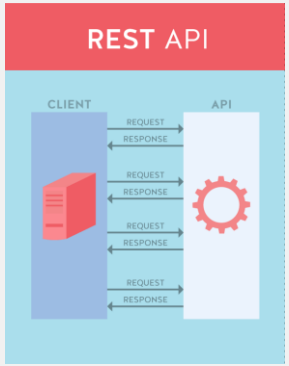
CLOUD COMPUTING APPLICATIONS

Cloud Computing Glue: RESTful Architecture

Prof. Reza Farivar

Representational State Transfer (REST)

- A style of software architecture for distributed hypermedia systems such as the World Wide Web
- Introduced in the doctoral dissertation of Roy Fielding
  - One of the principal authors of the HTTP specification
- The motivation for REST was to capture those characteristics of the Web that made the Web successful
  - URI-addressable resources
  - HTTP
  - Make a request – receive response – display response
- A collection of network architecture principles that outline how resources are defined and addressed
  - Based on HTTP methods to access resources via URL-encoded parameters and the use of JSON or XML to transmit data
  - Request/response between client and server, like a conversation
  - Something is requested, something is done, and then something is sent in return



RESTful API

- Uses HTTP verbs: GET, POST, PUT, PATCH, DELETE
  - Exploits the use of the HTTP beyond HTTP POST and HTTP GET
    - HTTP PUT and DELETE are not even supported in HTML
  - GET is safe (does not change state)
  - GET, PUT and DELETE are idempotent (you can execute them more than once and get the same state change result)
  - Example request:
    - `curl -X POST` https://api.github.com/user/repos
    - `Response:`

```
{
  "message": "Requires authentication",
  "documentation_url": "https://developer.github.com/v3/repos/#create"
}
```

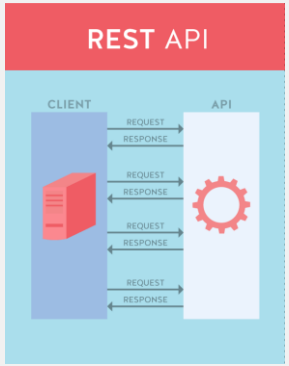


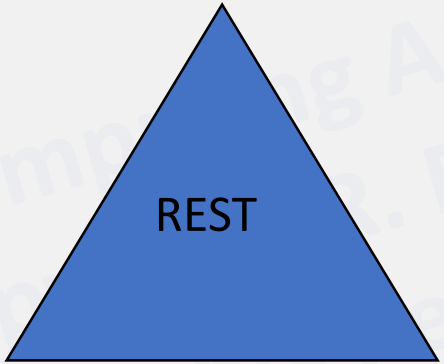
Image from https://nordicapis.com/rest-vs-streaming-apis-how-they-differ/

REST – Not a Standard

- There is no “official standard”, REST is an architectural style
  - JSR 311: JAX-RS: The Java™ API for RESTful Web Services
- But it uses several standards:
  - HTTP
  - URL
  - XML/HTML/GIF/JPEG/etc. (resource representations)
  - Text/xml, text/html, image/gif, image/jpeg, etc. (resource types, MIME types)
- Huge adoption for “Web mashup” applications, operations on When entities
- Many cloud SaaS and PaaS services
- LinkedIn, Twitter,

Main Concepts

Nouns (resources)
unconstrained
i.e., http://example.com/employees/12345



Verbs
constrained
i.e., GET

Representations
constrained
i.e., XML

Resources

- The key abstraction of information in REST is a resource

- A resource is a conceptual mapping to a set of entities
  - Any information that can be named can be a resource: a document or image, a temporal service (e.g., "today's weather in Los Angeles"), a collection of other resources, a non-virtual object (e.g., a person), and so on

- Represented with a global identifier (URI in HTTP)
  - http://www.boeing.com/aircraft/747

Naming Resources

- REST uses URI to identify resources

  - http://localhost/books/
  - http://localhost/books/ISBN-0011
  - http://localhost/books/ISBN-0011/authors

  - http://localhost/classes
  - http://localhost/classes/cs2650
  - http://localhost/classes/cs2650/students

- As you traverse the path from more generic to more specific, you are navigating the data

```
          userinfo          host      port
        ┌──────────────┐ ┌────────┐ ┌──┐
http://john.doe:password@www.example.com:123/forum/questions/?tag=networking&order=newest#top
└──┘   └──────────────────────────────────┘└──────────────┘└────────────────────────────┘└──┘
scheme          authority                     path                  query              fragment
```

Verbs

- Represent the actions to be performed on resources

- HTTP GET
- HTTP POST
- HTTP PUT
- HTTP DELETE
- HTTP PATCH

HTTP GET

- How clients ask for the information they seek
- Issuing a GET request transfers the data from the server to the client in some representation

- GET http://localhost/books
  - Retrieve all books
- GET http://localhost/books/ISBN-0011021
  - Retrieve book identified with ISBN-0011021
- GET http://localhost/books/ISBN-0011021/authors
  - Retrieve authors for book identified with ISBN-0011021

HTTP POST, HTTP PUT

- HTTP POST creates a resource
- HTTP PUT updates a resource

- POST http://localhost/books/
  - Content: {title, authors[], …}
  - Creates a new book with given properties

- PUT http://localhost/books/isbn-111
  - Content: {isbn, title, authors[], …}
  - Updates book identified by isbn-111 with submitted properties

HTTP DELETE

- Removes the resource identified by the URI

- DELETE http://localhost/books/ISBN-0011
  - Delete book identified by ISBN-0011

Representations

- How data is represented or returned to the client for presentation
- Two main formats:
  - JavaScript Object Notation (JSON)
  - XML
- It is common to have multiple representations of the same data
- XML

```
<COURSE>
        <ID>CS2650</ID>
        <NAME>Distributed Multimedia Software</NAME>
</COURSE>
```

- JSON

```
{course
        {id: CS2650}
        {name: Distributed Multimedia Software}
}
```

Architecture Style