**API**

**(Application Programing Interface)**

API stands for Application Programming Interface. In the context of APIs, the word Application refers to any software with a distinct function. Interface can be thought of as a contract of service between two applications. This contract defines how the two communicate with each other using requests and responses.

It act as a bridge between different software and devices.

API architecture is usually explained in terms of client and server. The application sending the request is called the client, and the application sending the response is called the server.There are four different ways that APIs can work depending on when and why they were created.

**SOAP APIs**

These APIs use Simple Object Access Protocol. Client and server exchange messages using XML. This is a less flexible API that was more popular in the past.

**RPC APIs**

These APIs are called Remote Procedure Calls. The client completes a function (or procedure) on the server, and the server sends the output back to the client.

**Websocket APIs**

[Websocket API](https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-websocket-api-overview?pg=wianapi&cta=websocketapi) is another modern web API development that uses JSON objects to pass data. A WebSocket API supports two-way communication between client apps and the server. The server can send callback messages to connected clients, making it more efficient than REST API.

**REST APIs**

These are the most popular and flexible APIs found on the web today. The client sends requests to the server as data. The server uses this client input to start internal functions and returns output data back to the client.

**What are REST APIs?**

REST stands for Representational State Transfer. REST defines a set of functions like GET, PUT, DELETE, etc. that clients can use to access server data. Clients and servers exchange data using HTTP.

**What are the benefits of REST APIs?**

1. Integration

2. Innovation

3. Expansion

4. Ease of maintenance

**What are the different types of APIs?**

APIs are classified both according to their architecture and scope of use. We have already explored the main types of API architectures so let’s take a look at the scope of use.

**Private APIs**

These are internal to an enterprise and only used for connecting systems and data within the business.

**Public APIs**

These are open to the public and may be used by anyone. There may or not be some authorization and cost associated with these types of APIs.

**Partner APIs**

These are only accessible by authorized external developers to aid business-to-business partnerships.

**Composite APIs**

These combine two or more different APIs to address complex system requirements or behaviors.

You would have to use HTTP method like **GET,POST,PUT,DELETE**,etc to make use of the endpoint

|  |  |  |
| --- | --- | --- |
| No. | SOAP | REST |
| 1) | SOAP is a protocol. | REST is an architectural style. |
| 2) | SOAP stands for Simple Object Access Protocol. | REST stands for Representational State Transfer. |
| 3) | SOAP can't use REST because it is a protocol. | REST can use SOAP web services because it is a concept and can use any protocol like HTTP, SOAP. |
| 4) | SOAP uses services interfaces to expose the business logic. | REST uses URI to expose business logic. |
| 5) | JAX-WS is the java API for SOAP web services. | JAX-RS is the java API for RESTful web services. |
| 6) | SOAP defines standards to be strictly followed. | REST does not define too much standards like SOAP. |
| 7) | SOAP requires more bandwidth and resource than REST. | REST requires less bandwidth and resource than SOAP. |
| 8) | SOAP defines its own security. | RESTful web services inherits security measures from the underlying transport. |
| 9) | SOAP permits XML data format only. | REST permits different data format such as Plain text, HTML, XML, JSON etc. |
| 10) | SOAP is less preferred than REST. | REST more preferred than SOAP. |

**5 Essential HTTP Methods in RESTful API Development**

**1.GET**

Use GET requests **to retrieve resource representation/information only** – and not modify it in any way. As GET requests do not change the resource’s state, these are said to be **safe methods**.

**2. POST**

The POST method sends data to create a ‘new record ‘on the server.

**3. PUT**

The PUT method sends data to update an ‘existing record ‘on the server.

**4. PATCH**

Like the PUT method, PATCH is also used to send data to update an ‘existing record’ on the server. But the important difference between PUT and PATCH is that PATCH only applies partial modifications to the record instead of replacing the whole record.

**5. DELETE**

The DELETE method is used to delete record(s) from the server.

**Difference between POST, PUT and PATCH**

**PUT:**

Put request is used for both creating and updating a new object in the database. If the resource already exists, then Put will update the resource. If not, it will create one.

**POST:**

Post request is used for creating a new resources. It allows clients to create resources without knowing the URI of the new resources.

**PATCH:**

Patch is used to apply the partial modification to a resources.

**What do you mean by RESTful web services?**

REST API is also known as RESTful web services that follow the REST architecture.

**What are cache-control headers?**

Cache-control headers are used to control catching and to attain caching ability. The most commonly used cache-control headers are public, private, and No-Store.

**What are the features of RESTful web services?**

REStful web services have the following unique features:

Client-server decoupling

Communication support

Lightweight

Uniform interface

Stateless

Layered system

Cacheable

Code on demand

**Explain ‘Addressing’ in RESTful web services.**

The process of locating various types of resources with the help of a URL on the REST server is known as ‘addressing’ in RESTful web services. Usually, single or multiple resources are addressed by resources.

**How can RESTful web services be tested?**

The RESTful web services can be tested with the help of tools such as Swagger and Postman, which enable users to inspect query parameters, response headers, and headers, documentation of the endpoints, and conversion of endpoints to XML and JSON.

**What are payloads in RESTful web services?**

Payloads refers to the data in the body of the http request and/or response message in GET,PUT or POST request.

Payloads are the request data passed through the POST or GET method and found in the message’s body of an HTTP request in RESTful web services.

**What is the maximum payload size that can be sent in POST methods?**

Theoretically, there is no such maximum limit for payload size that can be sent in POST methods. However, payloads with larger sizes can consume larger bandwidth. Thus the server could take more time to proceed with the request.

**Which protocol does REST APIs use?**

Protocols are used to communicate with clients where REST APIs use HTTP protocol for it.

**In REST APIs, which markup languages are used to represent the resources?**

The resources in REST APIs are represented with the help of XML (extensible markup language) and JSON (JavaScript Object Notation).

**Note**:To be idempotent, only the state of the server is considered. The response returned by each request may differ: for example, the first call of a DELETE will likely return a 200 , while successive ones will likely return a 404

**Differentiate POST and PUT methods.**

**POST Method**

POST can create a resource on the server.

POST is not idempotent.

POST responses are cacheable.

**PUT Method**

PUT is used to replace a resource at a specific URI with another resource.

PUT is idempotent that it will only result in one resource even after calling it multiple times.

PUT responses are not.

**What is CRUD?**

CRUD stands for “Create, Read, Update, and Delete.”

POST--Create

PUT--Update

GET--Read

DELETE --Delete

**The main parts of an HTTP response**

The main parts of the HTTP response are the HTTP version, Status line, HTTP Response Header, and HTTP Response body.

**What are the most common HTTP response status codes you see while working in REST API?**

Some of the most common response status codes are

200 OK,

201 Created,

400 Bad Request,

401 Unauthorized,

403 Forbidden,

404 Not Found,

500 Internal Server Error,

502 Bad Gateway,

503 Service Unavailable, etc.

**What is caching in the REST API?**

REST API stores a copy of a server response in a particular location of computer memory to retrieve the server response fast in the future. This method is temporary and called "catching."

What’s a real-world example of a REST API?

Youtube API,

Facebook API,

Weather Info API and many more

**Disadvantages of RESTful web services?**

RESTful web services are stateless and do not maintain session simulation responsibility as the client side does not provide a particular session id for it.

REST is not able to impose the security restriction inherently. However, it inherits them with the help of implementing protocols. Thus, the integration of SSL/TLS authentication needs to be done very carefully for better security measures of the REST APIs.

**Advantages of REST**

HTTP makes the implementation of REST easy.

REST fits in the existing infrastructure of the web, thus the web application can easily implement the REST. XML and JSON web technologies make REST easy to learn.

The client and server communication is stateless, thus the integration is easy to build and scalable, and manageable with respect to time.

The REST architecture can adapt to a huge variety of cases due to its flexibility.

The lightweight architecture of REST makes it easy to build the applications faster as compared to other types of APIs.

REST can be tested easily in the browser with the help of API testing tools.

**How do you keep REST APIs secure?**

REST APIs can be kept secure with the help of safety measures such as Authentication and authorization, API Server Validation, TSl/SSL Encryption, Rate-limiting for DDoS attacks, and sensitive information such as username, password, or authentication token should not be visible in URIs

**What does the HEAD method in REST APIs do?**

The HEAD method is used to return the HTTP Header in read-only form and not the Body.

**What are HTTP status codes and their meaning?**

Code 200: success.

Code 201:resource has been successfully created.

Code 204: no content in the response body.

Code 404: no method available.

**Why is the proper representation of resources required?**

Proper representations of resources in the proper format allow the client to easily understand the format and determine the identification of resources easily.

Important aspects of RESTful web services implementation.

ResourcesRequest

Headers

Request Body

Response Body

Status codes

**What is the difference between API and REST API?**

An Application Programming Interface entails rules used to define how different devices or applications communicate with each other and connect to each other. A REST API follows the principles of the REST architectural pattern to create web services.

API: General term for software talking to software.

REST API: A type of API that follows strict rules to make communication easier, faster and more predictable.

An API become a REST API when it follows the specific principles and constraints of the REST architectural styles.

REST API:

1. Uses HTTP protocol:

The API must uses http method like:

1. GET: To retrieve data
2. POST: To send data
3. PUT: To update data
4. DELETE: To remove data
5. PATCH: Partial modification of existing data
6. Stateless Communication

Each request must be independent. The Server doesn’t store any information about previous request. All the information needed must be included in each request.

1. Resource-Based Design

REST API’s focus on resources .each resource has a unique URL such as:

Hppts://api.example.com/users to get a list of users.

1. Representation of resources:

Data must be exchanged in standard format like :

JSON (Most common)

XML

Plain text

1. Uniform Interface:

REST API must follow a consistent structure such as:

* Endpoint (URLs) for accessing resources.
* Standard response codes(e.g. 200 for success,404 for not found)

API vs REST API

REST is a type of API. Not all APIs are REST, but all REST services are APIs.

* RESTful API: Ideal for web applications where scalability, statelessness, and simplicity are crucial. Examples include cloud services, mobile services, and public APIs for services like social media sites.
* Other APIs: Useful when there's a need for continuous communication or stateful interactions. For instance, gaming applications, real-time chat applications, or systems with specific protocol requirements.

**What is the full-form of REST API?**

REST API is the acronym used for Representational State Transfer Application Program Interface.

**What is URI?**

**Uniform Resource Identifier** is the full form of URI which is used for identifying each resource of the REST architecture. URI is of the format:

<protocol>://<service-name>/<ResourceType>/<ResourceID>

There are 2 types of URI:

**URN:**Uniform Resource Name identifies the resource by means of a name that is both unique and persistent.

These follow the urn scheme and usually prefixed with urn:. Examples include

urn:isbn:1234567890 is used for identification of book based on the ISBN number in a library application.

URL: Uniform Resource Locator has the information regarding fetching of a resource from its location.

Examples include:

http://abc.com/samplePage.html

URLs start with a protocol (like ftp, http etc) and they have the information of the network hostname (sampleServer.com) and the path to the document(/samplePage.html). It can also have query parameters.

**What is a messaging in the context of REST??**

In Rest, messaging refers to the back and forth communication between the client and API

**List Difference between API and Web Service??**

1. All web services are API but not all API are web services

2. Web services might not contain all the specification and cannot perform all the task that API would perform.

3. A web service always needs a network to operate while API don’t need a network for operation.

4. A Web service uses not only three styles of use: SOAP, REST and XML-RPC for communication whereas API may be exposed to in multiple ways.

**What is a resource??**

In Rest, every accessible piece of content on the server is labeled as resource.

A resource is identified with a uniform resource identifier or URI, Client access resources by including their URI’s in HTTP request