

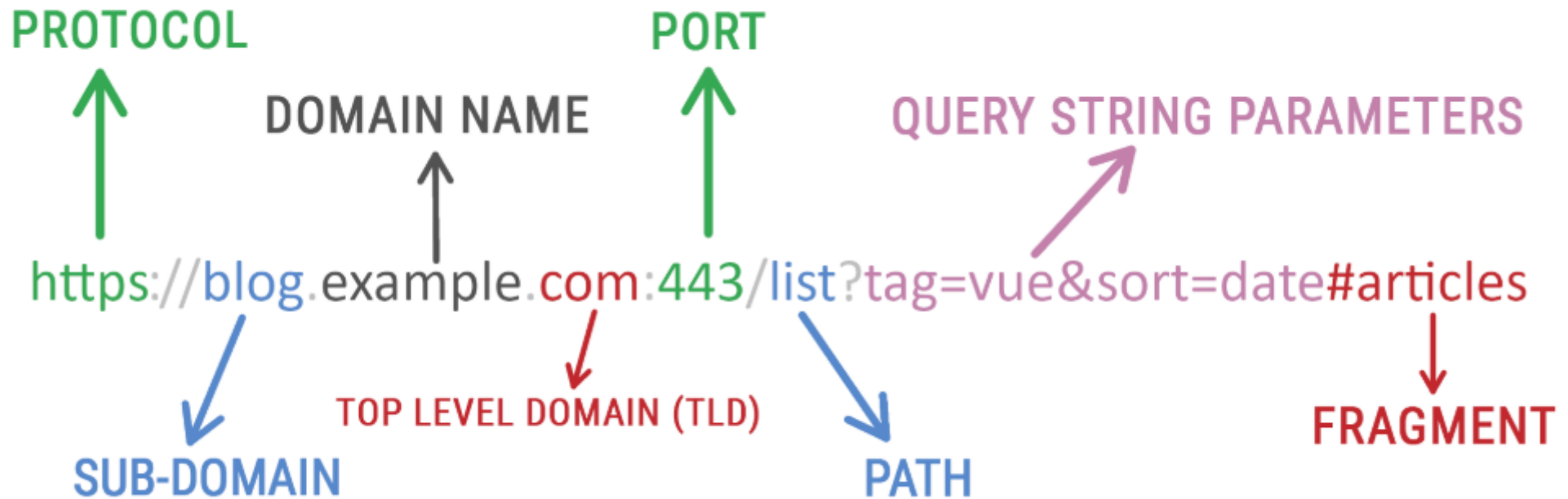
# Module 3 Day 1

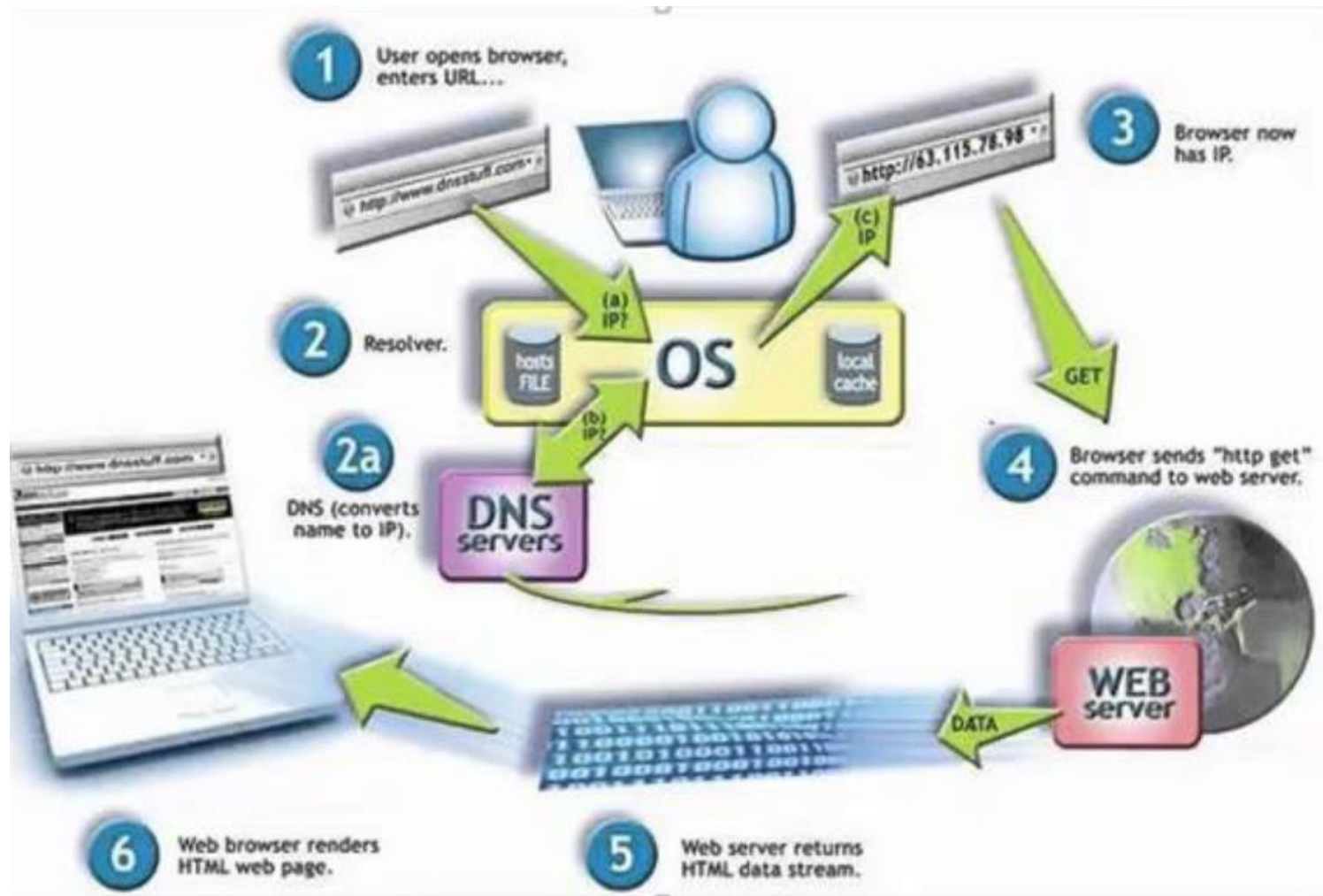
## HTTP and Consuming APIs - Part 1

# The Internet and the World Wide Web

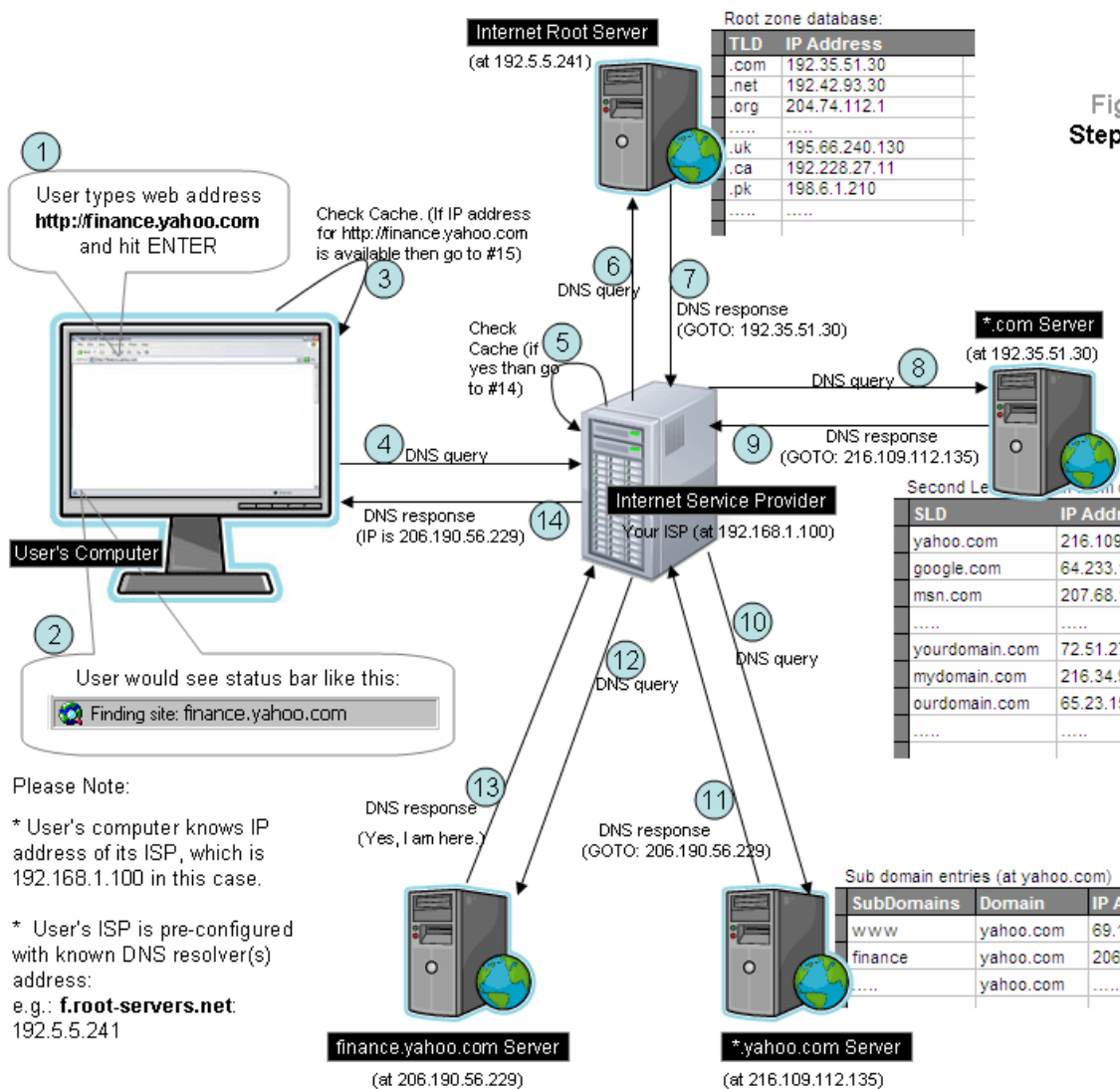
- Are the Internet and WWW the same thing?
- If not, what is the difference?
- What is the protocol that defines the Internet?
- What is the protocol that defines the WWW?
- What other applications / protocols run on the Internet?

# URL / URI





Fetching a  
URL

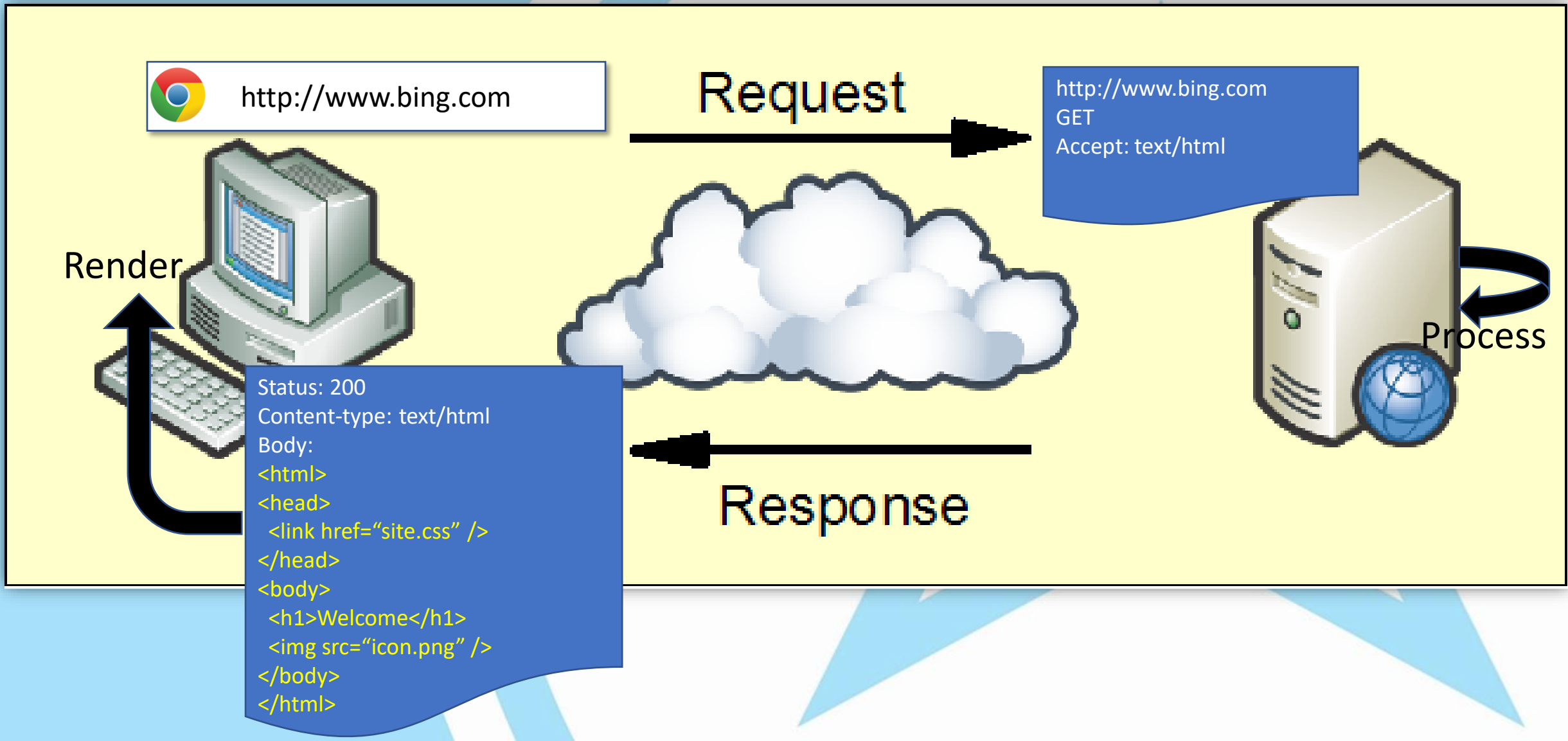


# Multi-level domain names

# HTTP Request-Response

- Request
  - URL – Uniform Resource Locator
  - HTTP Method / Verb (GET, PUT, POST, DELETE)
  - Headers (auth, content-type, cookies)
  - Body (sometimes)
- Response
  - Status (2xx, 3xx, 4xx, 5xx)
  - Headers (content-type, set-cookie)
  - Body (sometimes)

# HTTP Request-Response





# HTTP Request-Response

- Stateless
  - server “remembers” nothing about the client between requests
- Cyclic
  - Response from server contains references, links and redirects
  - Client (e.g., Browser) responsible for making multiple requests to completely fulfill the user’s query and display a complete page
- Browser Developer Tools (F12)
  - Help you see all the requests that are taking place
  - Help understand performance issues



Demo



# HTTP Headers

Header	What it does	Example
Accept	Request: tells server what type of content is acceptable	Accept: text/plain,application/json
Authorization	Request: Specifies the type of authorization to be sent	Authorization: Bearer
Content-type	Response: Tells what type of content has been sent	Content-type: application/json
Content-length	The length of the body of the request	Content-length: 255

# HTTP Methods (Verbs)

Method	Definition
GET	Read and return the specified resource. Requests using GET should only retrieve data, never change server data.
POST	Create a new resource, often causing a change in state on the server. Not idempotent (duplicate calls might cause two new resources).
PUT	Update the specified resource with the request payload. Idempotent (duplicate calls may be made without side effects).
DELETE	Remove the specified resource from the server.

# REST API

- Representational State Transfer (REST) is a software architectural style that defines a set of constraints to be used for creating Web services.  
[https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)
- A RESTful API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data.  
<https://searchmicroservices.techtarget.com/definition/RESTful-API>
- Generally speaking, when people use the term REST API, they are referring to an API that is accessed via the HTTP protocol at a predefined set of URLs (uniform resource locators) representing the various **resources** with which interactions can occur. <https://www.twilio.com/docs/glossary/what-is-a-rest-api>

# Tools

- JSON Viewer (Chrome extension)
- Browser Developer Tools (F12)
- Postman
- <https://api.exchangerate-api.com/v4/latest/USD>
- <https://api.openweathermap.org/data/2.5/weather?zip=44101&units=imperial&appid=b49555ced86dc4c82f40c75e15906a2e>



Demo

# Consuming an API in C#

- RestSharp is the 3<sup>rd</sup>-party package we use
- Manage NuGet Packages

```
string API_URL = "https://api.exchangerate-api.com/v4";  
1 reference  
public Exchange GetExchange(string currency)  
{  
    // /latest/USD  
    RestClient client = new RestClient(API_URL);  
    RestRequest request = new RestRequest($"latest/{currency}", DataFormat.Json);  
    //IRestResponse response = client.Get(request);    // This returns json as a string only  
    IRestResponse<Exchange> exchangeResponse = client.Get<Exchange>(request);    // This de-serializes json into Exchange  
    return exchangeResponse.Data;  
}
```

Demo

# Consuming an API in C#

- Json-server is a local "fake" api server
- We must prepare and start it
  - Npm install
  - Npm run serve
- Data shows up on <http://localhost:3000>
- Our lecture code shows hotels and reviews



Let's  
Code



# Lists of Public API's

- <https://imago.dev/10-free-public-apis/>
- <https://shkspr.mobi/blog/2016/05/easy-apis-without-authentication/>
- <https://apilist.fun/>
- <https://github.com/public-apis/public-apis>
- <https://public-apis.xyz/>
- <https://rapidapi.com/> (this one's a little commercial)