API Designing

API Signature, Web-Services, Security, & Frameworks

Background

- API - Application Programming Interface
  - An interface that enables a software program to interact with other software
  - E.g. malloc(), printf(), strcpy()
- Web API/Service design models - SOAP, **REST**, GraphQL, gRPC, XML-RPC, Webhooks, etc..
- Protocols - HTTP1, HTTP2
  - HTTP/2 introduces a new binary framing layer that breaks backward compatibility with HTTP/1.x clients and servers. Hence, the versioning jump to 2.0.
  - It is a simple design choice. HTTP/1.x traditionally used newline-delimited text to transfer data. It is slower, larger, and more error-prone compared to binary. Hence, HTTP/2 made the switch to binary format in the transfer layer.
  - Ref: https://blog.usejournal.com/migrating-your-rest-apis-to-http-2-why-and-how-8caee7d798fb

API (Web API)

- API: Now everyone refer it as **Web** API
- A software interface that allows two applications to interact with each other without any user intervention.
- APIs provides product or service to communicate with other products and services without having to know how they're implemented.

Web Services

- A software system designed to support interoperable machine-to-machine interaction over a network
- The term Web services describes a standardized way of integrating Web-based Application using the XML, SOAP, Wsdl and UDDI open Standard

- Ref:
  - https://wiki.python.org/moin/WebServices
  - https://medium.com/@umerfarooq_26378/web-services-in-python-ef81a9067aaf

API Vs Web Service

- Overlapping tech terms that regularly get confused
  - Why is a web API not a web service?
  - What do APIs and web services have in common?
- API: An interface that allows you to build on the data and functionality of another application, send data back and forth using HTTP requests, these requests often return textual data in the form of a JSON or XML response
- Web Service: Any piece of software that makes itself available over the Internet and standardizes its communication via XML encoding. A client invokes a web service by sending a request (usually in the form of an XML message), and the service sends back an XML response
- Ref
  - https://nordicapis.com/what-is-the-difference-between-web-services-and-apis/
  - https://blogs.mulesoft.com/dev/api-dev/apis-versus-web-services/

Let’s talk more about **REST** APIs

REST?

- Representational State Transfer – Most popular design model for Web APIs
- Entities (“resources”) = URLs
- Actions = HTTP commands – GET, POST, PUT, DELETE
- Resources are self-descriptive
- No hidden server-side state

Why REST?

- Separation between the client and the server
- Visibility, Reliability and Scalability
- Anytime, Anywhere, Any Device
- Business Agility
- Cloud Native
- Code on Demand
- Stateless
- Cacheable

API Signature

- URL (Uniform resource locator)
- Resource
  - DB resource, file resource, anything which is server side
- Status Codes
  - 1xx Information, 2xx Success, 3xx Redirects, 4xx Client Error, 5xx Server Error
- HTTP Verbs
  - GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE, PATCH
- HTTP Headers
- Query Strings (Search, Filters, Pagination, Partial Response, Sort)
- Payloads/Body
- Extra
  - URL structure: Nouns vs Plurals, Hierarchical, Versioning
  - Subdomain/Domain

Web Security

- Secure Protocol
  - HTTPS - TLS/SSL
- Auth Types
  - Basic
  - Token
  - Session
  - OAuth2
  - SSO/SAML2.0

Popular REST Frameworks

- Python
  - Django + DRF lib
  - Flask
  - Tornado
- Java
  - Spring
  - Dropwizard
- JavaScript
  - Express.js - based on Node.js
  - LoopBack
- Golang
  - gorilla/mux
  - gRPC based

Frameworks Comparison

- https://www.slant.co/topics/1397/~best-web-frameworks-to-create-a-web-rest-api
- https://www.techempower.com/benchmarks/#section=test&runid=7464e520-0dc2-473d-bd34-dbdfd7e85911&hw=ph&test=query&l=zijzen-7

Frameworks Use-Cases

Could be based on..

- Based on Developer’s Expertise / Ease of Use
- Projects Size (Small / Big)
- Monolithic / Microservices
- Fast Development
- High-Performance
- Synchronous / Asynchronous
- ....

Q & A

Feedback is welcome..