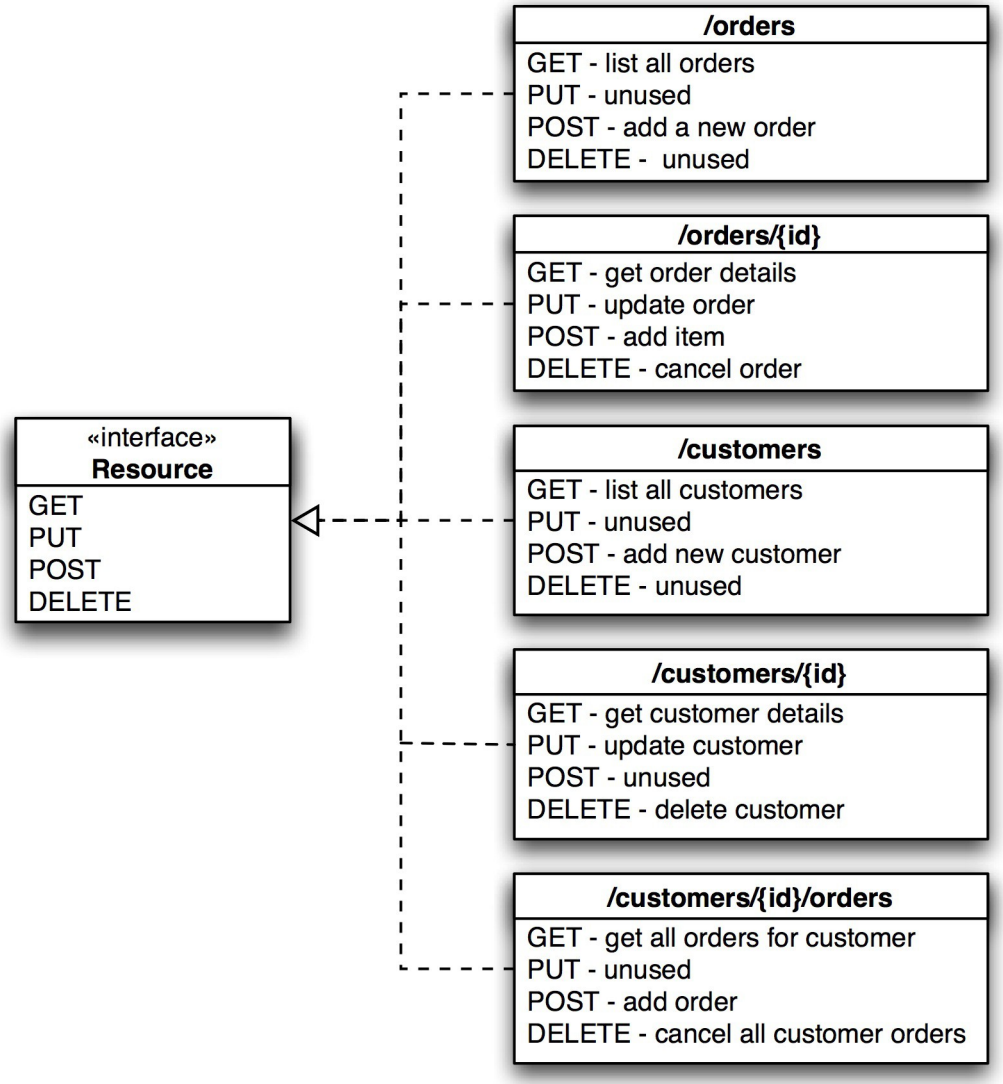
- Follow HTTP semantics – follow safety and idempotency requirements of HTTP methods
  - Safety – the method does not change the state of the resource
  - Idempotency – the method can be called repeatedly and always returns the same result
- Not following HTTP semantics results in clients who cannot make assumptions on your services

# Safety and Idempotency of HTTP methods

| Method | Safe? | Idempotent? |
|---|---|---|
| GET | Yes | Yes |
| HEAD | Yes | Yes |
| OPTIONS | Yes | Yes |
| PUT | No | Yes |
| DELETE | No | No |
| POST | No | No |

# HTTP Methods:

Customer Order Management Example




http://www.infoq.com/articles/rest-introduction

# HTTP Methods:

- /orders
  - GET - list all orders
  - POST - submit a new order

  /orders/{order-id}
  - GET - get an order representation
  - PUT - update an order
  - DELETE - cancel an order

  /orders/average-sale
  - GET - calculate average sale

- /customers
  - GET - list all customers
  - POST - create a new customer

  /customers/{cust-id}
  - GET - get a customer representation
  - DELETE- remove a customer

  /customers/{cust-id}/orders
  - GET - get all orders of a customer


http://www.infoq.com/articles/rest-introduction

# CRUD Operations

# CRUD Operations are Performed through “HTTP method” + “Resource”

## CRUD Operations

| | HTTP method | Resource |
|---|---|---|
| Create (Single) | POST | Collection URI |
| Read (Multiple) | GET | Collection URI |
| Read (Single) | GET | Entry URI |
| Update (Single) | PUT | Entry URI |
| Delete (Single) | DELETE | Entry URI |

# Code with Passion!
# JPassion.com