UNDERSTANDING REST API

WEB DEVELOPMENT I

Contents

What is an API?

What Is API?

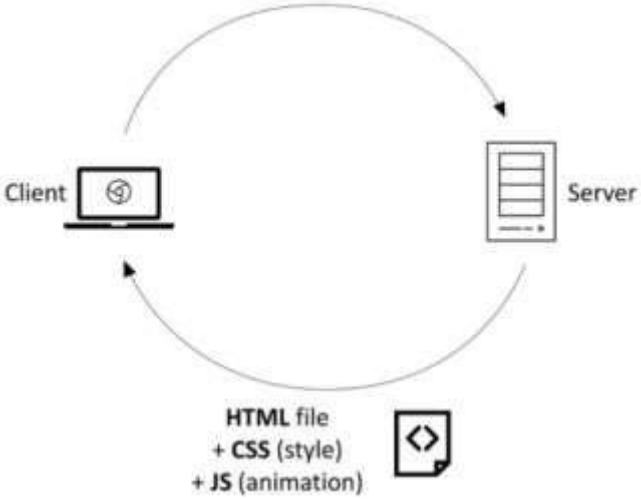
API Stands for Application Programming Interface

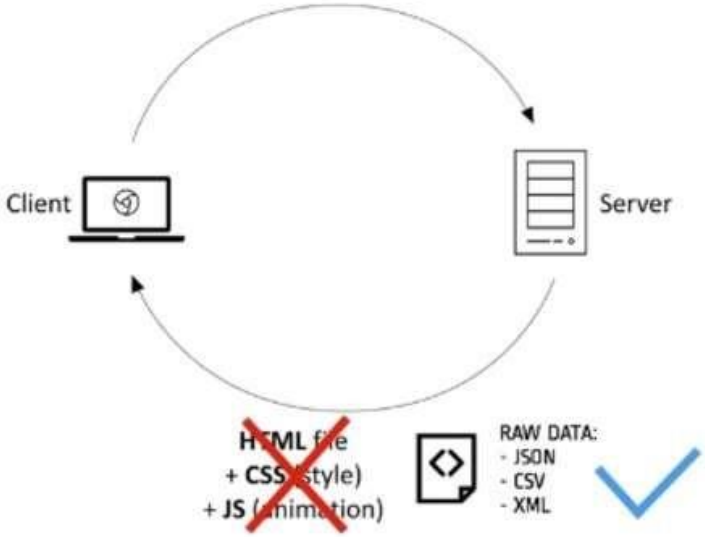
API is a set of protocols which acts as a medium of communication between programs. In other words, it is a way two programs talk to each other.

APIs

- API stands for Application Programming Interface.
- API's basically allow your product or service to talk to another product or service.
- Software-to-software interaction, not user interaction.
- They are used to give people access to your data/resources from outside the firewall.
- This means opening up your product's data and functionality to other developers both internally and externally.

Website vs. API






```
{"richestPeople":[
  {"firstName":"Bill", "lastName":"Gates", "value":"81"},
  {"firstName":"Carlos", "lastName":"Slim", "value":"79"},
  {"firstName":"Warren", "lastName":"Buffet", "value":"72"}
]}
```

Classification of APIs

Web Service API
- SOAP
- XML-RPC and JSON-RPC
- REST

WebSocket APIs

Library-based APIs
- JavaScript
- TWAIN

Class-based APIs (Object Orientation)
- Java API
- Android API

OS Functions and Routines
- Access to file system
- Access to user interface

Object Remoting APIs
- CORBA
- .NET Remoting

Hardware APIs
- Video acceleration
- Hard disk drives
- PCI buses

What is REST API?

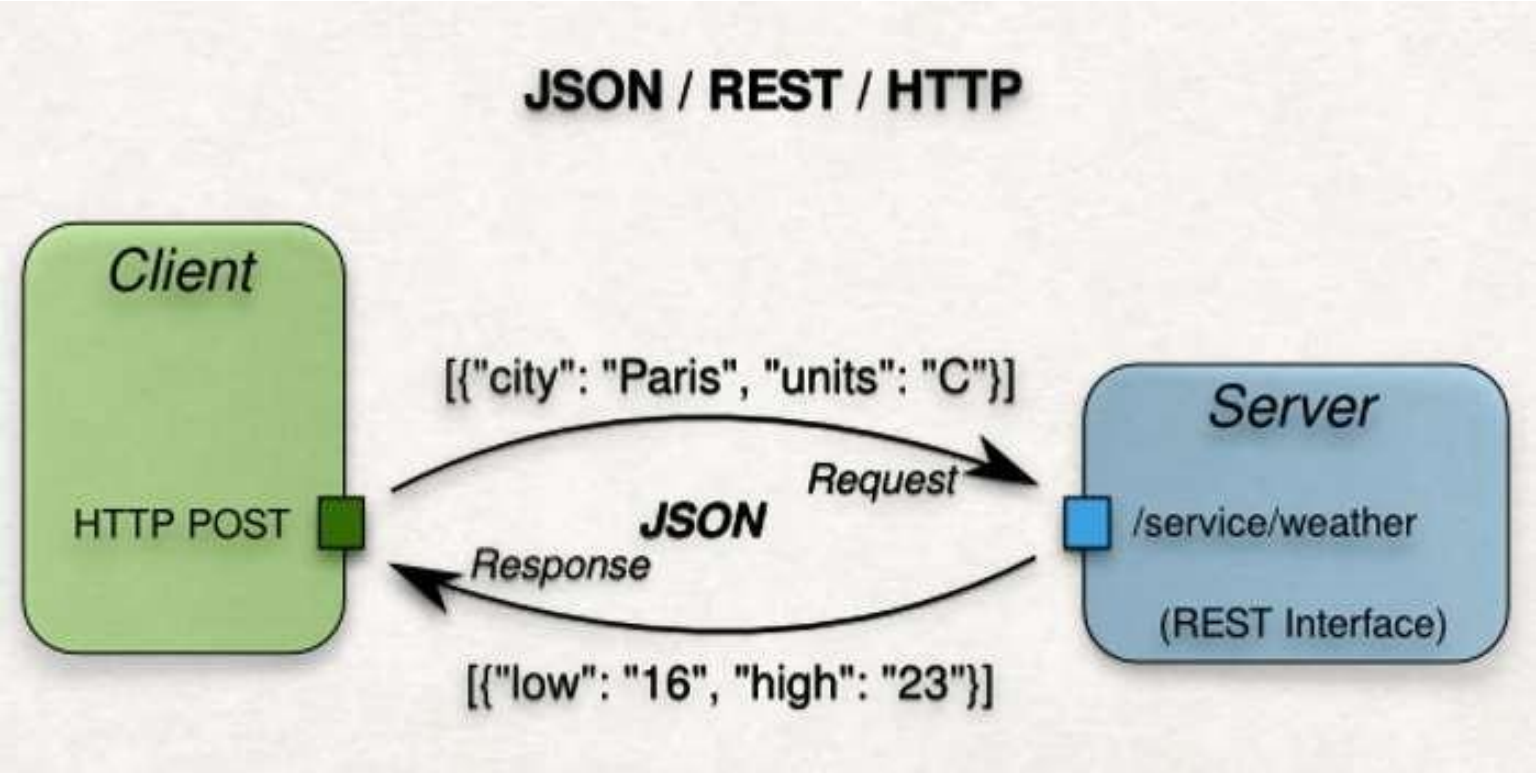
- REST stands for Representational State Transfer.
- It is a lighter weight alternative to SOAP and WSDL XML-based API protocols.

What model does REST use?

- REST uses a client-server model, where the server is an HTTP server and the client sends HTTP verbs (GET, POST, PUT, DELETE), along with a URL and variable parameters that are URL-encoded.

- The URL describes the object to act upon and the server replies with a result code and valid JavaScript Object Notation (JSON).

Client/Server through API



REST HTTP Methods

Method	Description
GET	Request to read a webpage
HEAD	Request to read a webpage's header
PUT	Request to store a webpage
POST	Append to a names resource
DELETE	Remove the webpage
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options

HTTP Codes

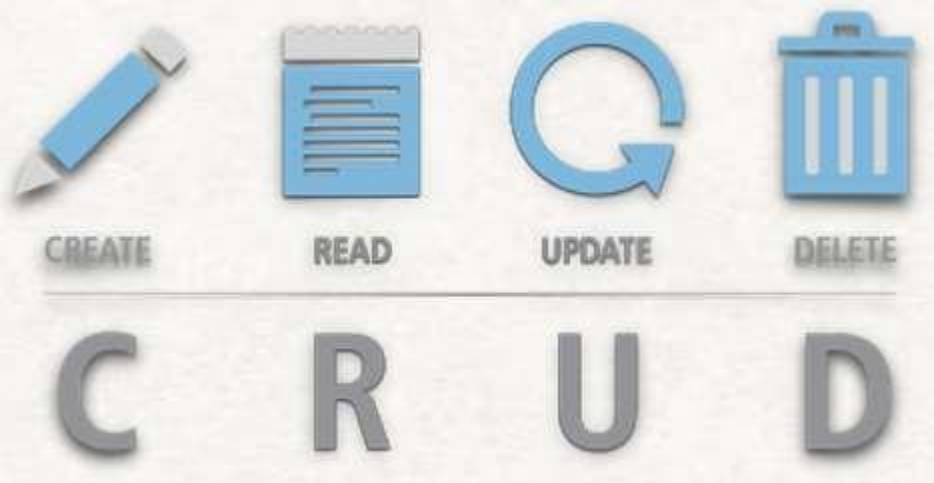
- 200 - “OK”.
- 201 - “Created” (Used with POST).
- 400 - “Bad Request” (Perhaps missing required parameters).
- 401 - “Unauthorized” (Missing authentication parameters).
- 403 - “Forbidden” (You were authenticated but lacking required privileges).
- 404 - “Not Found”.

Advantages of REST

- Separation between the client and the server.
- Visibility, reliability and scalability.
- The REST API is always independent of the type of platform or languages.
- Lighter weight alternative to SOAP and WSDL XML-based API protocols.

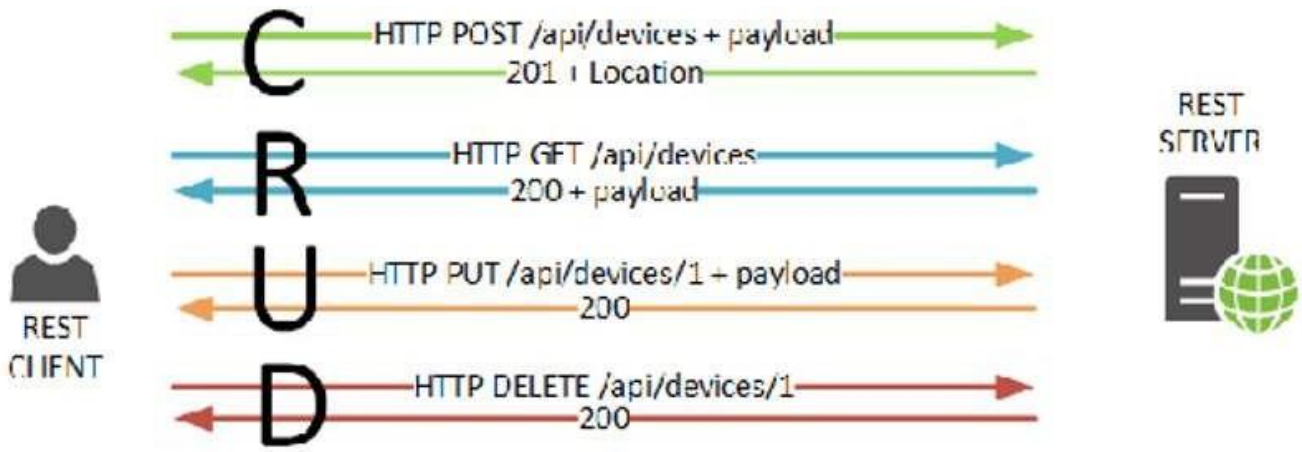
What is CRUD

The CRUD acronym is often used to describe database operations. CRUD stands for CREATE, READ, UPDATE, and DELETE.



CRUD Operations

- POST: A client wants to insert or create an object.
- GET: A client wants to read an object.
- PUT: A client wants to update an object.
- DELETE: A client wants to delete an object.



DB connection API example

CREATE A TABLE “users” under “api” database with the following fields

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking

Table structure | Relation view

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
☐	1	id	int(11)			No	*None*		AUTO_INCREMENT	Change	Drop	More
☐	2	name	text	utf8mb4_general_ci		No	*None*			Change	Drop	More
☐	3	age	int(11)			No	*None*			Change	Drop	More
☐	4	email	text	utf8mb4_general_ci		No	*None*			Change	Drop	More

←T→			id	name	age	email
☐ Edit	Copy	Delete	1	User1	30	user1@gmail.com
☐ Edit	Copy	Delete	2	user2	40	user2@gmail.com

Check all *With selected:* Edit Copy Delete Export

apitest.php file

apitest.php

```php
<?php
$con=mysqli_connect("localhost","elymambou","elymambou","api");
$response=array();
if($con){
    $sql="select * from users";
    $result=mysqli_query($con, $sql);
    if($result){
        $x=0;
        while($row=mysqli_fetch_assoc($result))
        {
            $response[$x]['id']=$row['id'];
            $response[$x]['name']=$row['name'];
            $response[$x]['age']=$row['age'];
            $response[$x]['email']=$row['email'];
            $x++;
        }
    echo json_encode($response, JSON_PRETTY_PRINT);
    }
else
{
    echo "Database connection failed";
}
}
?>
```

GET cmd from Postman or browser

http://localhost/authentication/view.php

Save

GET http://localhost/api/apitest.php Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies (1) Headers (7) Test Results Status: 200 OK Time: 25 ms Size: 482 B Save Response

Pretty Raw Preview Visualize JSON


```
[
    {
        "id": "1",
        "name": "User1",
        "age": "30",
        "email": "user1@gmail.com"
    },
    {
        "id": "2",
        "name": "user2",
        "age": "40",
        "email": "user2@gmail.com"
    }
]
```


Using PostMan as the client to view users

CRUD Application example

REST API objects and their operations

- Records
  - Read (all records)
  - Read one
  - Create
  - Update
  - Delete
  - Search

- https://codewithbish.com/crud-application-in-php-using-mysql-for-beginners/
  - CRUD app using templating and MySQL