Web Services & APIs

Lecture 8

Recall

- **HTTP** (Hyper Text Transfer Protocol)
  - Protocol for transferring data over internet
- **XML/JSON**
  - Standard format for sending data
- **Web Apps**
  - Programs that are accessed over internet

Web Service

- A Web Service is a way two machines communicate **over a network**
- When a client computer requests a resource over a network, the Web Service returns that resource
- This resource can be an *XML file, audio, image etc.*

API

- An **API (Application Programming Interface)** is a set of code definitions and protocols that allow one application to communicate with another

- Can you think of a local example of this?

API

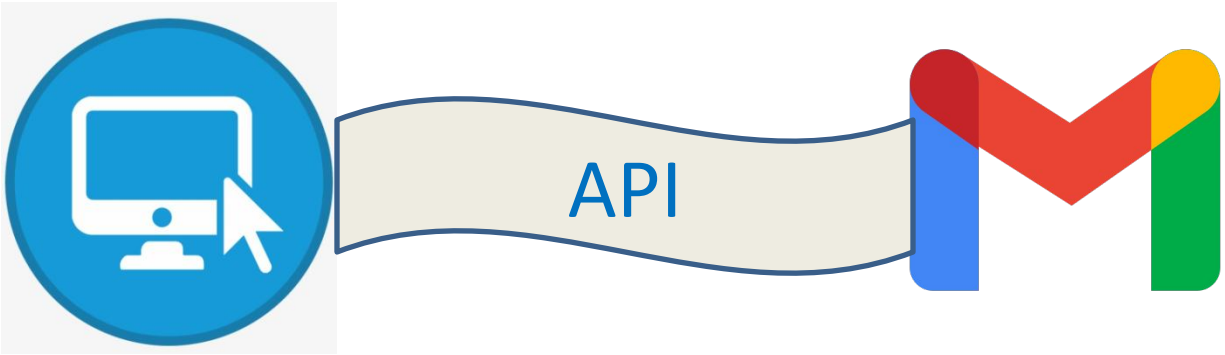
- An API (Application Programming Interface) is a set of code definitions and protocols that allow one application to communicate with another

- Can you think of a local example of this?
  - .h files in C/C++

Web API

- A Web API is one which is accessible over the internet

- For example, someone signing up for a website using Gmail



Web Service vs API

- Both are ways for two computers/applications to communicate
- Unlike Web Service, an API does not necessarily have to be accessible over network
- **Conclusion:** All Web Services are APIs, but not all APIs are Web Services

REST API

REST API

- **REST** (REpresentative State Transfer) is a standardized architectural style of creating a Web API

- REST requires that HTTP methods are used by applications to make a request over the network

REST API

- In order for a Web API to be a RESTful API, it must conform to some rules:
  a) Client-Server model
  b) HTTP as common protocol for sending/receiving requests
  c) All calls with REST API must be stateless (every request is independent of one another)
  d) Layered systems
  e) Caching of resources

REST API

- A REST request from client to server usually contains the following:
  1) **URL Path** (e.g. https://api.myapp.com/user)
  2) **HTTP method** (GET, PUT, POST, PATCH, DELETE)
  3) **Header** *(optional)* (e.g. authorization credentials)
  4) **Parameters** *(optional) – values that alter how resources will be returned*
  5) **Body** *(optional) – contains data that needs to be sent to the server*

Example

- You want to see what your friend posted on Facebook

Example

- You want to see what your friend posted on Facebook

  i. Your Facebook app (client) will make a request to Facebook server to request your friend's profile
  ii. This request will be a GET request to user's endpoint
  iii. The parameters in request are friend's account ID

Example

- You want to see what your friend posted on Facebook

- Your Facebook app (client) will make a request to Facebook server for your friend's profile. This would be a GET request to /users endpoint and user_id of your friend is included in parameter

  - HTTP Method: GET
  - URL: https://api.facebook.com/u/users/
  - Parameters: user={user_id}

Example

- Similarly, to create content on the platform
  - HTTP Method: POST
  - URL: https://api.facebook.com/u/media/
  - Parameters: post={text}&user={your_user_id}

Benefits of REST APIs

- Provides a standardized methodology for making API requests
- Can send data in many formats (e.g. XML, JSON, XML, plain text)
- Only needs minimum bandwidth

Practice

- You can check out some free APIs on: https://rapidapi.com/hub

- Any API testing tool such as Postman can be used to play around

Distributed Object Architecture

- There is no distinction between client and servers
- Objects co-exist across network and interact as if they were on a single machine
- Objects provides services to other objects
- Objects communicate through a middleware
  - e.g. CORBA

Web Services vs Distributed Objects

- Web services are inherently more coarse-grained than objects
- Recall that objects and their fine-grained nature were a drawback in terms of latency

Web API vs Web Service

- Web APIs are protocol-agnostic i.e. they can use any design style

- Web Services are not protocol-agnostic and are mostly restricted to Simple Object Access Protocol (SOAP)

SOAP

- SOAP (Simple Object Access Protocol) is a lightweight protocol for exchange of information between applications

SOAP

- SOAP (Simple Object Access Protocol) is a lightweight protocol for exchange of information between applications

- Doesn't that sound familiar?
  - Much like REST architecture???

SOAP

- SOAP is an object-oriented protocol that allows exchange of information between client and server applications

- This data (objects) are serialized to/from XML

SOAP

- A SOAP message (also called SOAP envelope) has at least one Body element and (optional) exactly one Header

SOAP vs REST

- SOAP operations are not limited to CRUD
- REST operations are limited to CRUD

- SOAP only works with XML as data format
- REST supports XML, JSON, plain text and more

- SOAP can work with any other TCP-based communication protocol than HTTP (e.g. SMTP)
- REST only supports HTTP

WSDL

- The WSDL (Web Service Description Language) is an XML-based grammar for describing the Web Services
  - WSDL defines the methods and the data associated with Web Service.
  - Since WSDL is XML-based, it is both human and machine readable. However WSDL is designed to be used by machines for automated implementation of interface contracts.

WSDL

WSDL Section	Use
Types	Defines types
Messages	Abstract message signatures
Operations	Abstract method definitions
Port Type	Abstract interface based on operations
Binding	Interface and method implementations
Port	Associates binding with a specific address
Service	Collection of ports

Example: WSDL

```
 <message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```

Some Key Questions

- Can you recall the difference between an API and a webhook?
- How is an API different than a Web Service?
- How are RESTful APIs stateless?
- Major differences between SOAP & REST?