API Testing interview questions:

What is the difference between Rest architecture and Soap protocol

|  |  |
| --- | --- |
| Soap uses XML | Rest uses xml, json |
| Little slow than Rest | Rest is faster than soap |
| Caching not support in Soap | Caching is supported |
| Soap we can send through SoapUI | We can send through Postman, Swagger |

What are URI, URL?

URI= (Endpoint) Uniform Resource Identifier (invisible for the users means there is no UI for URI)

URL= Uniform Resource Locator (visible to all the users). A URL aims to find the location or address of a resource on the web.

Example-1 of Endpoint = <https://reqres.in/api/users?page=2>

There are **three** parts to this endpoint:

1)<https://reqres.in> = **Base URL**

2)/API/users= **Path parameter/-- Folders**

3) ? page =2 = **Query parameter/ -- Filtering**

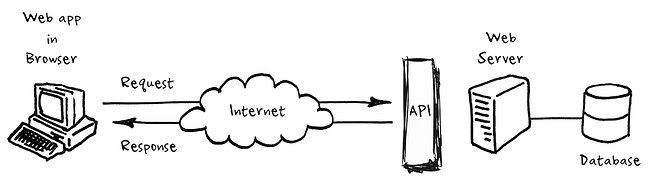
Example-2 of Endpoint =

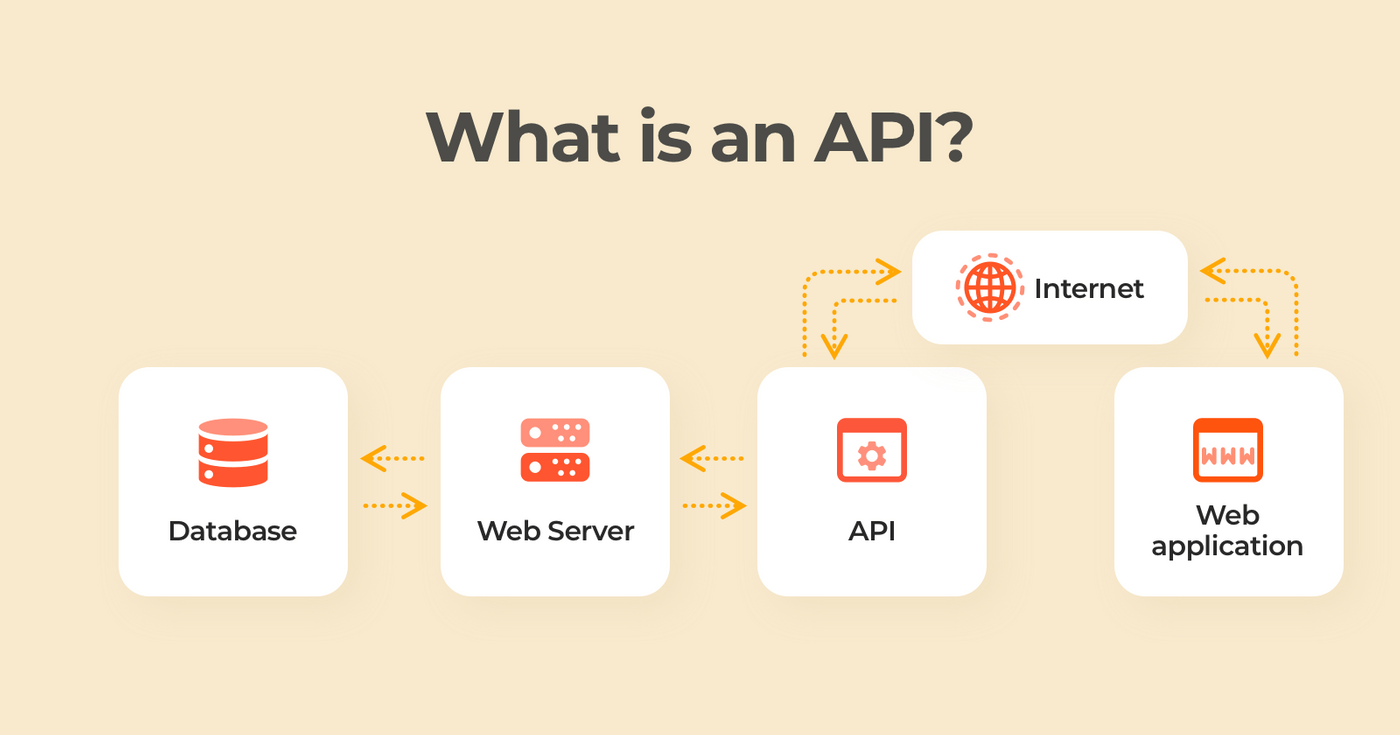
https://www.amazon.com – Base URL/

/gp/browse.html – Path Parameter

?node=10208590011&ref\_=nav\_em\_ods\_smp\_apps\_games\_0\_2\_6\_6 – Query Parameter

**What is an API? (Application Programming Interface)**





**How API works?**

mklAPI means application programming interface. API works as a middleman between 2 servers. API communicate between **2 server**(EX- T-mobile and AT&T),**2 machines(**Router and TV**)** 2 machines can be connected without internet.

What is Webservice Testing?

Here the link starts with “https” which means works with the web only.

we send **Json request** to the **server** and **getting a response**. So along with the request, we pass **headers, parameters, and cookies**.

What are the types of API

There are a few types of API

1. Public API
2. Partner API
3. Internal API

What is API Testing?

Testing the core business logic is API testing. We send a request from a client like Postman and we receive a response from the server.

**API integration**

API integrations are software components that automatically update data between clients and servers. Some examples of API integrations are when automatic data sync to the cloud from your phone image gallery, or the time and date automatically sync on your laptop when you travel to another time zone. Enterprises can also use them to efficiently automate many system functions.

Endpoint VS API

It’s important to note that endpoints and APIs are not the same things. Rather, an endpoint is a component of an API. An API is a set of rules that allow two applications to share resources. Endpoints are the locations of the resources, and the API uses endpoint URLs to retrieve the requested resources.

**What is Header?**

Header is an extra source of information for each API call/API request/http request/HTTP method. When developers create an API, he adds some particular key and value. Those keys and values should be added in each of the call/requests, to get API to respond. Like API Keys, Languages like Application/json, caches, cookies and Project Information.

**What is a body?**

When we need to send Data to the API, we send it as a request body. Such as POST, PUT we use body to send the resource.

**What is Payload?**

Payload is the data that we send from Post request. like in the request body we send some data in a POST request and in the response body, we receive data from the server. We can send and receive payloads in different formats like JSON, XML etc.

What are the key elements of an API Request?

1. Request line – Where we put endpoint
2. HTTP Headers
3. Message Body- we keep our data/payload to be sent to the server

What is Health Check in API?

To check the minimum viable health of the API, we can send a Get request that responds back if a service is available.

Route.add(‘/api/v1/healthcheck’, Method=’GET’) {

If (userdb.connect() == “Successful”) {

headers={‘http\_status’:200, ‘cache-control’: ‘no-cache’}

body={‘status’: ‘available’}

}

Else {

headers={‘http\_status’:500, ‘cache-control’: ‘no-cache’}

body={‘status’: ‘unavailable’}

}

}

What are the HTTP methods you used for an API?

* **GET**: retrieves a resource
* **POST**: creates a resource
* **PUT**: updates an existing resource
* **PATCH**: update partial resource
* **DELETE**: removes a resource

**Difference between Put and Patch?**

**What is the GET method**

This method we use to retrieve the information from the endpoint for the functionality that has been implemented by developers. We need to use a parameter that has key and value for the specific/filtering info by using a query parameter.

Here we don’t need to use the Request “Body section” because we aren’t adding any data.

We are just retrieving the existing records or data and need to make sure in the Response “Body Section”.

**What is the POST method**

This method sends data to the server to create a new resource. For example Customer info

We need to send some raw data in the Body section. Status code -201(Created resource successfully)

“Id” : 5820582385-044-;

"CreatedAt": "2021-04-15T15:19:13.399Z"

Two post calls with the same data either send an error message or create a duplicate resource in the server.

How do you validate that a new resource is created after sending a successful POST Request?

At first, I check the status code if the POST call is successful. Then I send a GET call with a new ID to retrieve all resources from the same ENDPOINT and check if the new resource is there or not. Also, I will check the database if new resources are there or not.

**What is the PUT method**

This method sends data to the server to update the existing data or record.

Status code-200 (resource updated successfully)

For example: tester will get the following response:

"updatedAt": "2021-04-15T15:19:13.399Z"

**What is DELETE method**

This method deletes the existing data or record that has been requested from the server.

**What are the Status codes you have encountered?**

2xx = successful

* 200 for success
* 201 – Created – once created we used to get an app id.
* 204 — No Content. **it at least tells the client that the request was successful but had no content, its intention is not really to be used to signal a "successful absence".**

4xx = Client error (client sends bad syntax or requirement cannot be fulfilled)

* 400 bad requests, malformed request syntax, if I put some wrong information for negative testing. Contract misses.
* 401 – If we pass invalid Authorization-we get the authorization key and value to get access to the API.

We should conduct an Environment team for Api key missing, and key changes. API key usually changes in 60 days. We need to generate a new API key.

* 403 – forbidden. This means the server understands the request but refuses to authorize it. Normally mistyped URL.
* 404 – The server was connected but not found. The server cannot find the requested resource. There is an issue with the path parameter or Endpoint

**Q: Difference between 204 and 404?**

* 204 — No Content. it at least tells the client that the request was successful but **The server has successfully fulfilled the request and that there is no additional content to send in the response payload body**
* 404 – Not Found. server was connected but not found. The server cannot find the requested resource.

5xx = Server error

* 500 – Internal error/server error….it was successfully entered into the server but somehow server has some error to process the request. It is the application issue the same URL we are using. I used to go to application log. They have some code breaking… probably null point exception…arithmetic exception, number format exception, Index out of bound exception. Whenever we found 500 and 504, we are sending some trace id… that id we put it in JIRA through the trace id the developer will directly get the log.
* 504 – gateway timeout…

**How many ways we can create variables in Postman?**

We can create variables in

1. Global – This variable can be used everywhere in the workspace
2. Environment—like QA Env, Pre-pod env, UAT1, UAT2
3. Collection – Three dots on Collection—Edit-- Variables
4. Data- Three dots on collection – Run collections –Data File. This is my external file like excel.
5. Local – variables from “pre-request Script” or “Tests” inside the request

**What is Environment in Postman? What is the use of the Environment?**

An environment in Postman is a set of key-value pairs.

When we create an environment inside Postman, we can change the value of the key-value pairs and the changes are reflected in our all requests. For Example, we can keep the base URL and API Keys.

**How to use the environment?**

1. We can use the base url and click on Add button as the side of the environment.
2. Then give a name like base URL and initial value and the current value would be <https://API.getpostman.com>.
3. We can use this in our search box like {{baseurl}}/workspace/id

**what is Collection?**

Collection in Postman means a set of API requests preferably of the same type that are grouped and already saved in Postman. This is one of the very important features of Postman. We can run the whole group of request together in just a single click.

You can group your Postman requests into collections to keep your workspace organized, collaborate with teammates, generate API documentation/test suites, and automate request runs.

**What is the monitor?**

Postman Monitoring allows regular check-ins so if anything changes, you'll know right away. Postman API monitoring helps you identify problems with your API including poor performance, response errors, and outages.

**What is the difference between collection and monitor?**

Postman monitors are based on collections. Monitors can be scheduled as frequently as every five minutes and will run through each request in your collection.

What is mapping success?--

Ans: Data mapping is an essential part of **ensuring that in the process of moving data from a source to a destination, data accuracy is maintained**. Good data mapping ensures good data quality in the data warehouse.

What is API Key (Authorization key)?

Ans: An application programming interface key (API key) is **a unique code that is passed into an API to identify the user**. API keys are used to track and control how the API is being used, for example to prevent malicious use or abuse of the API.

**What is the difference between Authorization & Authentication?**

|  |  |  |
| --- | --- | --- |
|  | **Authentication** | **Authorization** |
| 1 | Confirms Who you are- (username, password, one time pin) | Permission to get access to the resource. /folder .. .grant/revoke |
| 2 | Confirms that users are who they say they are | Gives those users permission to access a resource. |

What types of defects did you receive in API?

1. Wrong status code
2. Wrong response

What are the good ways to write the test cases for rest API?

API testing can be done in two ways

* Using Testing tools(Postman, Swagger) to run API

We use swagger for documentation only even though we can run from Swagger. We run our request from Postman.

* Using code to using tools like REST Assured Library

For both cases you need to have scenarios to perform to Test your API according to the functionality of the API. Things that are to be checked are its behavior for the happy path, and negative test cases regarding data, request methods, request body, response, and many more.

Let's take an example of registration API, where there are three fields

{

"username":"test",

"email":"test@gmail.com',

"password":"Test@1234"

}

Test cases for this would be

Test Case -1 Register user : Happy path (everything right)

Register user : Good username value

Register user : Good email address

Register user : Good password

{

"username": "test",

"email": "test@gmail.com',

"password": "Test@1234"

}

Test Case -2: (Negative testing)

Register user : without username value

Register user : without email value

Register user : without password value

{

"username":

"email”:

"password”:

}

Test Case -3 (Negative testing)

Register user : without username field

Register user : without email field

Register user : without password field

{

" test",

“test@gmail.com',

“Test@1234"

}

Scenarios:

Register user : username with short length

Register user : password with short length

Register user : invalid email

Register user : already used email

Register user : with wrong method like PUT instead of POST

Register user : without request body

So in short to write test cases in a good way you have to go for every possible error that could occur

Api sample story:

The user should be able to get a valid response from https://api.getpostman.com/workspaces.

A/C:

Given the user has a valid base URI and authorization(API key)

When the user enter the folder “workspaces”

Then he should receive a valid status code (200)

And get the valid workspaces name and their types.

(……)

Bank Teller API --- <https://teller.io/docs/api/2020-10-12>

* GET /accounts/:id
* GET /accounts/:id/details
* GET /accounts/:id/balances
* GET /accounts/:id/transactions
* GET /accounts/:id/transactions/:transaction\_id
* GET /accounts/:id/payments/zelle/payees
* POST /accounts/:id/payments/zelle/payees
* POST /accounts/:id/payments/zelle

**Rest Assured**

**Postman Test script Examples:** [**https://learning.postman.com/docs/writing-scripts/script-references/test-examples/**](https://learning.postman.com/docs/writing-scripts/script-references/test-examples/)

**Question: what is given(), When(), Then() methods for?**

Given() is referred as pre-condition. We keep baseURL, header in this section.

When() represents the action, Here our action is to execute the get, post, put, delete request.

Then() represents the outcome/result of the action(like response status code, body, header, cookies) that we took also we can validate the outcome. The assert and Hamcrest matcher method will come over here.

**How do you extract JSON response?**

We use JSONpath Online evaluator/Json Path Finder manually and also build in JsonPath library in RestAssured.

JsonPath jp = res.JsonPath();

String firstworkpsace = jp.get(workspace[0].name);

System.out.println(firstworkspace)

**How many ways we can send payload through RestAssured?**

1. Jsonfile path We send the Json file as under body

2. POJO

3. HashMap

**How to create a Json object payload using POJO?**

**So POJO stands for Plain Old Java Object. So for each object we create separate class and into the class we keep our keys as private variables. Then we use getter and setter method for each key. Also we use a constructor for the class and keep those keys as parameter. We also use “this” keyword to merge those keys with the private variables.**

**In my RestAssured class we create objects of the constructors where we initialize those keys like the value we want to pass for those keys. That object we send as my payload in my RestAssured.**

**What is POJO serialize?**

When a Java object being converted to JSON Object using POJO is called serialize.

**When you send a post request in Rest Assured what do you validate?**

.body("message", equalTo("Successfully! Record has been added."))

Status code,

Size,

Null or no null,

**How to send a complex payload for a Post API Using POJO classes?**

**V.V.I. Question: what are the validation you do in API?**

1) Status code (by AssertThat)

2) We validate the status as well like- success or created or ok etc. (by get text)

3) We validate response body using **assert utility**, we use **has item**, has items, we use **equal to**, **has size**, from Hamcrest as well.

4) **Not null or null** validation (particular data in the body is null or not)-- Assert.*assertNotNull*(**"workspace.id"**);

Assert.assertNull(“workspace.description”)

5)**Length of the body/ Length of the data** is greater than 0 or not.

6)We validate **API key**. (negative test)

7)We run **health check** every day. This is a **get request**.

**What are the different Authentication method in RestAssured?**

1. Basic Authentication

Username and Password Like

Given ().

Param (“page” “2”).

Auth (). basic(“bashir”, “bashir123”)

1. Digest Authentication
2. Form Authentication
3. API Key
4. OAuth1 and OAuth2 (notes being ready)

**How to check response time for an API?**

In Rest Assured I can use getTime () method– this will return the response time in millisecond.

Example:

res. getTime ().

How to log response for a POST API?

Using when (). Body ().log (). all () --------- for request

Using Then (). log (). all () -------------------for response

**Interview Questions:**

1. **What do you test in API…**
2. **Data accuracy -that means responses are correct or not**
3. **https status code**
4. **error codes**
5. **authorization check**
6. **non-functional testing like security testing, performance testing(response time)**
7. **healthcheck**
8. what is top-down and bottom up approaching in Web Serivce
9. What is the big challenges in API Testing?

Lot of times we have received server error… that is nothing to do from our Testers end. Also getting 400 bad requests while giving right input.

Also all query parameter needs to be inserted in order.

1. Common API Bugs and Test Cases for API Testing  
     
   1. Wrong Error Codes.(Develops made some mistakes and put wrong status code for Different request).  
     
   Following are correct status Code :  
   Post Request - 201 ,Created (The request succeeded, and a new resource was created as a result.)  
   Get Request - 200 , Successful (GET: The resource has been fetched and transmitted in the message body.)  
   Patch/PUT - 200 , Successful(The resource describing the result of the action is transmitted in the message body.)  
   Delete - 204 , (No Content ) status code if the action has been enacted and no further information is to be supplied.)  
     
   Client Error responses:  
     
   400 Bad Request - The server cannot or will not process the request due to something that is perceived to be a client error   
   (e.g., malformed request syntax, invalid request message framing, or deceptive request routing).  
     
   401 Unauthorized -Although the HTTP standard specifies "unauthorized", semantically this response means "unauthenticated".  
   That is, the client must authenticate itself to get the requested response.  
     
   403 Forbidden -  
   The client does not have access rights to the content; that is, it is unauthorized, so the server is refusing to give the requested resource.   
     
   404 Not Found -  
   The server cannot find the requested resource. In the browser, this means the URL is not recognized.  
   In an API, this can also mean that the endpoint is valid but the resource itself does not exist.   
     
   Server error responses:  
     
   500 Internal Server Error:  
   The request method is not supported by the server and cannot be handled.   
     
   502 Bad Gateway -  
   This error response means that the server, while working as a gateway to get a response needed to handle the request, got an invalid response.   
     
   503 Service Unavailable: The server is not ready to handle the request.   
     
   504 Gateway Timeout -This error response is given when the server is acting as a gateway and cannot get a response in time.  
     
   2. Missing Keys  
   You have certain keys which are important , like Id should be integer or not null.  
   But developer missed that key.  
     
   3.Empty Post or Patch : Empty Post or Patch (Update) are not Handled properly.  
     
   4. Black Box: We can do Black box Testing in API testing, like there is some input field is there   
   Age, we can enter empty field, less than minimum value , within the range, more than maximum value.  
     
   5. Keys Verification: If we have JSON ,XML APis we should verify its that all the keys are coming.  
     
   6. Schema: Have to test JSON, XML Schema validation.  
     
   7.Verify Schema and Fields: Verify the JSON Schema validation , Verify the Field type, Verify the Mandatory fields  
     
   8.Negative test: Verify the Response Header & Negative Test cases response   
     
   9. Error Code Handling: Verify that How the Error codes are handled  
     
   10.HTTP status Code: Verify the response HTTP status code  
     
   11. Payload: Valid Response Payload  
     
   12. Chaining : Chaining Request verification   
     
   13.Data Parameters: Verification of API with Data parameters  
     
   14. End to End: End to End CRUD flows Test cases   
     
   15. Database: Database integrity Test Cases  
     
   16.File Upload: File Upload test cases(with different extension file like pdf, txt, exe etc).