Microsoft Learn
Student Ambassadors

Introduction to ASP .NET Core Web API

Sandu Cristian, Beta Student Ambassador

Content

- What is an API?
- API Architecture Type
- REST
- Hypertext Transfer Protocol
- Status Codes

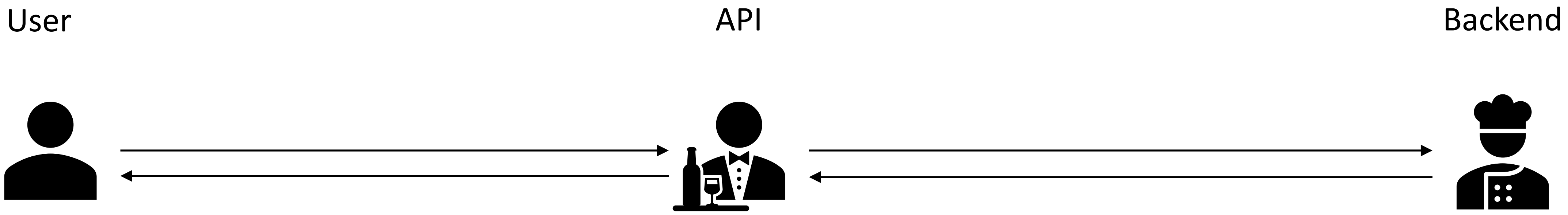
What is an API?

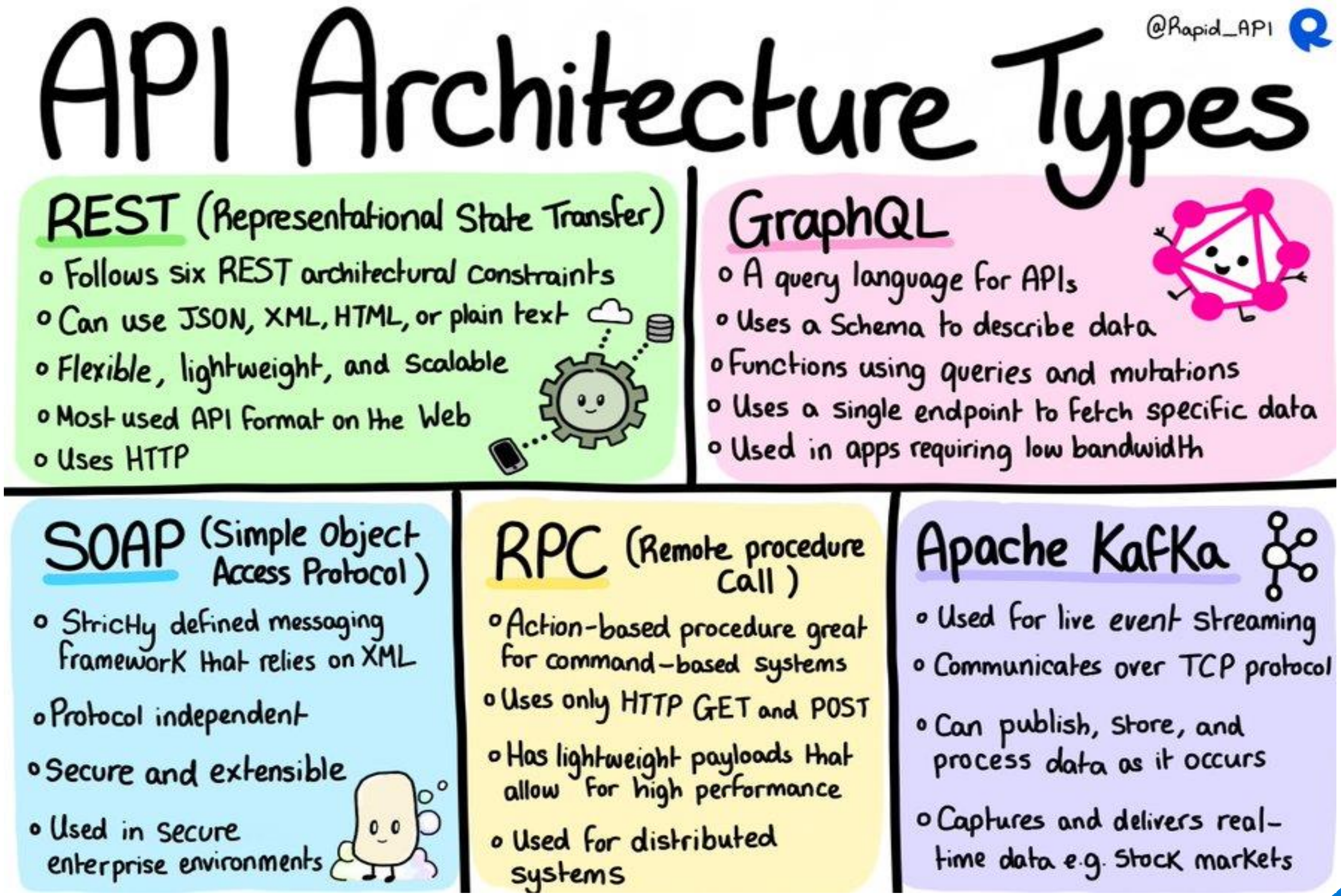
API stands for **A**pplication **P**rogramming **I**nterface.

A Web API is an application programming interface for the Web.

A Browser API can extend the functionality of a web browser.

A Server API can extend the functionality of a web server.





API Architecture Types

@Rapid_API

REST (Representational State Transfer)

- Follows six REST architectural constraints
- Can use JSON, XML, HTML, or plain text
- Flexible, lightweight, and Scalable
- Most used API Format on the Web
- Uses HTTP

GraphQL

- A query language for APIs
- Uses a Schema to describe data
- Functions using queries and mutations
- Uses a single endpoint to fetch specific data
- Used in apps requiring low bandwidth

SOAP (Simple Object Access Protocol)

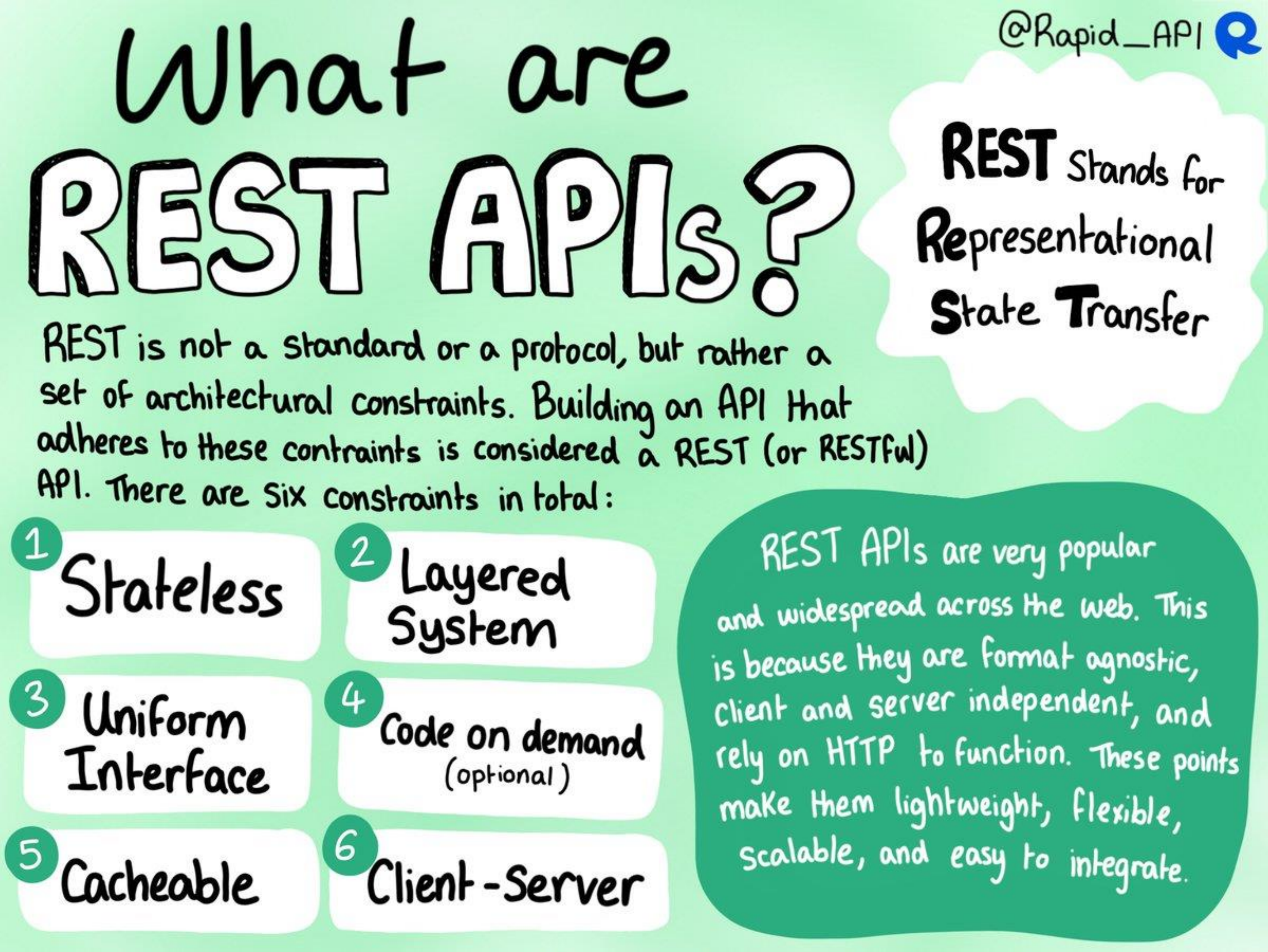
- Strictly defined messaging framework that relies on XML
- Protocol independent
- Secure and extensible
- Used in secure enterprise environments

RPC (Remote procedure Call)

- Action-based procedure great for command-based systems
- Uses only HTTP GET and POST
- Has lightweight payloads that allow for high performance
- Used for distributed systems

Apache KafKa

- Used for live event streaming
- Communicates over TCP protocol
- Can publish, store, and process data as it occurs
- Captures and delivers real-time data e.g. Stock markets



@Rapid_API

What are REST APIs?

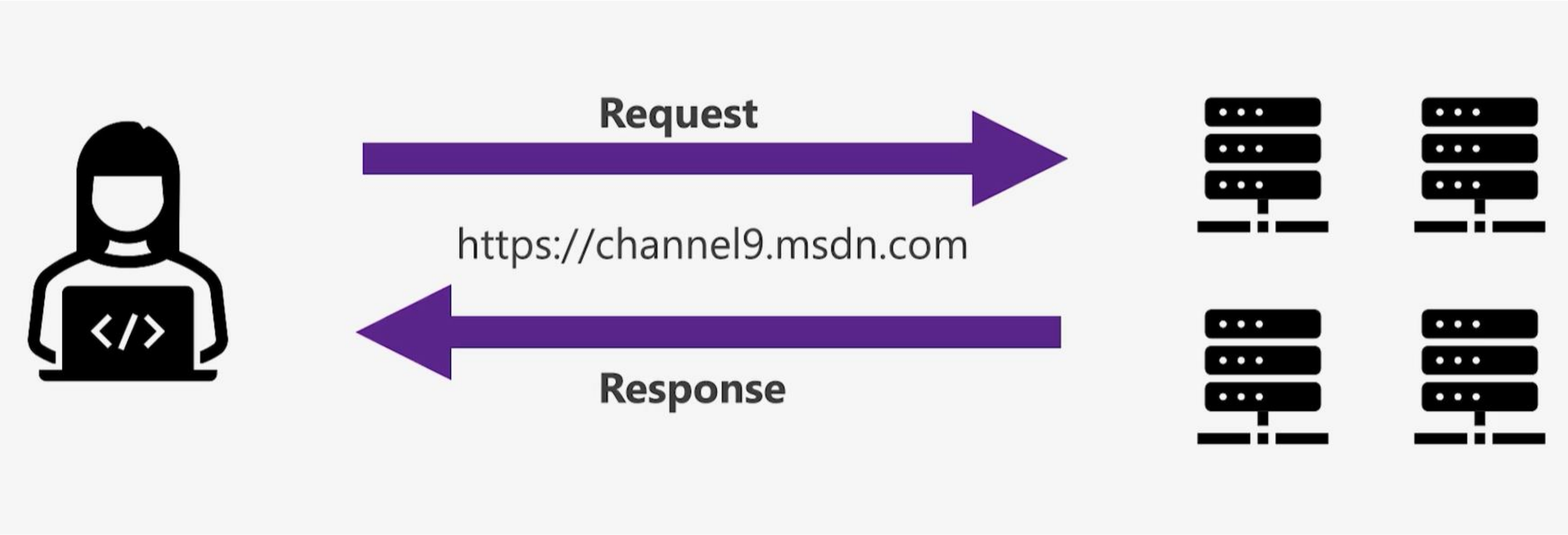
REST Stands for **Re**presentational **S**tate **T**ransfer

REST is not a standard or a protocol, but rather a set of architectural constraints. Building an API that adheres to these contraints is considered a REST (or RESTful) API. There are six constraints in total:

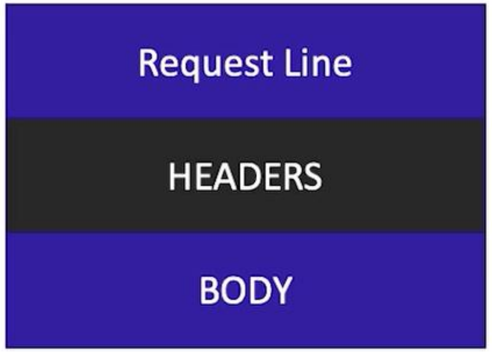
1. Stateless
2. Layered System
3. Uniform Interface
4. Code on demand (optional)
5. Cacheable
6. Client-Server

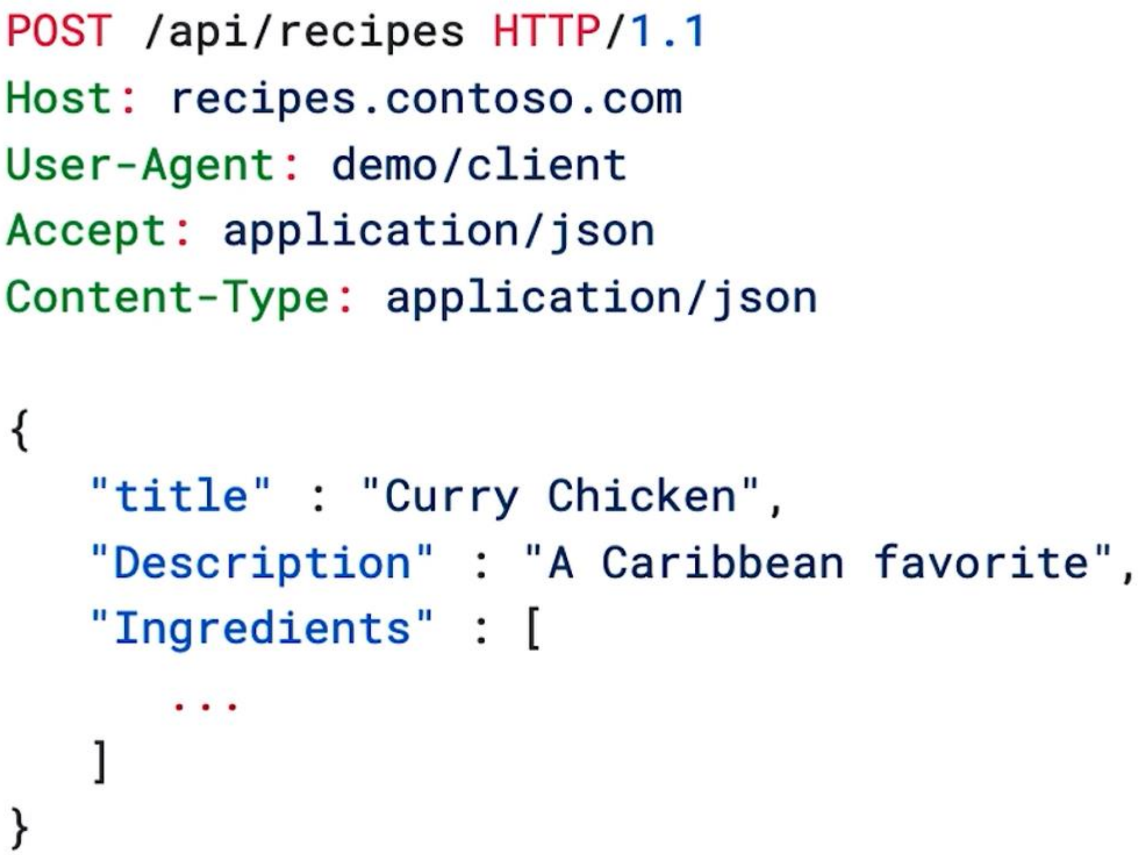
REST APIs are very popular and widespread across the web. This is because they are format agnostic, client and server independent, and rely on HTTP to function. These points make them lightweight, flexible, scalable, and easy to integrate.

Hypertext Transfer Protocol



HTTP Request

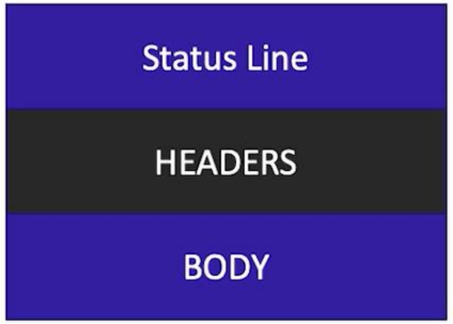




```
POST /api/recipes HTTP/1.1
Host: recipes.contoso.com
User-Agent: demo/client
Accept: application/json
Content-Type: application/json

{
   "title" : "Curry Chicken",
   "Description" : "A Caribbean favorite",
   "Ingredients" : [
      ...
   ]
}
```

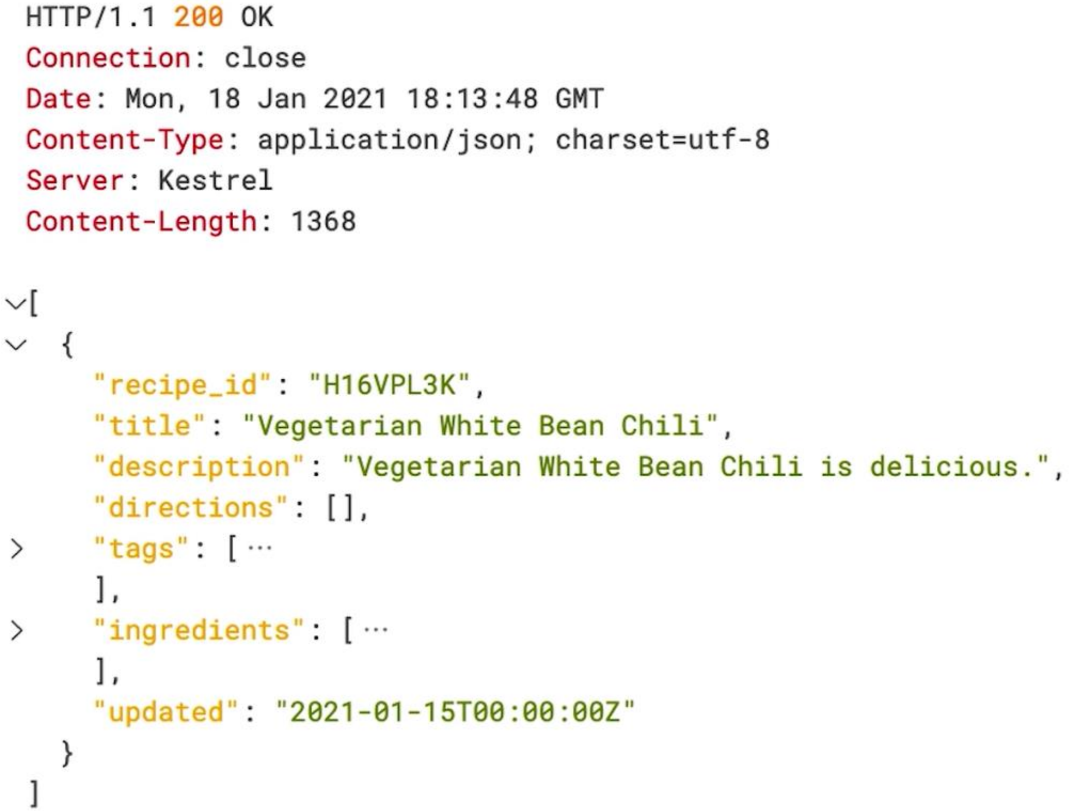
HTTP Response



Status Line

HEADERS

BODY



```
HTTP/1.1 200 OK
Connection: close
Date: Mon, 18 Jan 2021 18:13:48 GMT
Content-Type: application/json; charset=utf-8
Server: Kestrel
Content-Length: 1368

[
  {
    "recipe_id": "H16VPL3K",
    "title": "Vegetarian White Bean Chili",
    "description": "Vegetarian White Bean Chili is delicious.",
    "directions": [],
    "tags": [ ⋯
    ],
    "ingredients": [ ⋯
    ],
    "updated": "2021-01-15T00:00:00Z"
  }
]
```

Status Codes

1xx: Informational

2xx: Success

3xx: Redirection

4xx: Client Errors

5xx: Server Errors

- 200 OK: The request was successful, and the response contains the requested data.
- 201 Created: The request was successful, and a new resource has been created.
- 204 No Content: The request was successful, but there is no data to return.
- 400 Bad Request: The request was malformed or invalid.
- 401 Unauthorized: The request requires authentication, but the user has not provided valid credentials.
- 403 Forbidden: The user does not have permission to access the requested resource.
- 404 Not Found: The requested resource could not be found.
- 500 Internal Server Error: The server encountered an error while processing the request.

Homework 1

https://learn.microsoft.com/en-us/shows/beginners-series-to-web-apis/