REST

Exploring REST API’s

April 30, 2021

Manulife

Agenda

What is a API?

- What is an API?
- What isn't necessarily an API?
- Types of APIs

What is a REST API?

- Message vs Transport protocol
- Commands vs Resources
- What is REST?
- The six constraints of REST
- Exercise

Doctors and Patients

- Problem statement
- REST Implementation
- Potential pitfalls of REST
- Documentation with Swagger
- Swagger editors
- Excercise

What is an API?

What is an API?

API ?

- Interface = The portion of an application that is exposed to the **outside world** for users to interact with

- UI = User **Interface**
  - Audience: **Users**
- API = Application Programming **Interface**
  - Audience: Developers, other application
  - A collections of functions or methods with known signatures and data models

Manulife

What is an API?

Different types of API's

- Libraries and Frameworks
  - A software library
  - Ex. Java/JDK API, React API
- Operation Systems (OS)
  - Specifies the interface between and application and the operating system
  - Ex. WinAPI, POSIX
- Web Services
  - The interface for one application to exchange data with another over the web
  - Ex. SOAP, REST

Manulife

What is an API?

Web Services / Web API

- For the remainder of this course, we'll be focused on web services / web API's

https://towardsdatascience.com/api-management-platform-data-science-projects-43120558a000

Manulife

What is a REST API?

What is a REST API?

Message Protocol vs Transport Protocol

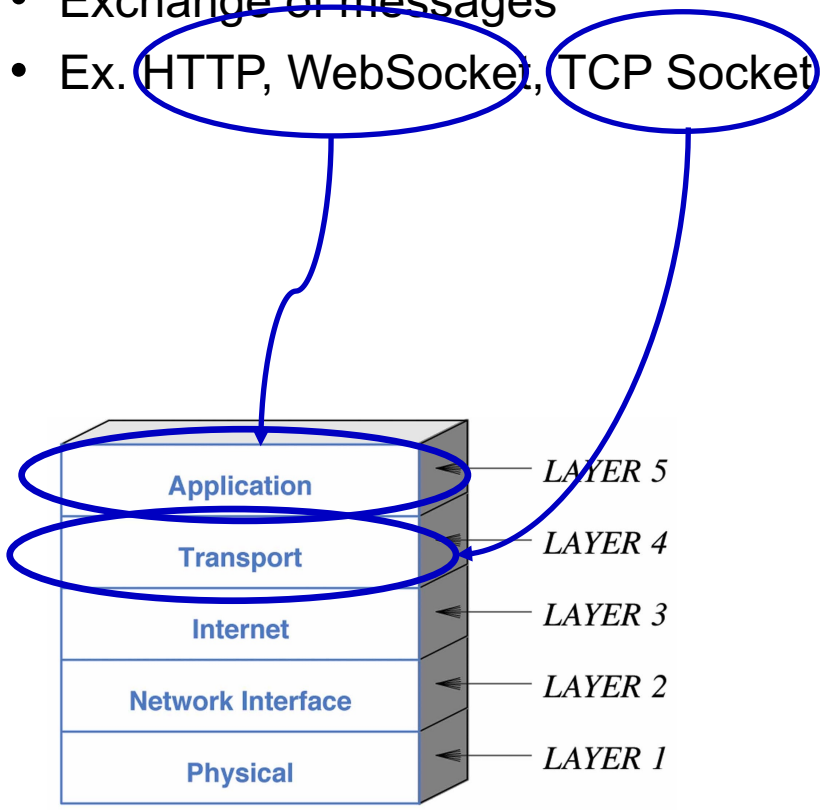
Message Protocol

- Structure and format of messages
- Ex. SOAP (Simple Object Access Protocol)

```
<?xml version="1.0"?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
  soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">

  <soap:Header>
    <m:Trans xmlns:m="https://www.w3schools.com/transaction/"
    soap:actor="https://www.w3schools.com/code/">234
    </m:Trans>
  </soap:Header>

  <soap:Body>
    <m:GetPriceResponse xmlns:m="https://www.w3schools.com/prices">
      <m:Price>1.90</m:Price>
    </m:GetPriceResponse>
  </soap:Body>

</soap:Envelope>
```

Transport Protocol

- Exchange of messages
- Ex. HTTP, WebSocket, TCP Socket



Manulife

What is a REST API?

What is REST?

- **Re**presentational **S**tate **T**ransfer
- An architectural style for sharing data between applications
- Comprised of 6 architectural constraints
- Implemented over HTTP

REST was defined by Roy Fielding, a computer scientist. He presented the REST principles in his PhD dissertation in 2000.

What is a REST API?

What is REST?

Representational
state **t**ransfer

REST is comprised of the following:

- Services
- Resources
- Representations
- Verbs

Example

- Service: Doctor Profile
- Resource: Doctor
- Representation: {Id: 123, Name: 'Dr. Phil'}
- Verb: GET

Manulife

What is a REST API?

The 6 architectural constraints of REST

Uniform Interface

- Define the interface between the client and the server
- Simplifies and decouples architecture
- Restful design
  - HTTP verbs (e.g. GET, PUT, POST, DELETE)

Stateless

- The server does not remember anything about the user using the API
- Each individual request contains all the information the server needs to perform the request and return a response, regardless of other requests made by the same API user

Manulife

What is a REST API?

The 6 architectural constraints of REST

Client-Server Separation

- The client and server act independently, each on its own
- Interaction between them is only in the form of requests, initiated by the client only
- The server only responds in reaction to a request from a client

Cacheable

- Server response contains information about whether or not the data is cacheable
- Client can avoid requesting the same data again and again
- Client should know when the current version of the data expires

Manulife

What is a REST API?

The 6 architectural constraints of REST

Layered System

- Between the client who requests a resource and the server who responds, there might be a number of servers in the middle
- These servers might provide a security layer, caching layer, load-balancing layer or other functionality
- Additional layers should not affect request or response

Code-on-demand (optional)

- Client can request code from the server, usually in the form of a script
- The client can then execute that code

What is a REST API?

Exercise

- Make a REST API with Express
- GET /home → { response: “Hello World!” }

Manulife

Doctors and Patients

Doctors and Patients

Problem Statement

- You are web developer building a healthcare product
- You need to retrieve patient data from the server
- What are some of the typical ways to retrieve data in this scenario?

Manulife

Potential Solutions

Custom protocol implementation

Direct database access

SOAP

REST APIs & similar implementations

	Custom protocol implementation	Direct database access	SOAP	REST APIs & similar implementations
Avantages	• Full control over implementation • Security can be integrated into existing systems	• SQL is easy & ubiquitously known • Flexibility of a query engine	• Very descriptive API (WSDL) • Utilizes secured authentication	• More detailed discussion to follow
Inconvénients	• Considerable time spend • Potential for security breaches	• Tight coupling between client & database • Potential for security breaches	• Tight coupling between client & database • Lack of supported data formats	

Doctors and Patients

REST Implementation

- Supports any hypertext format (JSON, XML etc.)
- CRUD Operations (GET, PUT, POST, DELETE)
- APIs are independent of the client
- Fewer resources in transport as well as description
- Caching utilizes less network resources

Doctors and Patients

API path naming conventions

Verbs

- /getDoctors

Nouns

- More specifically, plurals
- /doctors

Filter

- Use ?
- GET /doctors?lastname=smith&city=waterloo

Manulife

Doctors and Patients

HTTP Verbs

GET	Retrieve a representation of a resource
PUT	Update an existing entity
DELETE	Delete an entry (could be async)
POST	Create a new entity

Manulife

Doctors and Patients

HTTP Status Codes

1XX – Information

- Ex. 101 Switching Protocols (HTTP → WebSocket)

2XX – Success

- Ex. 200 OK, 201 Created, 204 No Content

3XX – Redirect

- Ex. 301 Moved Permanently

4XX – Client Error

- Ex. 400 Bad Request, 401 Unauthorized, 404 Not Found

5XX – Server Error

- Ex. 500 Internal Server Error

Doctors and Patients

Patients	Visits	Doctors
• ID • Name	• Patient ID • Doctor ID • Completed	• ID • Name

Simplified REST API

```
var express = require("express");
var app = express();
var router = express.Router();

router.route("/patients/:id").get(function(req, res) {
  var patient = db.getPatient(req.body.id);
  res.json(patient);
});

router.route("/doctors/:id").get(function(req, res) {
  var doctor = db.getDoctor(req.body.id);
  res.json(doctor);
});

router.route(“/visits/:patientId”).get(function(req, res) {
  var visits = db.getVisits(req.body.patientId);
  res.json(visits);
});
```

…but where might our solution struggle?

Manulife

Doctors and Patients

Potential pitfalls of REST

- REST APIs are synchronous & hence resource intensive
- REST offers fewer verbs to operate on (e.g. no upsert functionality)

- Loses relationships between entities

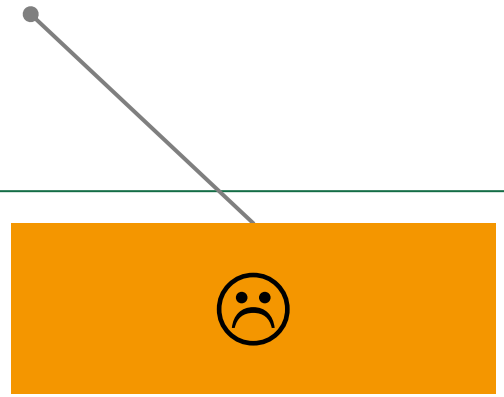
- Often misused losing scalability

Manulife

The wrong approach

Popular “REST” API

```
router.route("/patientsWithDoctors/:patientId").get(function(req, res) {
  var result = {
    patient: {},
    doctors: []
  };
  result[‘patient’] = db.getPatient(req.body.patientId);
  var visits = db.getVisits(req.body.patientId);
  visits.forEach(function(visit) {
    result.doctors.push(db.getDoctor(visit.doctor_id));
  });
  res.json(result);
});
```



- Does not conform to standard
- Increases implementation complexity
- Endpoints tailored to specific use case – restricts reuse
- Any new feature requires editing an endpoint, resulting in potential breaking changes

Doctors and Patients

Documentation with Swagger

- Swagger is a framework for describing an API using a common language everyone can understand
  - Think of it like a blueprint for a house

- Comprehensible for developers and non-developers
- Both human readable and machine readable
- Easily adjustable

Manulife

Sample Swagger file

Header

```yaml
swagger: '2.0'
info:
  title: polaris-demo1-service
  description: Polaris Demo 1 Service
  version: 1.0.0
schemes:
  - https
basePath: /
produces:
  - application/json
```

API Paths

```yaml
paths:
  '/demo1/v1/{name}':
    post:
      operationId: sayHello
      parameters:
        - name: name
          in: path
          description: name to say hello
          required: true
          type: string
        - in: body
          name: echoMessage
          required: false
          description: data for query
          schema:
            $ref: '#/definitions/EchoData'
      responses:
        200:
          description: "OK"
          schema:
            $ref: '#/definitions/DemoResponse'
        404:
          description: "Resource Not Found"
```

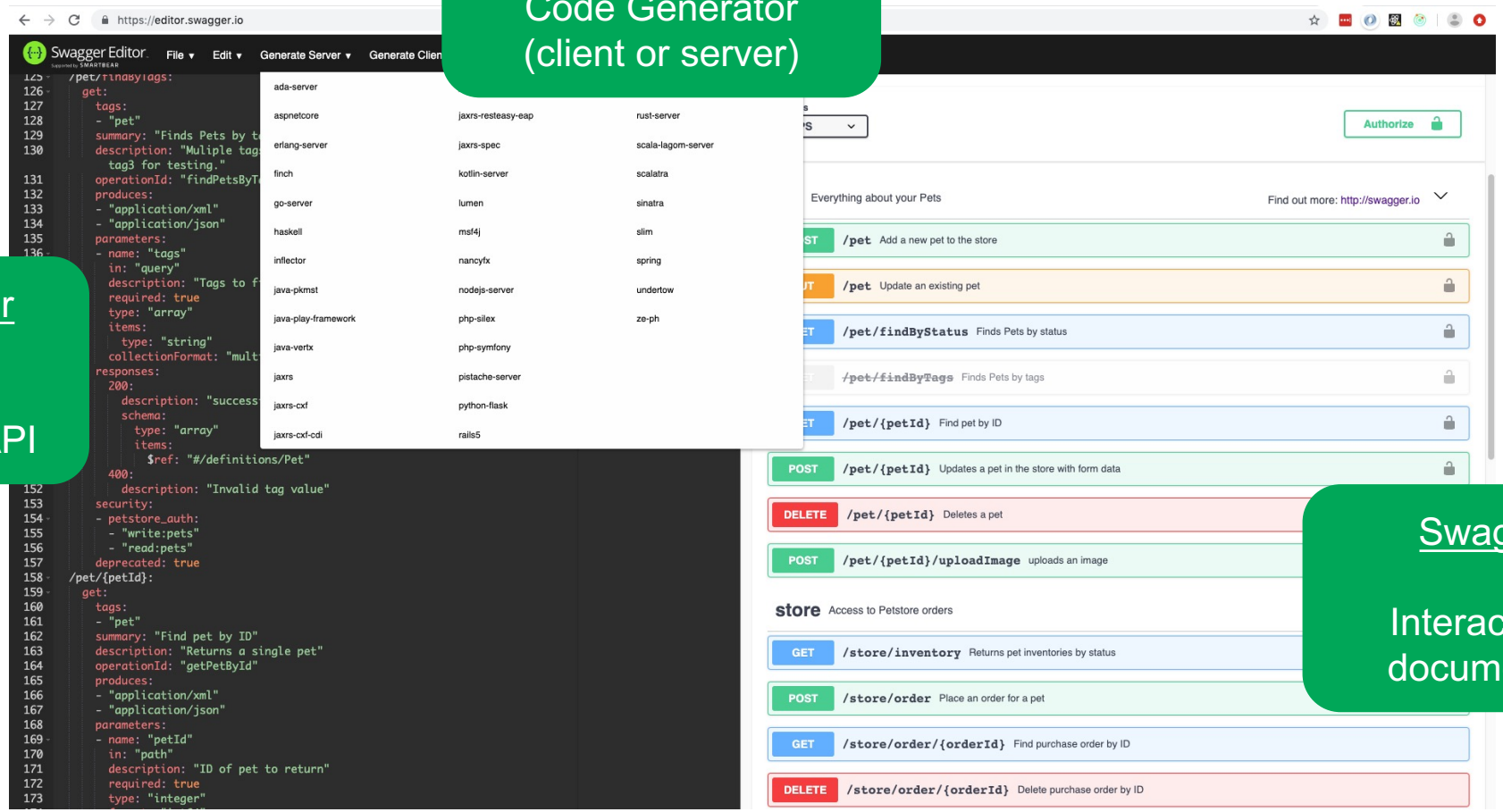
Definitions

```yaml
definitions:
  EchoData:
    type: object
    properties:
      message:
        type: string
        example: "Hello, how are you?"
  DemoResponse:
    type: object
    properties:
      result:
        type: string
      errors:
        $ref: '#/definitions/Errors'
  Errors:
    type: object
    description: error model for exception
    properties:
      errorList:
        type: array
        items:
          type: string
```

Swagger Editor

Swagger Codegen

Code Generator
(client or server)



Swagger Editor

Editor for
designing and API

SwaggerUI

Interactive API
documentation

Manulife

Doctors and Patients

Two approaches to Swagger

1) Top-Down
   - Design-first
   - Swagger Editor → Swagger/OpenAPI Spec → Swagger Codegen

2) Bottom-Up
   - Generate Swagger documentation for an existing API

Manulife

Doctors and Patients

Exercise

Manulife


Manulife