1. **What is API**

API stands for 'application programming interface', but basically describes one way to plug your website into another.  APIs just allow applications to communicate with one another.

1. **What is REST API**

API stands for 'application programming interface', but basically describes one way to plug your website into another.  APIs just allow applications to communicate with one another.

1. **Mention what are the HTTP methods supported by REST?**

HTTP methods supported by REST are:

* **GET:**It requests a resource at the request URL. It should not contain a request body as it will be discarded. Maybe it can be cached locally or on the server.
* **POST:**It submits information to the service for processing; it should typically return the modified or new resource
* **PUT:**At the request URL it update the resource
* **DELETE:**At the request URL it removes the resource
* **OPTIONS:**It indicates which techniques are supported
* **HEAD:**About the request URL it returns meta information
* **Use:** WITH HELP OF THE API THE LIFE EASY FOR A DEVELOPER BECAUSE DEVELOPER DESIGN CODE EASILY .AN API CONTROL THE HARDWARE AND SOFTWARE

SUCH THAT IF A USER WANT TO MAKE THE ANY APP IN THE IOSs OPERATING SYSTEM SO APPLE ‘IOS PROVIDE THE MANY APIs TO HELP FOR MAKING THE APPLICATION

* + **Summary: -**

Like if we talk about the only on API is basically called that application interface whenever A client or user want to interact on the server so it’s a more powerful on interacting server side At the old time whenever a developer designing the any code they always has the many difficulties in the developing in the software and consuming too much time for developing a software

* **Next Topic: -** REST ASSURED

This is the library tools for the rest API automation .it is the java-based lib. Also, to design the web services. This is transfer the message or etc form the server side and validate to http response

* **Client server architecture: --** we simply say about it is path for sending the information

For client if we talk about old era, we usually fetch the inform. By the newspaper like regional or not reg. but we fail to fetch the real inform. But in the age of the com. We talk about client on client server side and server on client server side. Now we say clearly a user can update with real time inform.

* **http protocol b/w. client and server: --**

Mainly it is the mediator between system also medium of the communication it is in the xml and json and response on the base of both formatted, server responses

* **http req**. It is the packet of inform. that helpful for communication something from to one computer to another if we talk about on this all inform. send in the form of binary by the client on the server
  + - **Resource URL: http://restapi.demoqa.com/utilities/weatherfull/**
    - **Parameter: city/city Name=city/<NameFoTheCity***>*

It is in the xml and json and response on the base of both formatted , server responses

http req—it is the packet of inform. that helpful for communication something from to one computer to another if we talk about on this all inform. send in the form of binary by the client on the server

• Resource URL: http://restapi.demoqa.com/utilities/weatherfull/

• Parameter: city/cityName=city/<NameFoTheCity*>*

http://restapi.demoqa.com/utilities/weatherfull/city/hyderabad

* **REQUEST LINE ------**In this we use the token of get and put followed by req. URI and then by the http protocol

Request header have the additional information such that req. method client , client port

* **Request body ------- under filling**
* **HTTP RESPONSE**--- Send information from server to client Response status line is in the three part

**1.** HTTP Protocol Version

**2.** Status Code

**3.** Reason Phrase

* It is clearly visible that the Status Line has; following information**:**

***• HTTP Protocol Version as (HTTP/1.1)***

***• Status Code as 200***

***• Status Message as OK***

* **Response body** --- It is having the information that is required by the client
* **Client -Server architecture style----** If back end means front end is user and back end is the data base.

a server serves the information to many clients and a client can use different technologies

* **STATELESS-**-- In this s server have not the context of the client on the previous request basics
* **UNIFORM INTERFACE-** The uniform interface constraint is fundamental to the design of any *REST* service. The uniform interface simplifies and decouples the architecture, which enables each part to evolve independently. The four guiding principles of this interface are:
* ***Identification of resources***
* ***Manipulation of resources through these representations***
* ***Self-descriptive messages***
* **Layered System*=***A layer should not know about the existence of other layers apart from the layers that it directly interacts with***.***
* **Code on Demand==** It is the optional a developer can extend or change the code in the form of
* --- INFORMATION STORE ON THE SERVER FOR THE CLIENT

Resource----- IT IS CALLED THE COMPLETE URL BECAUSE URL IS AS SHOW THE ID.

* REPERESNTATION==A RESOURCE IS THE ACTUAL DATA THAT IS STORE IN THE FROM OF AN XML AND JSON FORMAT
* REPERSENTATION METADATA=
* applet or script what is the rest architectural elements
* **Validate Response Status using Rest Assured ==**

Functional testing of Web Services involves verifying the Responses returned from various End Points. Some of the important Test Verification are

* Response Status
* Response Header
* Response Body
  + This tutorial will cover the following verification:

1. **Validating Response Status Code**
2. **Validating Error Status Code**
3. **Validating Response Status Line**

In the next tutorial will cover the verification of ***Response Body Content***.

From the previous [***First Rest-Assured Test***](http://toolsqa.com/rest-assured/rest-assured-test/) example, a simple test

which calls the Weather web service using a Get Method looks like this

* + **How to Validate Response Status Line?**

The first line returned in the *Response* from Server is called *Status Line.*Status line is composed of three sub strings

* **Http protocol version**
* **Status Code**
* **Status Code’s string value**

During a success scenario a status line will look something like this ***“HTTP/1.1 200 OK”.***First part is Http protocol **(HTTP/1.1).**Second is Status Code **(200).**Third is the Status message**(OK).**

**REST API Interview Q&A**

**Q-1. Explain REST?**

**Ans.** REST stands for Representational State Transfer. REST is an architectural style of developing web services which take advantage of the ubiquity of HTTP protocol and leverages HTTP method to define actions. It revolves around resource where every component is a resource which can be accessed by a common interface using HTTP standard methods.

In REST architecture, a REST Server provides access to resources and REST client accesses and presents those resources. Here each resource is identified by URIs or global IDs. REST uses different ways to represent a resource like text, JSON, and XML.XML and JSON are the most popular representations of resources these days.

**Q-2. Explain The RESTFul Web Service?**

**Ans.** Mostly, there are two kinds of Web Services which are quite popular.

**1.** SOAP (Simple Object Access Protocol) which is an XML-based way to expose web services.

**2.** Web services developed using REST style are known as RESTful web services. These web services use HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines a URI, Uniform Resource Identifier a service, provides resource representation such as JSON and set of HTTP Methods.

**Q-3. Explain What Is A “Resource” In REST?**

**Ans.** REST architecture treats every content as a resource. These resources can be either text files, HTML pages, images, videos or dynamic business data.

REST Server provides access to resources and REST client accesses and modifies these resources. Here each resource is identified by URIs/ global IDs.

**Q-4. Which Protocol Is Used By RESTful Web Services?**

**Ans.** RESTful web services make use of HTTP protocol as a medium of communication between client and server.

**Q-5. What Is Messaging In RESTful Web Services**?

**Ans.** RESTful web services make use of HTTP protocol as a medium of communication between client and server. The client sends a message in the form of an HTTP Request.

In response, the server transmits the HTTP Response. This technique is called Messaging. These messages contain message data and metadata, i.e., information about itself.

**Q-6. State the Core Components Of An HTTP Request?**

**Ans.** Each HTTP request includes five key elements.

**1.** The Verb which indicates HTTP methods such as GET, PUT, POST, DELETE.  
**2.** URI stands for Uniform Resource Identifier (URI). It is the identifier for the resource on the server.  
**3.** HTTP Version which indicates HTTP version, for example-HTTP v1.1.  
**4.** Request Header carries metadata (as key-value pairs) for the HTTP Request message. Metadata could be a client (or browser) type, the format that the client supports, message body format, and cache settings.  
**5.** Request Body indicates the message content or resource representation.

**Q-7. List The Main Differences Between SOAP And REST?**

**Ans.**

|  |  |
| --- | --- |
| **SOAP** | **REST** |
| **1.** SOAP is a protocol through which two computer communicates by sharing the XML document. | **1.** Rest is a service architecture and design for network-based software architecture. |
| **2.** SOAP supports the only XML format. | **2.** It supports many different data formats. |
| **3.** SOAP does not support caching. | **3.** It supports caching. |
| **4.** SOAP is like a custom desktop application, closely connected to the server. | **4.** A REST client is just like a browser and uses standard methods. An application has to fit inside it. |
| **5.** SOAP is slower than the REST. | **5.** It is faster than SOAP. |

**Q 9: Explain the Factors That Help To Decide About The Style Of Web Service To Use? SOAP Or REST?**

**Ans.** In general, using REST-based web service is preferred due to its simplicity, performance, scalability, and support for multiple data formats.

However, SOAP is favorable to use where service requires an advanced level of security and transactional reliability.

But you can read the following facts before opting for any of the styles.

**1.** Does the service expose data or business logic? To expose data REST will be a better choice and SOAP for logic.  
**2.** If the consumer or the service providers require a formal contract, then SOAP can provide such a contract via WSDL.  
**3.** Need to support multiple data formats. REST supports this.

**4.** Support for AJAX calls. REST can use the XMLHttpRequest.  
**5.** Synchronous and asynchronous calls – SOAP enables both synchronous/asynchronous operations whereas REST has built-in support for synchronous.  
**6.** Stateless or Stateful calls -REST is suited for stateless operations.

Here are some of the advanced-level facts that you can consider as well.

**1.** Security requirement – SOAP provides a high level of security.  
**2.** Transaction support – SOAP has good support for transaction management.  
**3.** Limited bandwidth – SOAP has a lot of overhead when sending/receiving packets since it’s XML based, requires a SOAP header. However, the REST requires less bandwidth to send requests to the server. Its messages are mostly built using JSON.  
**4.** Ease of use – It is easy to implement, test, and maintain REST-based application.

**Q-20. Can You Tell Us Which Java API Helps In Developing A RESTFul Web Service?**

**Ans.** There are many frameworks and libraries available that a developer can use to create RESTful web services in Java. For example, the JAX-RS library is a standard way to develop a REST web service.

Also, Jersey is another most popular implementations of JAX-RS which offers more than what the specs recommend. There are others like RESTEasy, RESTlet, and Apache CFX.

If you like Scala, then you should be using the Play framework to develop RESTful web services.

* + **Validate Response Header using Rest Assured**

Headers are used to send across extra information by the server. This extra information is also referred to as Meta data of the Response

* + **How to read different Header Types from HTTP Response*?***

Let’s just see how to read a Header using Rest-Assured. To do that lets do a simple exercise in which the test would record the following Header Types from the Response:

* *Content-Type*
* *Server*
* *Content-Encoding*

The Response interface provides direct methods to access individual header or all the Headers. Simply do a Response followed by a dot ***(Response.head),***all the available methods to get headers will be displayed. Below image shows the available methods

* + **How to Print all the Headers from HTTP Response ?**

Response interface provides two methods

***headers()*** : returns ***Headers***

***getHeaders()*** : returns ***Headers***