

# REST Web Service

By: Mulia Arifandi Nasution



# Topics

- History
- Basic REST concepts
- HTTP Methods (verbs)
- HTTP status code and error handling
- Using HTTP request with REST
- Versioning REST APIs
- Content-Type
- Content Negotiation
- Tools

# History

Roy Fielding defined REST in his 2000 PhD dissertation "Architectural Styles and the Design of Network-based Software Architectures" at University of California, Irvine. He developed the REST architectural style in parallel with HTTP 1.1 of 1996–1999, based on the existing design of HTTP 1.0 of 1996.

# What is REST?

- **R**epresentational **S**tate **T**ransfer
  - Representational
    - Clients possess the information necessary to identify, modify, and/or delete a web resource.
  - State
    - All resource state information is stored on the client.
  - Transfer
    - Client state is passed from the client to the service through HTTP.
- Use HTTP
- Stateless
- Lightweight
- Nouns as URI, verbs as HTTP method
- No “official” standards

# Example of Resourceful

- GET
  - `http://localhost/api/v1/users`
  - `http://localhost/api/v1/posts/1/comments`
- POST
  - `http://localhost/api/v1/users`
  - `http://localhost/api/v1/posts/1/comments`
- PATCH/PUT
  - `http://localhost/api/v1/users/64`
- DELETE
  - `http://localhost/api/v1/users/64`

# Continued ...

A typical HTTP REST URL:

```
http://my.store.com/fruits/list?category=fruit&limit=20
```

protocol

host name

path to a resource

query string

- The **protocol** identifies the transport scheme that will be used to process and respond to the request.
- The **host name** identifies the server address of the resource.
- The **path** and **query string** can be used to identify and customize the accessed resource.

# HTTP Methods (verbs)

- GET
- POST
- PUT/PATCH
- DELETE
- HEAD
- OPTIONS

# GET

- Requests a resource at the request URL
- Should not contain a request body, as it will be discarded.
- May be cached locally or on the server.
- May produce a resource, but should not modify on it.



# POST

- Submits information to the service for processing.
- Should typically return the new or modified resource.

# PUT

- Add a new resource at the request URL.

# DELETE

- Removes the resource at the request URL.

# OPTIONS

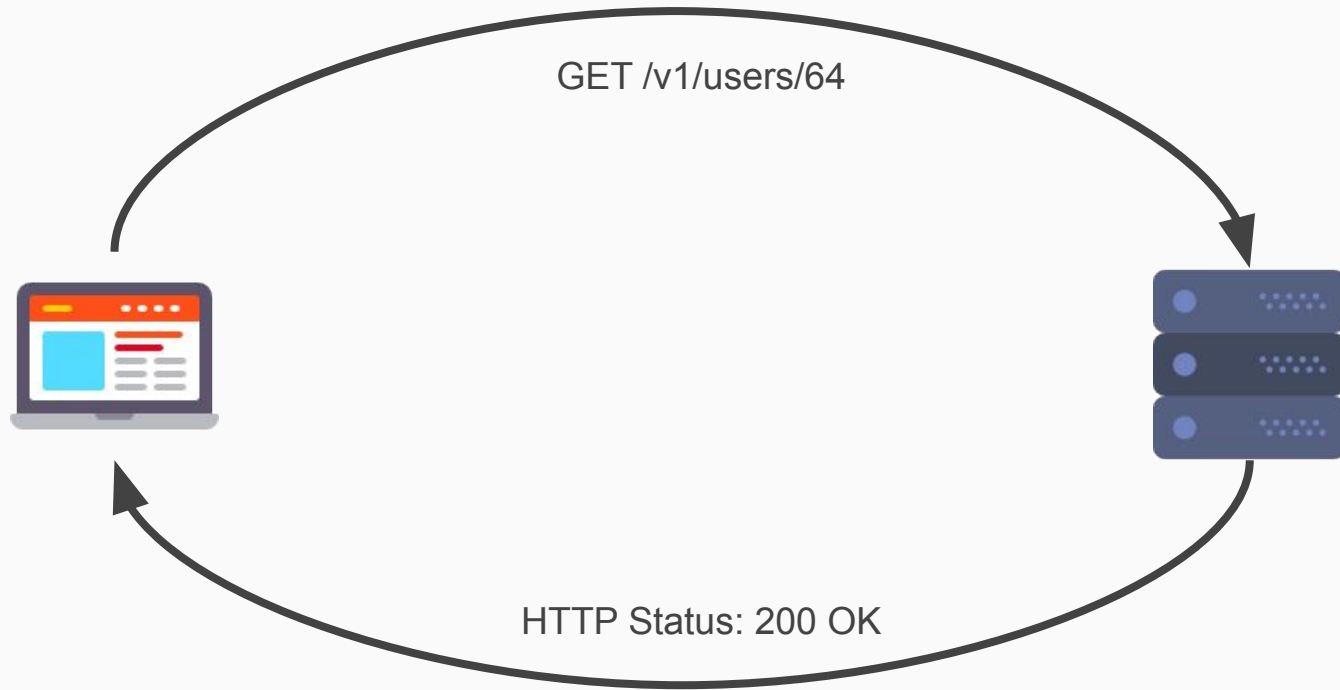
- Indicates which methods are supported.

# HEAD

- Returns meta information about the request URL.

# HTTP Status Code

- **2xx** Success
  - **200** OK
  - **201** Created
- **3xx** Redirection
  - **301** Moved Permanently
  - **304** Not Modified
- **4xx** Client errors
  - **401** Unauthorized
  - **404** Not Found
- **5xx** Server errors
  - **500** Internal Server Error
  - **502** Bad Gateway
  - **504** Gateway Timeout



# Versioning

- Version as prefix
  - /v1/users/64
  - /v2/users/64



# Content Negotiation

- Resources can be served in different representations
  - JSON, XML, HTML, etc
- Content negotiation methods
  - Headers
    - Accept or Content-Type
  - Query parameters
    - GET /v1/users/64?format=json
  - URI extension
    - GET /v1/users/64.json



apiary





Insomnia

curl://



BURP



# Question & Answer

Thank you