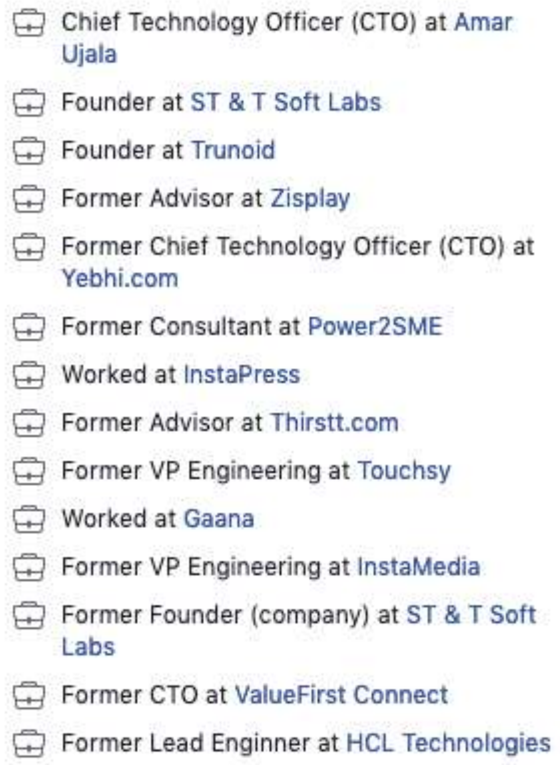
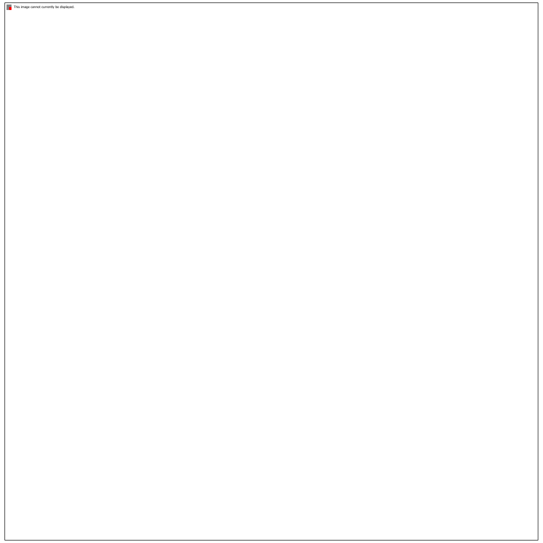
Demystifying API

Ashok Gautam



- Chief Technology Officer (CTO) at Amar Ujala
- Founder at ST & T Soft Labs
- Founder at Trunoid
- Former Advisor at Zisplay
- Former Chief Technology Officer (CTO) at Yebhi.com
- Former Consultant at Power2SME
- Worked at InstaPress
- Former Advisor at Thirstt.com
- Former VP Engineering at Touchsy
- Worked at Gaana
- Former VP Engineering at InstaMedia
- Former Founder (company) at ST & T Soft Labs
- Former CTO at ValueFirst Connect
- Former Lead Enginner at HCL Technologies



Agenda

API Introduction

RESTFul paradigm

Design, Development & Challenges

Best practices

Tools

Resources

Q&A

API: Introduction

What is API

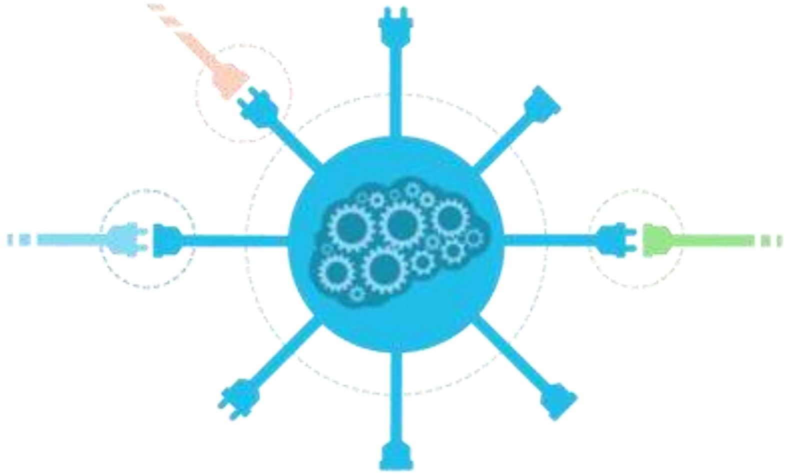
Evolution

Benefits

Introduction

API stands for Application Programming Interface. API is a set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.

API is a software intermediary that allows two applications to talk to each other.



Evolution

APIs have existed for a long time. Since the first computer programs were written, APIs have been providing “contracts” for information exchange between programs.

OS APIs

Platform APIs

Application APIs

Web APIs

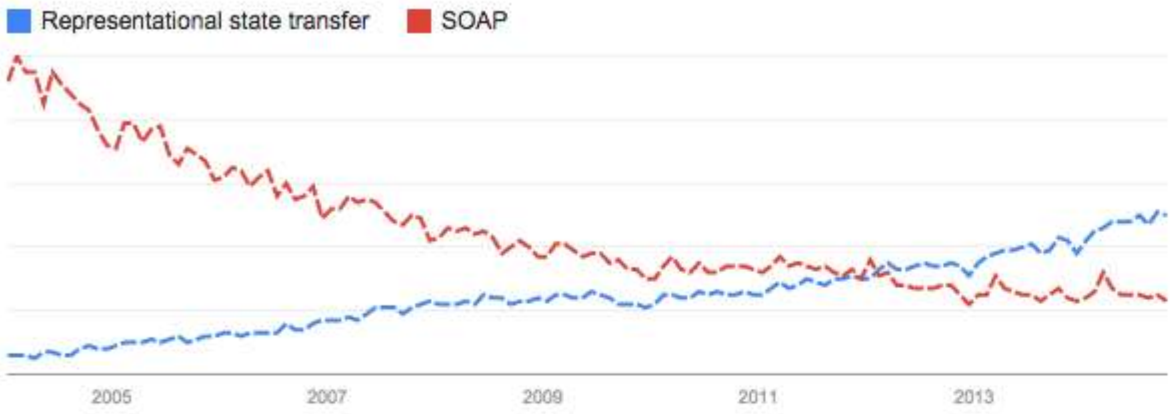
XML-RPC

SOAP

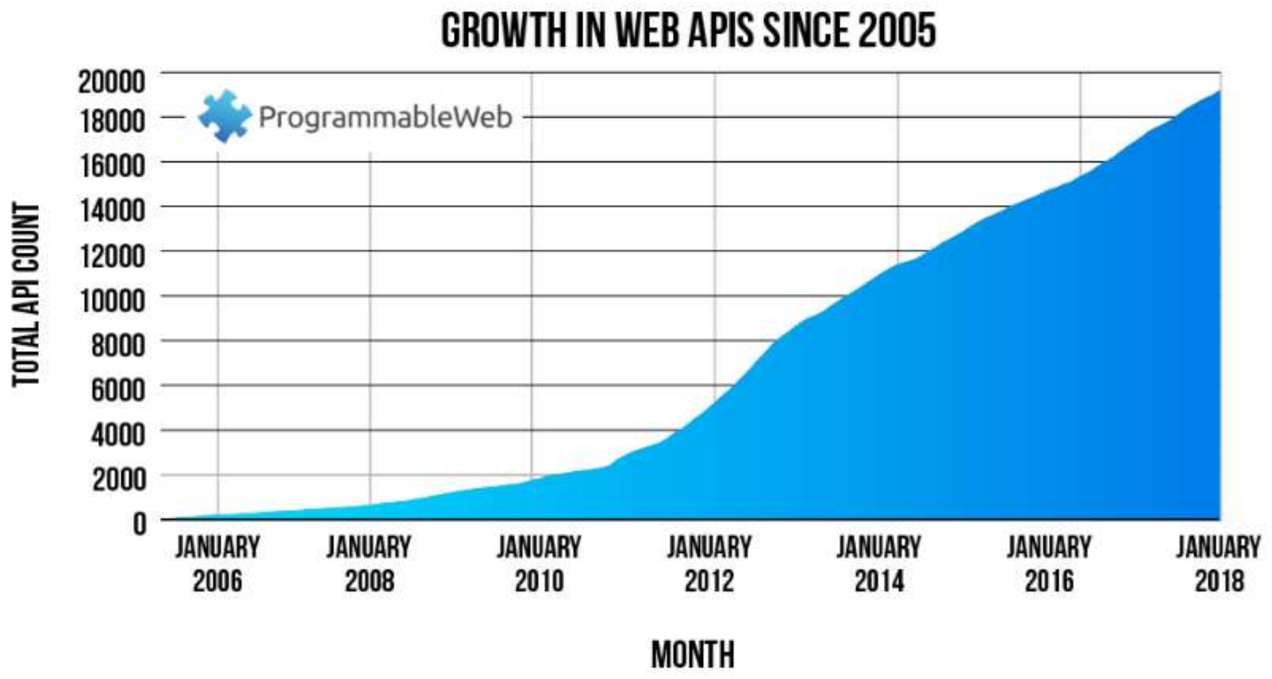
RESTful

GraphQL

Interest over time. Web Search. Worldwide, 2004 - present.



Google™ View full report in Google Trends



The growth over time of the ProgrammableWeb API directory to more than 19,000 entries

Why should you have API

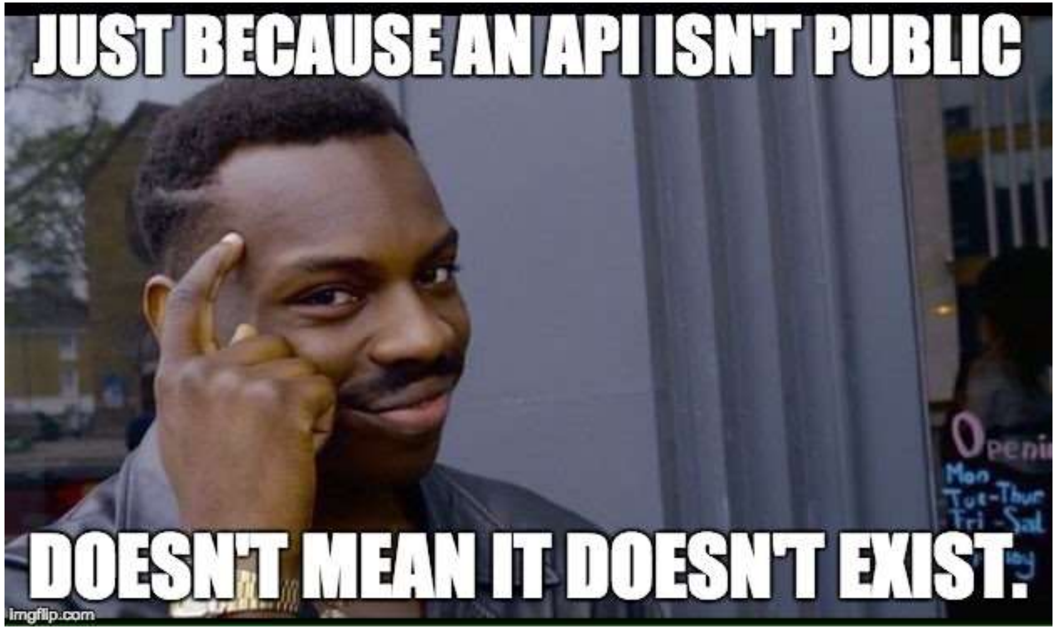
Efficiency

Flexibility

Integrations

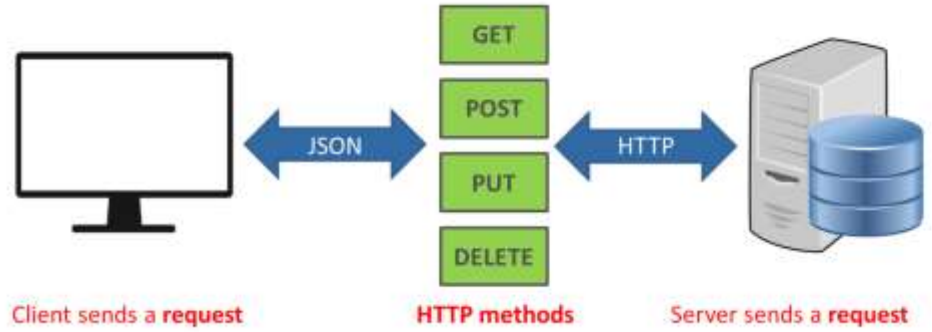
Security

Metered Usages


JUST BECAUSE AN API ISN'T PUBLIC
DOESN'T MEAN IT DOESN'T EXIST.
imgflip.com

RESTful

A RESTful API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data.



RESTful

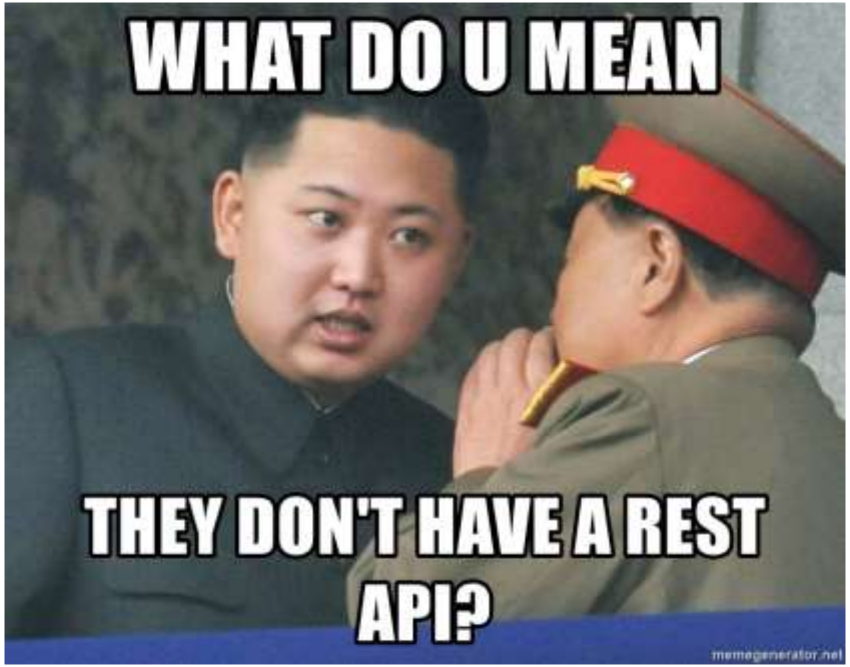
Uniform interface

Client–server

Stateless

Cacheable

Layered system



https://www.bitnative.com/2012/08/26/how-restful-is-your-api/

Design, Development & Challenges

Design

OAS https://swagger.io/resources/open-api/

Use nouns and NOT the verbs

Use of right HTTP methods

Use Plurals

Use parameters

Use proper HTTP codes

Versioning

Use Pagination

Supported Formats

Use Proper Error Messages

https://hackernoon.com/restful-api-design-step-by-step-guide-2f2c9f9fcdbf

Development

Express.js

HAPI.JS

LoopBack

Swagger

Flask

Spring Boot

Kong

APIGEE

Swagger

Mulesoft

FireBase

Postman

JMeter

Katalon

Hello World

https://medium.com/@purposenigeria/build-a-restful-api-with-node-js-and-express-js-d7e59c7a3dfb

```
import express from 'express';import db from './db/db';

  // Set up the express app
  const app = express();
  // get all todos
  app.get('/api/v1/hello', (req, res) => {
   res.status(200).send({
     success: 'true',
     message: 'Hello World',
     todos: db
   })});
  const PORT = 5000;
  app.listen(PORT, () => {
   console.log(`server running on port ${PORT}`)
  });
```

```
'use strict';
const Hapi=require('hapi');
// Create a server with a host
and port
const server=Hapi.server({
   host:'localhost',
    port:8000
});
// Add the route
server.route({
    method:'GET',
    path:'/hello',
    handler:function(request,h) {
       return'hello world';
    }});
```

```

// Start the server
const start = async function() {
   try {
       await server.start();
   }
   catch (err) {
        console.log(err);
        process.exit(1);
    }
   console.log('Server running
at:', server.info.uri);
};
start();
```

Kong

- Cloud-Native
- Dynamic Load Balancing
- Hash-based Load Balancing
- Circuit-Breaker
- Health Checks
- Service Discovery
- Serverless
- WebSockets
- OAuth2.0
- Logging
- Security
- Syslog
- SSL
- Monitoring
- Forward Proxy
- Authentications
- Rate-limiting.
- Transformations
- Caching
- CLI
- REST API
- Geo-Replicated
- Failure Detection & Recovery
- Clustering
- Scalability
- Performance
- Plugins

Challenges

Security

Authentication & Authorization

Rate Limit

Scalability

Security

HTTPS

Access Control

Restrict HTTP methods

Input validation

Validate content types

Management endpoints

Error handling

Audit logs

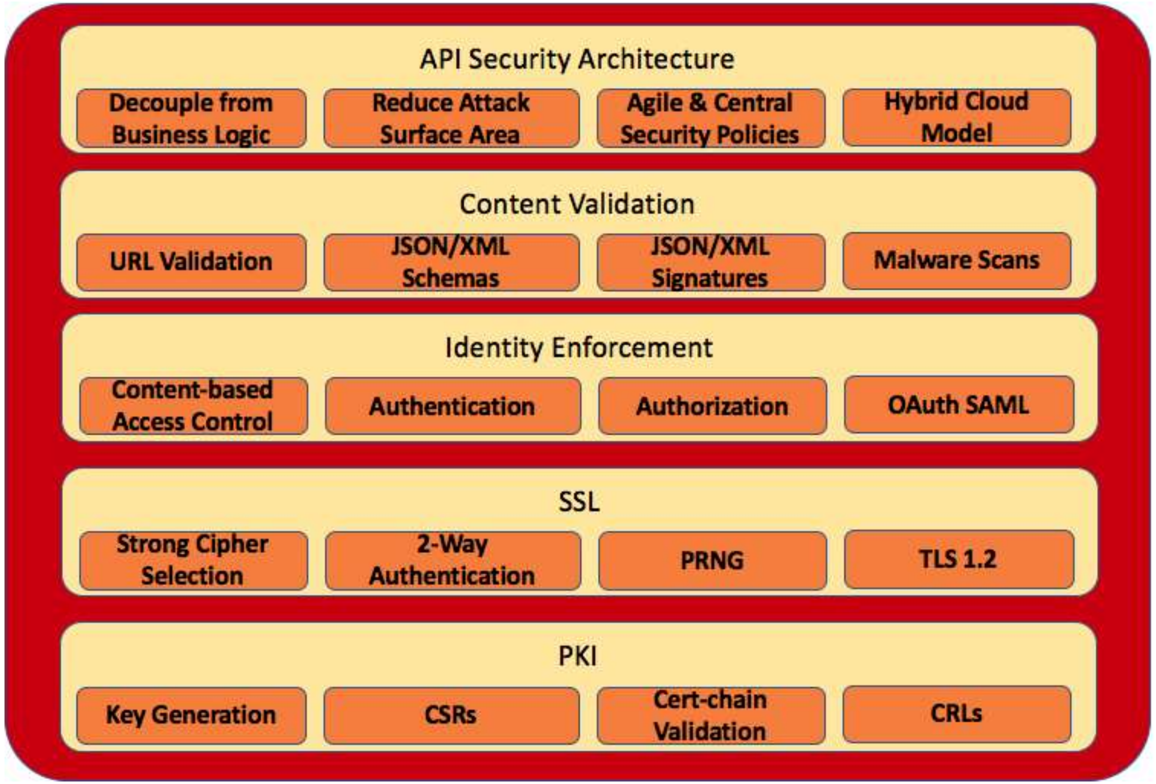
Security headers

CORS

Sensitive information in HTTP requests

- Parameters Exploitation
- Identity Theft
- Abusing authorization system
- Man-In-The-Middle
- DOS & DDOS

Security


API Security Architecture
Decouple from Business Logic
Reduce Attack Surface Area
Agile & Central Security Policies
Hybrid Cloud Model
Content Validation
URL Validation
JSON/XML Schemas
JSON/XML Signatures
Malware Scans
Identity Enforcement
Content-based Access Control
Authentication
Authorization
OAuth SAML
SSL
Strong Cipher Selection
2-Way Authentication
PRNG
TLS 1.2
PKI
Key Generation
CSRs
Cert-chain Validation
CRLs

Authentication & Authorization

API keys

OAuth access tokens

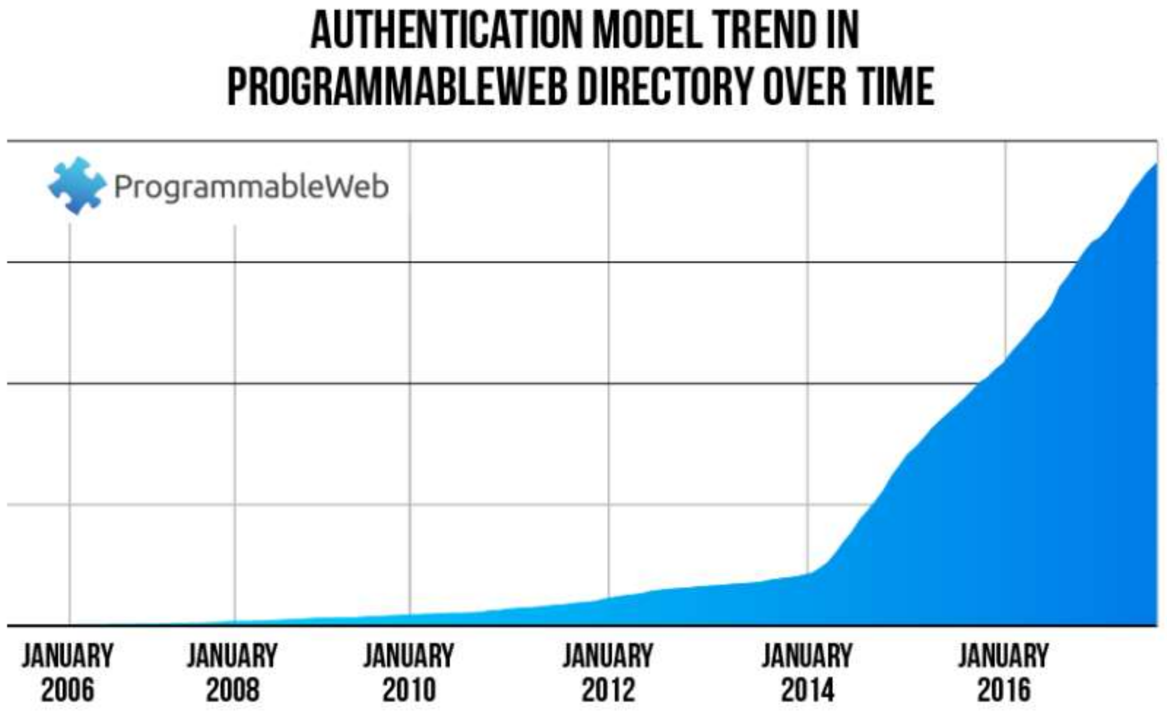
JSON Web Tokens

https://zapier.com/engineering/apikey-oauth-jwt/

https://blog.restcase.com/restful-api-authentication-basics/

- Use API keys if you expect developers to build internal applications that don't need to access more than a single user's data.
- Use OAuth access tokens if you want users to easily provide authorization to applications without needing to share private data or dig through developer documentation.
- Use JWT in concert with OAuth if you want to limit database lookups and you don't require the ability to immediately revoke access.

API : Authentication



Type	Count
API Key	3894
Token	1489
HTTP Basic Auth	1153
Unspecified	1128
OAuth 2	920
App ID	256
OAuth 1	232
Other/Custom	226
Shared Secret	143
Session	55
SAML	14
WS_Security	8

Rate Limit

Leaky Bucket

Fixed Window

Sliding Log

Sliding Window

express-rate-limit

hapi-ratelimiter

flask-limiter

User rate limits

IP/Network rate limits

Server rate limits

Regional data limits

Resource specific rate limits

Dynamic rate limits

Rate Limit

```
const rateLimit = require("express-rate-limit");

app.enable("trust proxy"); // only if you're behind
a reverse proxy (Heroku, Bluemix, AWS ELB, Nginx,
etc)

const apiLimiter = rateLimit({

  windowMs: 15 * 60 * 1000, // 15 minutes

  max: 100

});

app.use("/api/", apiLimiter);
```

Scaling

CDN	Cloudflare/ Cloudfront/Akamai
Application level caching	Varnish/NGINX
Database Caching	Redis/Memcache

Resources

https://hackernoon.com/restful-api-design-step-by-step-guide-2f2c9f9fcdbf
https://www.apiacademy.co/lessons/2015/04/api-design-101-api-design-basics
https://docs.microsoft.com/en-us/azure/architecture/best-practices/api-design
https://blog.mwaysolutions.com/2014/06/05/10-best-practices-for-better-restful-api/