



# REST-ful in Practice

SPRINT3R

Siam Chamnankit Co., Ltd., Odd-e (Thailand) Co., Ltd. and Alliance

# REST ?

**RE**presentational **S**tate **T**ransfer

**Architectural style** for design distributed system

Not a standard, but a **set of constraints**

Not tied to HTTP, but associated with it !!

# REST Constraints

1. Uniform Interface
2. Stateless Interaction
3. Cacheable
4. Client-Server
5. Layered System
6. Code on Demand

# Uniform Interface

**Guidelines** than **Rules**

- Identification of resources
- Manipulation of resources
- Self-Descriptive of messages
- Hypermedia as the Engine of Application State

# HTTP's Uniform Interface

**URI's** identify resources

**HTTP verbs** describe a set of operations  
that used to manipulate a resource

**Header** help to describe the message

# URI vs URL ?

# HTTP Verbs

**GET**

**DELETE**

**PUT**

**POST**

less used other verbs

# GET

Use to retrieve information

Must be safe and idempotent/repeatable

No side effect

GET can be conditional or partial

**GET /games/1**



# DELETE

Request that a resource be removed

The resource **doesn't** to be removed Immediately

Removal may be a long running process

**DELETE /games/1**

# PUT

Request that have entity

Entity passed by stored at the URI

Entity called **PAYLOAD**

Use to create new entity

Use to modify and existing one

**PUT /games/1/doors/2**

**{ “status” : “SELECTED” }**

# POST

Request that the resource at URI do **something**  
Something is could be **Create, Modify**

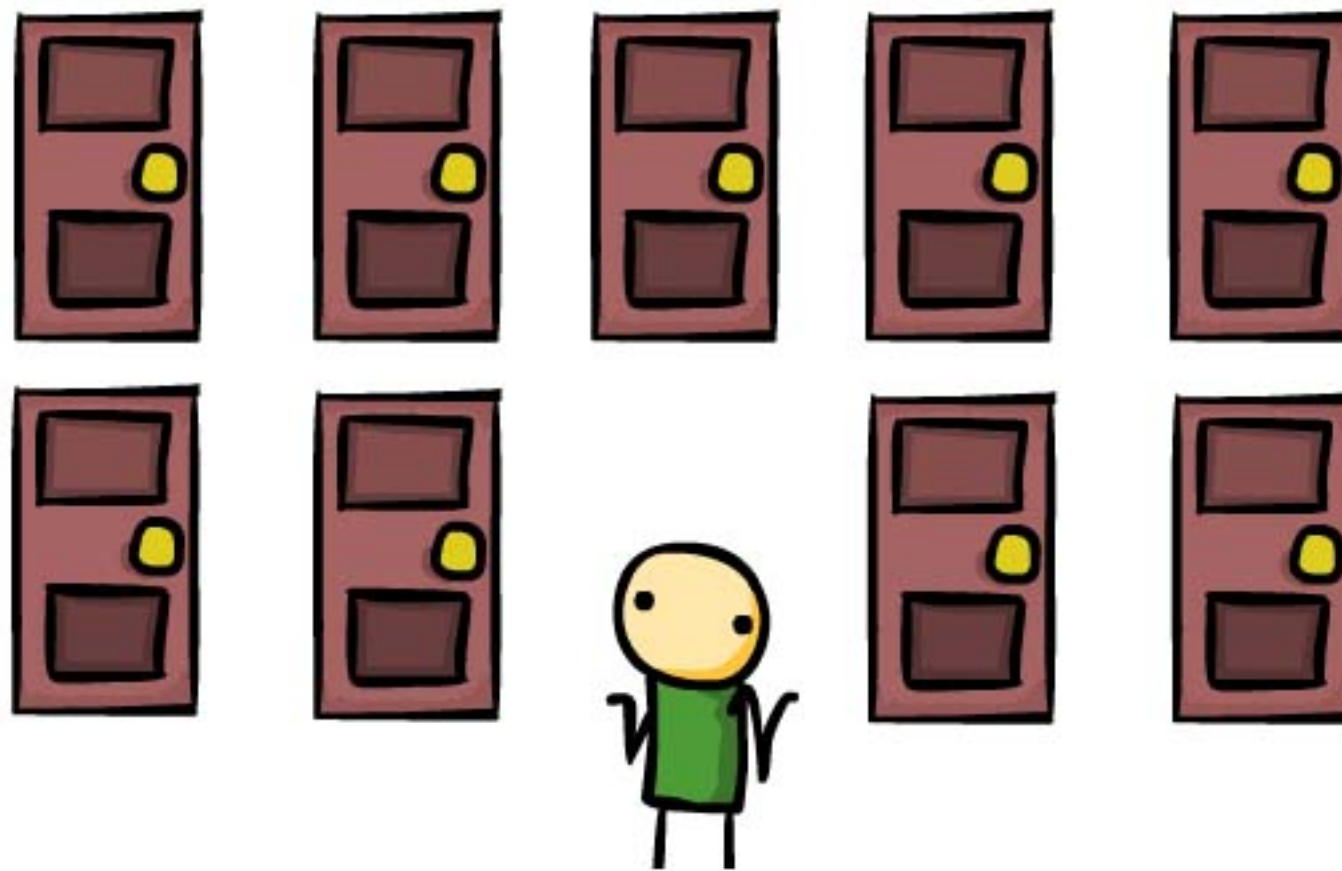
**POST /games/**

# PUT vs POST ?

# Design REST APIs

# Door Game

One of the doors lead to yayness, the rest lead to death. Good luck!



# Interaction model ?

# Interaction model

Create new game

List current state of all doors

Select a door

Open a door

List the final state of the game

Delete game

...



# Create new game

First endpoint of system !!

Doesn't require any input

Need to return us a resource identifier (URI)  
of the new created game

**HTTP Verb and URI ?**

# Create new game

First endpoint of system !!

Doesn't require any input

Need to return us a resource identifier (URI)  
of the new created game

**POST /games**

# List current state of all doors

Need to return a collection of door's state

*Design doesn't specific number of door !!*

**HTTP Verb and URI ?**

# List current state of all doors

Need to return a collection of door's state

*Design doesn't specific number of door !!*

**GET /games/0/doors**

[{ "status" : "CLOSED" }, { "status" : "OPEN", { ... } }]

# Select a door

No SELECT in HTTP verb !!

How to represent the selection of a door ?

**HTTP Verb and URI ?**

# Select a door

No SELECT in HTTP verb !!

How to represent the selection of a door ?

**PUT /games/0/doors/2**

{ “status” : “SELECTED” }

# Open a door

Like select a door !!

Use same endpoint !!

**HTTP Verb and URI ?**

# Open a door

Like select a door !!

Use same endpoint !!

**PUT /games/0/doors/2**

{ “status” : “OPEN” }



# List final state of the game

Need to return an object that represent  
the state of the game

**HTTP Verb and URI ?**

# List final state of the game

Need to return an object that represent  
the state of the game

**GET /games/0**

{ “status” : “WIN” }

# Delete the game

No input required

No output required

**HTTP Verb and URI ?**

# Delete the game

No input required

No output required

**DELETE /games/0**

# Develop REST APIs

# Framework with Java ?



spring  
boot



Spring  
HATEOAS



Jersey

RESTful Web Services in Java.



DROPWIZARD

Make features not WAR



# HTTP status code

Indicator of the result of the server's attempt to satisfy the request

## **Main category of status code ::**

1xx : Informational

2xx : Success

3xx : Redirection

4xx : Client error

5xx : Server error

# Success status code

200 OK

201 Created

202 Accepted



# Client error status code

400 Bad Request

401 Unauthorised

403 Forbidden

404 Not Found

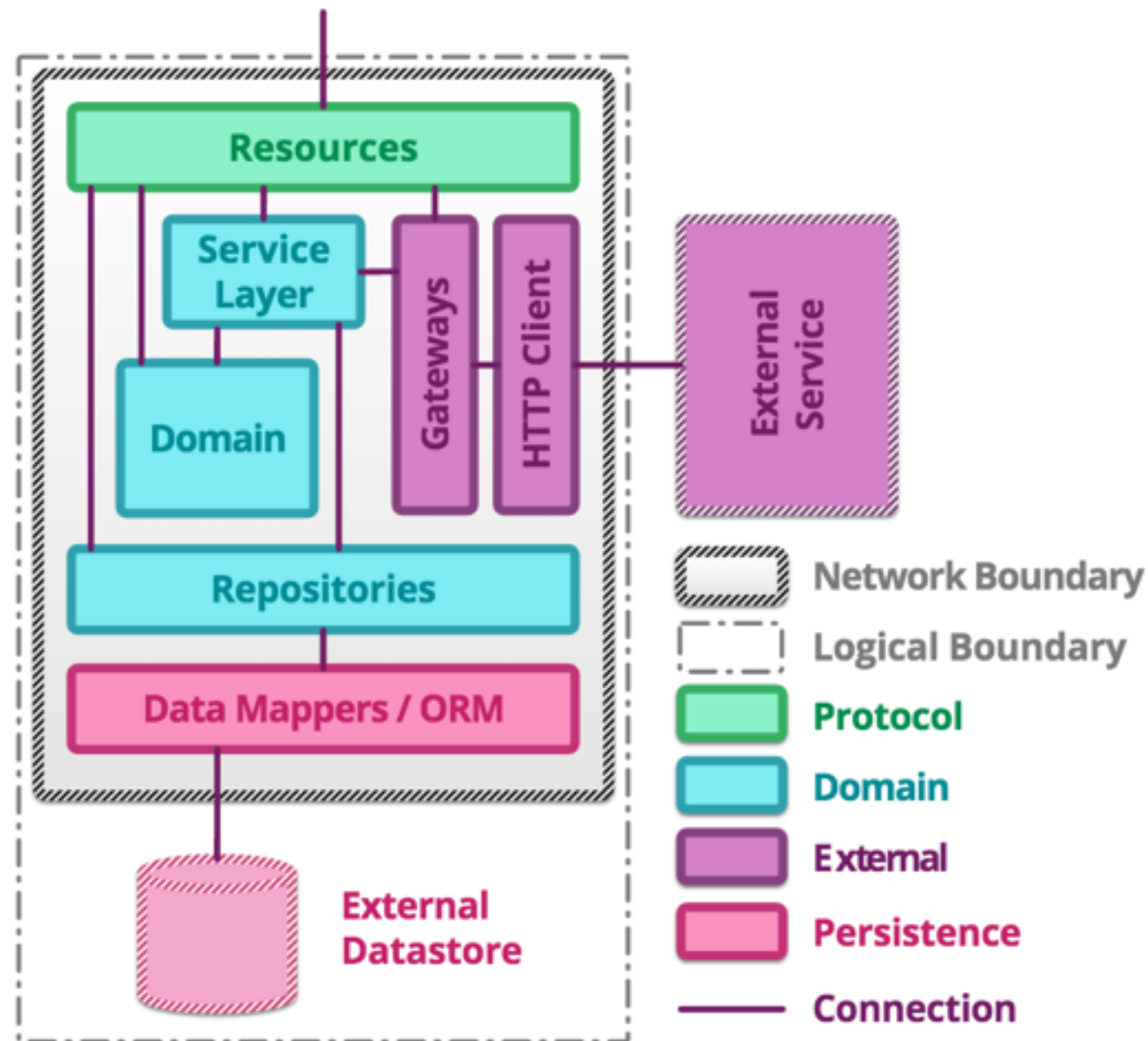
406 Not Acceptable

409 Conflict

# Workshop :: Develop APIs

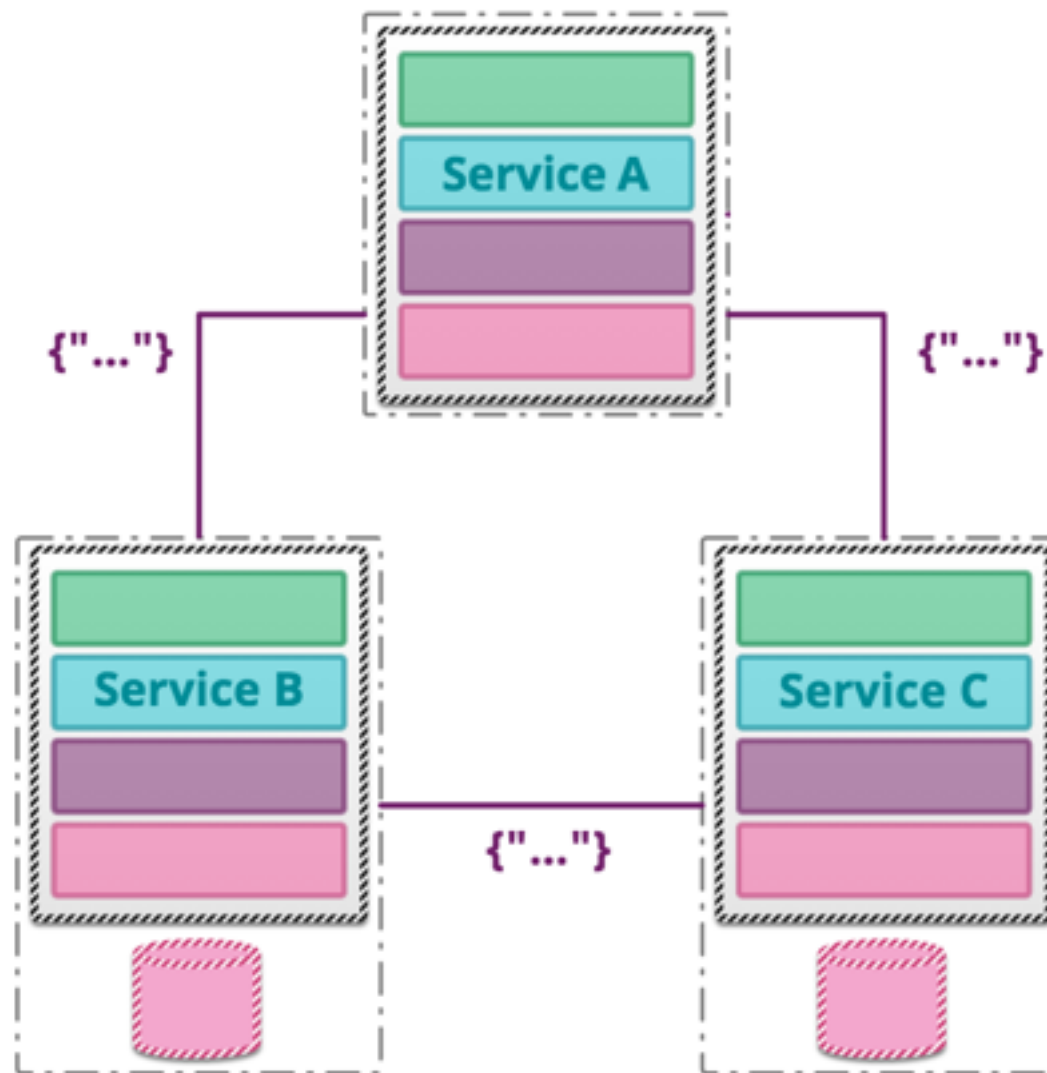
# Testing REST APIs

# Service



<http://martinfowler.com/articles/microservice-testing>

# Multiple service

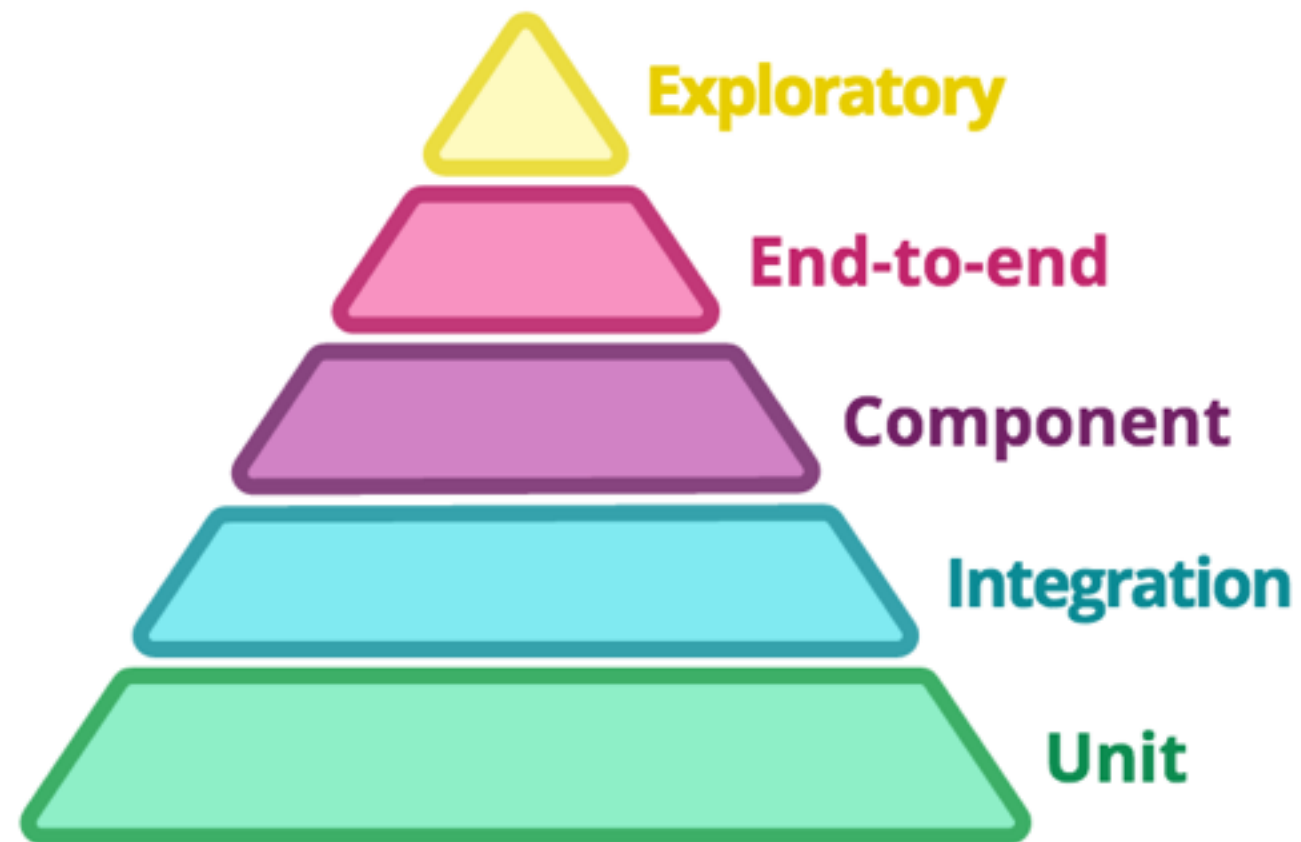


<http://martinfowler.com/articles/microservice-testing>

# How to test the service ?

# Testing

Testing web's APIs isn't easy



<http://martinfowler.com/articles/microservice-testing>

# Testing

**Testing web's APIs isn't easy**

End-to-End testing ?

Contract testing ?

Component testing ?

Integration testing ?

Unit testing ?

<http://martinfowler.com/articles/microservice-testing>



# Workshop :: Testing APIs

<https://github.com/up1/course-rest-in-practice>