

Rahul shetty API automation

15 January 2022 11:59 PM

Consider a scenario when marriot hotel has front end developed in Angular and backend developed in Java. It is difficult for two components of different technology to interact with each other. So API is used. The same is also applicable in case booking.com access marriot hotel rooms. The API is exposed. The request is passed as form of JSON/XML and response is received as JSON/XML

Example google.com/maps where google.com is a API endpoint and maps is a resource

Before API testing ask for API contract. It contains all documentation regarding API

Request Endpoint=BaseURL+resource+Query params

An endpoint is one end of a communication channel. When an API interacts with another system, the touchpoints of this communication are considered endpoints. For APIs, an endpoint can include a URL of a server or service. Each endpoint is the location from which APIs can access the resources they need to carry out their function.

URI=(URL+URN)

Where URI=Uniform Resource identifier (can use both name and location or both)

URL=Uniform resource locator(Use address to locate)

URN=Uniform Resource name(Use name to locate)

What is Rest Assured:

It is java based library which is used to test RESTful webservice/API's

Dependencies required for API automation

1)RestAssured 2)TestNG 3)Hamcrest

Note:On pasting a string with "(quotes)", eclipse automatically add escape char. This is due to setting in editor ide

To replace a value with a variable use "+placeid+" (Notice the " and + position)

Given() method of RestAssured is static, so we won't get autosuggestion. Hence import manually "import static io.restassured.RestAssured.*;"
Similarly equalTo() method of hamcrest is static so import import static org.hamcrest.Matchers.*;

Mock your server response when the development is not ready yet.

Use size() method to get the count of json Array. This is applicable only for Array

Note:In json parsing get() method can be used for returning all datatypes.

In RestAssured methods

//given - all input details

//when - Submit the API -resource, http method

//Then - validate the response

There are two ways in which we can assert a response.

1)Directly with then() chaining with assertAll and body()

2)Extracting the whole json response and parsing and doing the validation

Observed on Own:When using get methods of Json parser, if the passed argument is not matching with json then null is returned

Payload:

In simple words, the payload means body in the HTTP request and response message. It's optional and depends on the HTTP method name

Advanced payload creation strategies:

1)Dynamically build json payload with external/user data(Pass as argument to a method which returns json string)

2)Parameterize the API test with multiple data sets(Use DataFactory)

3)How to send static Json files(Payload) directly into Post method of Rest Assured(Use json File)-Convert the content of file to string->convert it into Byte->Byte into string

eg

```
body(new String(Files.readAllBytes(Paths.get("C:\\Users\\kkrid\\eclipse-workspace\\RestAPI\\nio.json"))))
```

4)Feed Json payload From Excel using Hashmap

Steps to install JIRA

Download and run the JIRA server

Account used :hhh@mailinator.com username:hhh pwd:kiranji@123

<http://localhost:8080/>

C:\Users\kkrid\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Jira Software

To use JIRA cloud use basic auth as authentication and give API token as password. Email address as username

The API token can be generated at [Atlassian account](#)

Path parameter vs query params

Path parameters are variables in a URL path. They are used to point to a specific resource within a collection

We can use query parameters to control what data is returned in endpoint resources. It appears at the end of the URL after the question mark (?) and helps us to control the set of items and properties in responses, and the order of the items returned.

There is another way to parse JSON apart from jsonPath ie SessionFilter sf=new SessionFilter();

TO set and access the session use it in given

SessionFilter sessionFilter=new SessionFilter();

```
given().log().all().filter(sessionFilter)
```

When Passing path param use `pathParam("KEY", "10202")` in given. similarly the Key must be caught in resource passed
`le)post("/rest/api/2/issue/{KEY}/comment")`

TO pass file/attachement in API request use multipart
`multiPart("file", new File("C:\\Users\\kkrid\\eclipse-workspace\\RestAPI\\json.txt"))` in given() and also use header("Content-Type", "multipart/form-data") in given

Note interview questions: To bypass website which does not have certificates then use `relaxedHTTPSValidation()` in given

OAuth 2.0

OAuth 2.0 is the industry-standard protocol for authorization
It comes with multiple grant types (ie an umbrella under which many are present)
Authorisation code and client credentials are the most commonly used grant types

Why application rely on other (Google/Facebook) for authentication

- 1) No data breach headaches for the app
- 2) Need not maintain user profile data
- 3) This allow richer website to talk to each other

Some terminologies (When Book my show uses google/fb authentication)

Client=BookMyshow
ClientID=Google allocated to identify bookmyshow
Client secret ID=password
Resource Owner=Human
Resource/Authorisation server=Google

Steps involved

- 1) User sign into Google Authorization server and get code (OTP kind of)
 - 2) Application will use this code to hit google resource server in backend to get (Access token, Firstname, lastname, Email, etc..)
 - 3) Application grants access to user by validating accesstoken
- Note: The first bulletin Authorization server and in second resource server

`testuser1997insta@gmail.com kiranji@123`

By Default RestAssured will encode the URL if it contains any special character (ie convert special character to Numeric value). TO avoid it urlencoding is set as false

URL encoding is a mechanism for translating unprintable or special characters to a universally accepted format by web servers and browsers. Mostly to Ascii

The second grant type "Client credentials" is used when application request its own data.

Eg. Rahul shetty application needs to fetch the details of tweet tagged with rahul shetty.
In this type User/human is not involved and auth code or the first step is not required

POJO class

Serialization in RestAssured context is a process of converting a java object into Request body (payload)
RestAssured also supports deserialisation by converting Response body back to java object

Advantages:

- 1) Easy to parse and extract response (JSON/XML) values if they are wrapped as java object
 - 2) User friendly method can be created which makes code more readable (ie getters/setters)
- POJO-Plain old java object

Libraries required:

For Json: Jackson, Jackson 2, Gson
For XML: JAXB

`expect().defaultParser(parser.json)` in given method for declaring the expected response type

Extract as class object ie `extract().as(JSONResponse.class);`
In our exercise we are using jackson library

In pojo class all json key must be declared .if not then the deserialisation will not work

specbuilder is used for reusing the same code in multiple tc
Can be used in Request and Response. Usually end with build method

```
RequestSpecification requestSpec= new RequestSpecBuilder().setBaseUrl("https://rahulshettyacademy.com").setContentType(ContentType.JSON).addQueryParam("key", "qaclick123")
    .build();
```

```
ResponseSpecification responseSpec=new ResponseSpecBuilder().expectStatusCode(200).expectContentType(ContentType.JSON).build();
```

```
RequestSpecification req=given().spec(requestSpec).body(addplace);
```

```
req.post("/maps/api/place/add/json").then().spec(responseSpec).log().all().extract().response();
```

Cucumber:

Full path can be given for feature package
But in case of stepdefinition glue only package name is enough

TO log the request and response in a file use

```
addFilter(RequestLoggingFilter.logRequestTo(ps)).addFilter(ResponseLoggingFilter.logResponseTo(ps)) where ps is object of printstream
PrintStream ps = new PrintStream(new FileOutputStream(System.getProperty("user.dir") + "\\log.txt"));
requestSpec = new RequestSpecBuilder().setBaseUrl("https://rahulshettyacademy.com")
.addFilter(RequestLoggingFilter.logRequestTo(ps)).addFilter(ResponseLoggingFilter.logResponseTo(ps))
.setContentType(ContentType.JSON).addQueryParam("key", "qacllick123").build();
```

It is good practice to use ENUM for declaring resource endpoint. TO learn about enum: <https://www.geeksforgeeks.org/enum-in-java/>

```
public enum APIResources {

    addplace("/maps/api/place/add/json"), deleteplace("/maps/api/place/delete/json"),
    getplace("/maps/api/place/get/json"), updateplace("/maps/api/place/update/json");

    String resource;

    APIResources(String resource) {
        // TODO Auto-generated constructor stub
        this.resource = resource;
    }

    public String getResource() {
        return resource;
    }

}
```

To access the enum

```
APIResources apiResource=APIResources.valueOf(string);
System.out.println("RESOURCE "+apiResource.getResource());
```

TO provide cucumber option in maven command

Mvn test -Dcucumber.options="--tags @Addplace"(Only in old version of cucumber) .In new version use
commands mvn test -Dcucumber.filter.tags="@smoke"

Third party report Cucumber maven report <https://github.com/damianszczepanik/maven-cucumber-reporting>

This report reads the json file and creates the report. TO generate the json file use
plugin = "json:target/jsonReports/cucumber-report.json" (Inside target folder the subfolder a mentioned is created)

Copy and paste the dependency as mentioned in the report documentation above given link. Remove optional
classificationDirectory

In the configuration tag notice the input and outputDirectory. The input directory should point to json file generated by using plugin statement

TO run in maven use mvn verify command (Rahul shetty uses mvn test verify) ,ie both test and verify together

Note: multiple tags can be given to a scenario in feature file

API Interview questions

<https://www.interviewbit.com/api-testing-interview-questions/>

Difference between API Documentation, Specification, Definition

Documentation: The documentation tells developers and other API consumers how to use the API. After all, how can your API be successful if developers don't know how to use it?

Specification :An API specification provides a broad understanding of the functionality of the API and the expected results. The specification is largely about the design of the API or your design philosophy. API design and functionality are key factors when choosing to integrate an API with an application.

Definition :API definition is about machine consumption of an API and providing a machine-readable format for use by automated tools like automatic API documentation and SDK generators

Rahul interview questions video

1)Challenges in API testing

API documentation,

Access to DB,(API belong to third party and db access is not provided.You cannot verify value from DB),

Authoriation overhead

2)Difference between POST and PUT

POST request=Creating new object on the server

PUT=update the object in the server with a new value

3)Commonly used HTTP methods

GET=Retrieve data from server

POST=Add data in existing file or resource in a server

PUT=It lets you replace an existing file.

Delete=Delete data from a server

4)List out commonly used Authentication technique used in API

Session/cookie based

Basic Authentication

Digest Authentication

OAuth

5)Why API testing is most suitable for Automation

Lightweight(Eg Adding employer detail via UI involving login navigation and data entry vs hitting an API to add details)

Stable not flaky due to internet conditions

6)What is REST API

REST Stands for Representational state Transfer.It is a set of functions helping developer in performing request and receive response.Interaction is made through HTTP protocol in REST API

7)What needs to be verified in API testing

*Accuracy of Data

*Verify HTTP status code

*Verify response time

*Error code in API return any error

*Authorization

*Non functional testing such as performance and security testing

8)What is path param and query param

Path parameters are variables in a URL path

Rahulshetty.com/order/11234?location=India;

Path param is without ? and query param is with question.

9)What are core components of an HTTP request

*)HTTP Request methods like "GET",POST,PUT,DELETE

*)Base Uniform Resource Identifier(URI)

*)Resource and parameters

*)Request header which carries metadata(As key-value pairs) for the Http request

*)Request body which indicate the message content or resource representation

10)What could be HTTP method for the below scenario:

An Api which has Endpoint,parameter,Headers,cookies and payload.

Since payload is involved it could be either of POST,PUT,DELETE .

GET cannot have payload

11)Difference between UI and API Testing

UI testing involves GUI.Look and feel of the app.Testing how user interact with app elements

API allows communicates between two software system.API testing also called as backend testing

12)What protocol is used by RestFUI services

It uses HTTP protocol.It uses HTTP protocol as medium of communication between server and client

13)What are SOAP services

Soap stands for simple object access protocol.It is a xml based messaging protocol.It helps in exchanging information among computers

14)How do you represent a resource in REST

Using HTTP methods

15)Can you use GET request instead of PUT to create a resource

No GET request has only read only rights.It enables to retrieve data from a server but not create a resource. PUT or POST methods are used to create a resource

16)Can you user POST request instead of PUT to create a resource

Yes we can.POST is superset of all HTTP methods expect GET

17)What is payload in Restful web services

Payload/Body is secured input data which is sent to API to process the request

Payload is represented in JSON generally in REST API

18)Difference between query and path params in API request endpoint

19)What is REST Assured

It is a java library which is used Can automate REST API

20)How would we define API details in Rest Assured automation

We shall define all API details and sent it to server using Given ,When , Then methods

21)what is json serialisation & Deserialisation in Rest Assured

Serialisation in rest assured context is the process of coverting the java object into request payload

Deserialisation is the process of converting response body back into java object

22)What are few common JSON parsing techniques in Rest Assured

Json Path class

Deserialisation of json using POJO class

23)How would you send attachments to an API

Using Multipart method

24)Different status codes and their description

200 OK The request was successfully completed

201 Created A new resource was successfully created

400 Bad request The request was invalid(Resource found details sent are invalid).Eg payload missing

401 unauthorised The request dinot have authentication token or it got expired

403 Forbidden The client did not have permission to access the resource

404 Not found The requested resource not found(Different from 400)

405 Method not allowed The HTTP method was not supported by resource.Eg:DELETE cannot be used in place of GET

409 conflict

500 Internal server error The request was not completed due to internal server error

503 Service unavailable The server was not available

25)Get the json path of a variable from a json

26)Soap vs REST(own question)

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 . In SOAP, the WSDL file(Web Service Description Language) provides the client with the necessary information which can be used to understand what services the web service can offer.	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.(Presentation layer, Network layer)
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.

27)Challenges of Soap And REST and its comparison(Own question)

<https://www.guru99.com/comparison-between-web-services.html#:~:text=KEY%20DIFFERENCE,stands%20for%20Representational%20State%20Transfer.&text=Comparing%20SOAP%20vs%20REST%20API,can%20make%20use%20of%20SOAP.>

28)Difference between POST and PUT

POST=Create a new entity

PUT=Create and update a new entity

Naveen:

1)What is a API

AN API is a software intermediary that enables two software systems to communicate with each other

EG:Google maps API, Twitter API, Amazon advertising API, Youtube API

2)Difference between WebServices and API

<https://www.javatpoint.com/soapui-web-services-vs-web-api>

All webservises are API but not all API are webservices

WebServices might not contain all specifications and cannot perform all task that API would perform

WebServices uses only three styles SOAP, REST, and XML-RPC for communication where as API may be exposed in many ways

A webServices need network to operate where as API don't need network

3)What are limits of API usage?

Many API have certain limit set by provider. No of request hits per second

4)Architectural styles for creating an API

HTTP for client server communication

XML/JSON as formatting language

Simple URI as the address of the service

Stateless communication

5)Who can use Web API

Any client which supports HTTP verbs. it does not require any configuration. portable devices such as mobile, laptop can use web API

6)What is API testing?

API testing is a kind of software testing which determines if developed API meets expectation regarding the functionality, reliability, performance, security of the application (Can be internal API (Used in same company, order team API with user management team) or external API)

7)What are advantages of API testing

Test of core functionality

Time effective

Language independent

Easy integration with GUI. For example API test and GUI test can be clubbed together.

8)Some protocols used in API testing

JMS (java messaging service), REST, SOAP, HTTP, UDDI

9)What is test env in API testing

10)What are principals of API design

Setup: Create objects, start services, initialise data

Execution: Steps to apply API or scenario, including logging

Verification: Assertion

Reporting: Pass, Fail, Blocked

Clean up: Pre state data

11)What are common API testing types

Validation Testing

Functional testing

Load testing

Runtime/Error detection

Security testing

Penetration testing

Interoperability testing

12)What is procedure to perform API testing

Choose the suite to add the API test case
Choose the test development mode(Env)
Demand the development of testcase for the required API methods(This is from DEV side)
Configure the control parameters of the application and test conditions
Configure method validation
Execute the API test
Check test reports
Arrange all API test cases

13)What must be checked when performing API testing

Accuracy of data
Schema validation
HTTP status code
Data Type,validations,order and completeness
Authorisation checks
Implementation of response timeout
Error codes in case of API returns
Non functional testing such as performance,security

14)Best Approach to perform an API testing

Defining the correct input parameters
Verifying the call of mixture of two or more added value parameter(Eg:Retrieve by ID and Retrieve by name,ID)
Basic functionality and Scope of API program
Writing appropriate API test case and making sure of testing techniques such as equivalence class ,Boundary value analysis
Test case execution
Comparing test result with expected result
Verifying the API behaviour under condition such as connection to a file/Database

15)What are tools for API testing

POSTMAN,SoapUI,Tosca,Apigee,Rest Assured,HTTP client etc..

16)Difference between AP and Unit testing

Conducted by QA team
Mostly blackbox testing
Aimed to access the full functionality of the system for it will be employed by the end user(External developers who will use your API)
Often run after build is ready and authors(QA) does not have access to source code

Unit testing:

Conducted by dev team
White box testing
Used to verify each unit in isolation works as expected or not
Each of the code modules must be endured to pass the unit test before being built by developers

17)Difference between API and UI test

Refer answer from some blog.Note:API can communicate with different UI.Eg.Amazon web,Amazon app andriod, Amazon app ios are all different UI but the logic is defined in API also called as backend.The Ui are called front end

18)What are major challenges in API testing

Parameter selection
Parameter combination
Call sequencing
Output verification and validation
Providing/Identifying input values since GUI is not available.This can be solved only when proper API documentation is available

19)Testing methods under API Testing

Unit and Functional testing
Load Testing
Usability and reliability testing
Security and penetration testing

Automation testing to create and run scripts that require regular API calls
END to end integration and Web UI testing
API documentation testing to determine the efficiency and effectiveness

20)Why API testing considered as most suitable form of automation testing

It verifies all functional part of the system under test very effectively(If backend is working then front end will work for sure)

It provides most stable interface

It is easier to maintain and provides fast feedback

21)What are common API errors

<https://blog.runscope.com/posts/6-common-api-errors>

22)What kinds of Bugs that API testing would find

Missing or duplicate functionality

Fail to handle error conditions gracefully(Proper error/Warning message)

Stress and load Issues

Reliability

Security

Not implemented errors

Performance

Multithreading(In shopping site when one item left,The first hit request should add it to cart and then next request should not be able to do it)

23)Difference between OAuth vs OAuth2.0(own question not Naveen)

<https://naveenautomationlabs.com/oauth-2-0/#:~:text=Authentication%20%26%20Signatures%3A-,OAuth1.,and%20is%20easier%20to%20implement.>

24)Difference between Authentication vs authorization(own question not Naveen)

<https://www.geeksforgeeks.org/difference-between-authentication-and-authorization/>

25)What is API documentation

The API documentation is a complete,technical writing giving instruction on how to effectively use and integrate with an API.It is complete reference manual which has complete information need to work with an API.For example when indigo exposes with booking API ,the dev at make my trip needs to understand how it works and then only then can integrate

26)API documentation template

Swagger(Popular), Miredot, slate,Flatdoc,API blueprint,RestDoc

27)When writing an API document what must be considered

Source of the content

Document plan or sketch

Delivery layout

Information needed for every function of the document

Automatic document creation programs

28)How often API's are changed and more importantly deprecated

It depends on project to project.For example createuser API accepts name and Email address as params.Now it is also asking for gender to be passed.

29)What is REST

30)What is Restful webServices

31)What is Resource in API

32)Most popular way to represent a resource in REST

Text,JSON,XML.

33)Different Authentication: <https://blog.restcase.com/4-most-used-rest-api-authentication-methods/>

34)HTTP verbs <https://restfulapi.net/http-methods/>

Which is better Postman or RestAssured

Rest assured suitable for end to end testing which involves web test,some Api Test and Db validation

Rest assured has framework advantage (utilities) Code Reusability

Postman uses js where as Rest assured uses many language binding.May become a problem if the tester is not aware of javascript.

Postman no support for parallel execution

Customize Reports: As there are a lot of open source reporting tools. We can customize reports for the REST-assured. But where in Postman doesn't allow us to customize the reports

Designing a Data-Driven framework: For Postman automation runner we can provide only with one data file for each collection. But for REST-assured there is no limitation for this.

CI/CD integration: This is not any drawback. We can integrate both to any CI tool. For Postman we can integrate into Jenkins using Newman tool. For more info check [here](#).

And moreover, REST-assured provides us DSL (Domain Specific Language). So that our tests can be behavior driven.

So that we can conclude for less maintenance and more efficiency its better to prefer REST-assured than Postman for automating these RESTful web services

Youtube(RD automation)

Difference between Rest and soap API

What you will test in API

Challenges in API testing

Diff top down and Bottom up approach in API testing(Which is preferred)

Which Test is executed first. Blackbox or white box testing

Ans :I will ask the dev about unit test coverage.If not covered Then I will perform whitebox together with him and continue blackbox.If unit test are covered then I will start blackbox

What is workbench <https://blog.qatestlab.com/2011/10/28/workbench-concept-in-software-testing/>

Pratice Rest API

<https://reqres.in/>

<http://fakereestapi.azurewebsites.net/> swagger API doc

<https://resttesttest.com/>

<https://restful-booker.herokuapp.com/apidoc/index.html#api-Ping-Ping>

Difference between HTTP and HTTPS

HTTP

- 1)Insecured
- 2)Request and response over network are sent as plain text
- 3)Light weight
- 4)Default port 8080
- 5)Acts on layer 7-Application layer

HTTPS:

- 1)secured by encryption
- 2)port 443
- 3)Certificate installed in server. It will be verified by certificate authority present in browser
- 4)Heavier than HTTP
 - Uses SSL-Secure socket layer
 - TLS-Transport layer security

<https://www.guru99.com/difference-http-vs-https.html>

<https://www.cloudflare.com/learning/ssl/how-does-public-key-encryption-work/>

Postman

Postman can be directly downloaded and installed.To install newman you need node to be installed

TO set the workflow or chaining use postman.setNextRequest("Delete user")

cURL is the most used command line tool for making API calls. It stands for client URL

We can import or convert curl to postman request and vice versa

Curl url -o filename (This command is used to output the response to a file)

Newman is built on top of node. Newman is a command-line Collection Runner for Postman. It enables you to run and test a Postman Collection directly from the command line

First node needs to be installed

TO install new man npm install-g newman

TO run collection newman run my collection.json Refer:<https://learning.postman.com/docs/running-collections/using-newman-cli/command-line-integration-with-newman/>

Html report in postman <https://www.npmjs.com/package/newman-reporter-html> .This report will be available in newman folder by default

Advance htmlExtra report

Install :npm install -g newman-reporter-htmlextra

usage: newman run collection.json -r htmlextra

<https://www.npmjs.com/package/newman-reporter-htmlextra>

TO run Collection in jenkins ,Got to build section and select windows batch command and type the newman run collection.json (as required)

If the console in jenkins is not showing newman run in proper format(special characters appearing) then start jenkins server with file encoding utf8

java -Dfile.encoding=UTF8 -jar jenkins.war.

Github and postman integration

1)Export the collection as json. Push the json to github

2)select source code management option as git in jenkins job conf

The problem is that everytime changes in postman ,we need export json and push everytime to github .But using postman paid version we can directly integrate postman and github and on every changes in postman it is pushed to github.

Another option is to use ci with postman api

1)we need postman key and collection/env id

To generate postman key ,go to account settings and generate postman key

2)We need collection id which can be retrieved via postman api <https://documenter.getpostman.com/view/631643/JsLs/#8ca888b7-ef54-f3b4-312f-3f3e2e2cf04e>

Syntax : refer here <https://learning.postman.com/docs/running-collections/using-newman-cli/continuous-integration/>

newman run https://api.getpostman.com/collections/{{collection_uid}}?apikey={{postman-api-key-here}}

To skip a postman test use syntax pm.test.skip

Suggestion :use this skip with if block

```
pm.test.skip("Your test name", function () {  
  var jsonData = pm.response.json();  
  pm.expect(jsonData.data[5].first_name).to.eq("Tracey");  
});
```

Collaboration in Postman

1)Create a fork of collection you want to have in your workspace(For practice purpose we can fork a existing collection from our workspace)

2)Create a pull request

3)approve the pull request

4)merge the changes

Use this json server to create a API in 5 mins for practice purpose <https://www.npmjs.com/package/json-server>

To stringify and to parse as json use `Json.stringify()` and `Json.parse()`;

Eg

```
pm.test("set response variable",function(){  
  var jsonData=pm.response.json();  
  pm.globals.set("jsonData",JSON.stringify(jsonData));  
});
```

```
pm.test("Get response ",function(){  
  var jsonData=JSON.parse (pm.globals.get("jsonData"));  
  console.log(jsonData.per_page);  
});
```

Working with XML response in postman

SOAP API works with XML. You can refer endpoint here <https://www.w3schools.com/xml/tempconvert.aspx>

Or Multiple public soap endpoint <https://www.postman.com/cs-demo/workspace/public-soap-apis/overview>

Note : Make sure header is added Content-Type : text/xml

TO convert xml to json `var jsonObject = xml2Json(responseBody)`; here `responseBody` is keyword/inbuilt variable. Alternatively you can select from code snippet as well given in right
Compare the response by printing/consoleing and use the below syntax for iteration

```
var jsonObject = xml2Json(responseBody);  
console.log(jsonObject ["soap:Envelope"]["soap:Body"]["m:ListOfCountryNamesByNameResponse"]["m:ListOfCountryNamesByNameResult"]["m:tCountryCodeAndName"]["0"]  
["m:sISOCode"])
```

Notice how in square brackets it is given

Import collection from swagger

We can import collection from swagger.We need to copy the json from swagger

Mock a server in postman.

Mock servers simulate an API by returning predefined data, enabling you to develop or test against an API before it's production-ready

Steps:

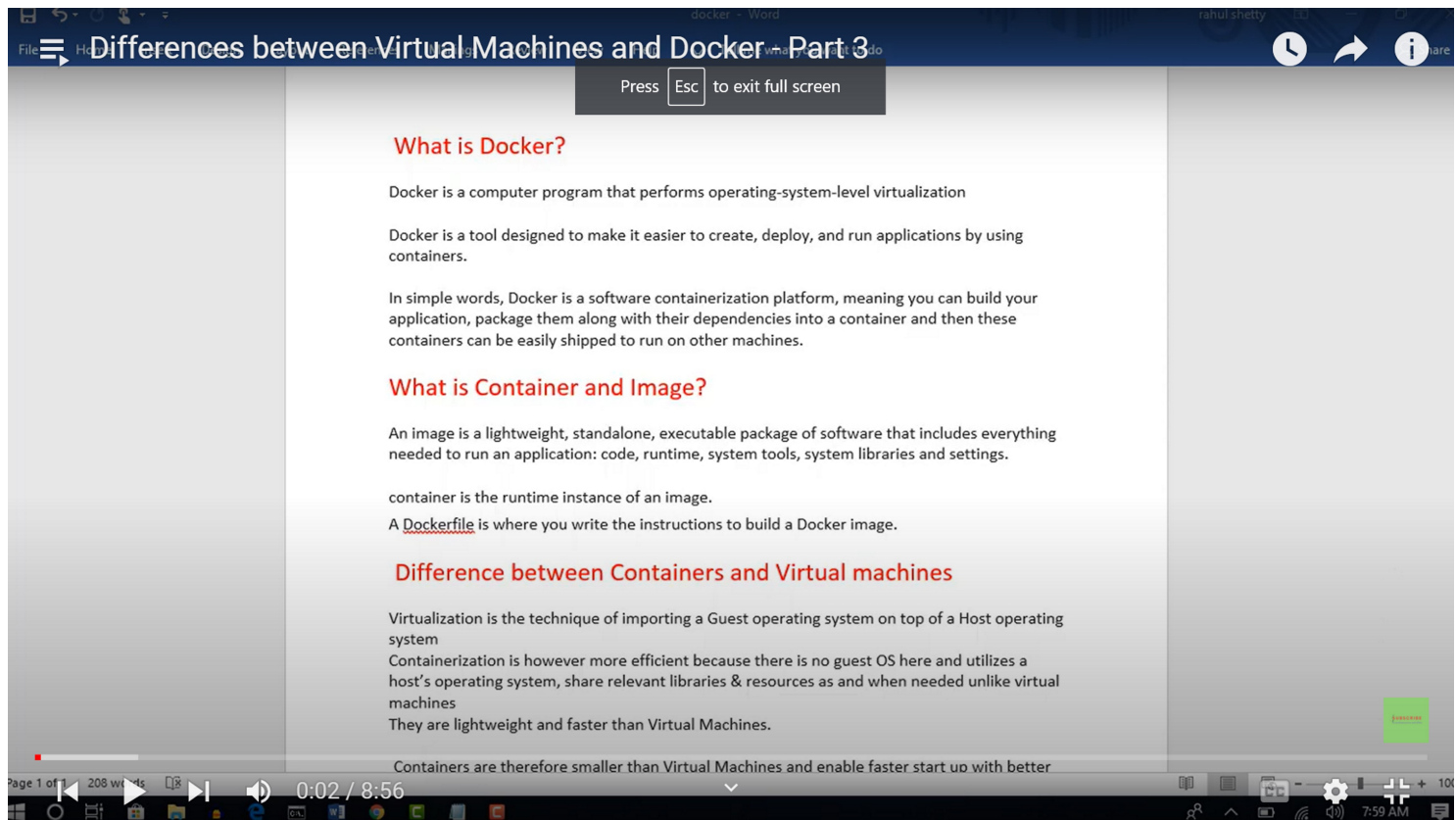
1)create new mock server. Define the resource endpoint and response.

2)A collection will be created automatically.We can write our testcase using mock server before the API is ready and swap one real API is available

Docker (Automation step by step)

Docker is the world's leading software container platform Docker makes the process of application deployment very easy and efficient and resolves a lot of issues related to deploying applications Docker is a tool designed to make it easier to deploy and run applications by using containers Docker gives you a standard way of packaging your application with all its dependencies in a container Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package. Understand Docker with analogy of the Shipping industry How a real world problem was resolved using container

Benefits of using Docker Build app only once No worries that the application will not perform the same way it did on testing env Portability Version Control Isolation Productivity Docker simplifies DevOps



Commands to remember

Docker ps

Docker images

Docker run (pulls image and start container)

Docker stop <container ID>

Docker start <container id>

Docker ps -a (list all running and stopped containers)

docker run -d -p 4444:4444 --shm-size="2g" selenium/standalone-chrome:4.1.2-20220217

Here -d represent detached mode(runs in background) -p 4444:4444(first port represent the host port and second port represents container port)

If same image of different version running we need to change the port for a container mapping

Eg

docker run -d -p 4444:4444 --shm-size="2g" selenium/standalone-chrome:4.1.2-20220217

docker run -d -p 4443:4444 --shm-size="2g" selenium/standalone-chrome:4.1.2-20220217

Example:

```
public void googleSearchDocker() throws InterruptedException, MalformedURLException {
```

```
    URL url=new URL("http://localhost:4444/");
```

```
    DesiredCapabilities cap=new DesiredCapabilities();
```

```
    cap.setBrowserName("chrome");
```

```
    RemoteWebDriver driver=new RemoteWebDriver(url, cap);
```

```
    driver.get("https://www.google.com/");
```

```
    System.out.println(driver.getTitle());
```

```
}
```

Docker logs <container id>

Docker logs <name of container> (Name can be know via docker ps command)

TO give name to a container docker run -d -p 4444:4444 --name Selenium --shm-size="2g" selenium/standalone-chrome:latest (notice --name selenium)

TO go into a container terminal docker exec -it ae025735d055 /bin/bash where ae025735d055 is container id. Type ls ,pwd or exit to quit

Docker file=Blue print for creating docker images

Docker Basic Commands Step by Step for Beginners

Basic : docker version

: docker -v

: docker info

: docker --help (eg: docker-compose --help)

: docker login

----- Images

: docker images

: docker pull

: docker rmi

----- Containers

: docker ps

: docker run

: docker start

: docker stop

----- System

: docker stats

: docker system df command displays information regarding the amount of disk space used by the docker daemon.

: docker system prune Remove all unused containers, networks, images (both dangling and unreferenced), and optionally, volumes.

A dangling image is a image that is not tagged or referenced by any container. Docker images consist of multiple layers. Dangling images are layers that have no relationship to any tagged images. They no longer serve a purpose and consume disk space.

What are Images

Docker Images are templates used to create Docker containers

Container is a running instance of image Where are Images Stored Registries (e.g. docker hub) Can be stored locally or remote

: docker images --help

: docker pull image

: docker images

: docker images -q

: docker images -f "dangling=false"

: docker images -f "dangling=false" -q

: docker run image

: docker rmi image

: docker rmi -f image

: docker inspect Return low-level information on Docker objects

: docker history imageName

What are Containers: Containers are running instances of Docker Images

COMMANDS

: docker ps

: docker run ImageName

: docker start ContainerName/ID

: docker stop ContainerName/ID

: docker pause ContainerName/ID

: docker unpause ContainerName/ID

: docker top ContainerName/ID Display the running processes of a container

: docker stats ContainerName/ID

: docker attach ContainerName/ID

: docker kill ContainerName/ID

: docker rm ContainerName/ID

: docker history ImageName/ID

Notes

A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.

Features of Containers: Are lightweight Fewer resources are used Booting of containers is very fast Can start, stop, kill, remove containers easily and quickly Operating System resources can be shared within Docker Containers run on the same machine sharing the same Operating system Kernel, this makes it faster You can use the command docker container create to create a container in stopped state

To use jenkins in docker <https://github.com/jenkinsci/docker/blob/master/README.md>

Docker file:

A text file with instructions to build image .Automation of Docker Image Creation

Steps to create docker file

Step 1 : Create a file named Dockerfile

Step 2 : Add instructions in Dockerfile

Step 3 : Build dockerfile to create image

Step 4 : Run image to create container

Example:

#getting base image

FROM ubuntu

MAINTAINER Kiranji

RUN apt-get update

CMD ["echo","hello first docker image"]

Here cmd will be exec when container is created where as run will executed during docker build. FROM is to indicate the base image.It can be scratch(https://hub.docker.com/_/scratch?tab=description)

COMMANDS

: docker build

: docker build -t ImageName:Tag directoryOfDockerfile (eg: docker build -t myimage:1.0 . Notice Here "." is used in place of docker file location since the command is executed from inside the folder or else give full path)

: docker run image

Docker compose

tool for defining & running multi-container docker applications

: use yaml files to configure application services (docker-compose.yml)
: can start all services with a single command : docker compose up
: can stop all services with a single command : docker compose down
: can scale up selected services when required

Step 1 : install docker compose
(already installed on windows and mac with docker)
docker-compose -v

2 Ways to install if not available by default

1. [https://github.com/docker/compose/rel...](https://github.com/docker/compose/releases)

2. Using PIP
pip install -U docker-compose

Step 2 : Create docker compose file at any location on your system
docker-compose.yml

Step 3 : Check the validity/format correctness of file by command
docker-compose config

Step 4 : Run docker-compose.yml file by command
docker-compose up -d

Steps 5 : Bring down application by command
docker-compose down

TIPS
How to scale services

—scale
docker-compose up -d --scale database=4

Example docker-compose file(Indentation is important)

```
version: "3.9"
services:
  web:
    image: nginx
    ports:
      - "8082:80"
  database:
    image: redis
```

Docker volumes

Volumes are the preferred mechanism for persisting data generated by and used by Docker containers

: docker volume //get information
: docker volume create <volumename>
: docker volume ls
: docker volume inspect
: docker volume rm
: docker volume prune

Instead of deleting containers one by one of docker ps -a , we can use docker container prune.

Use of Volumes

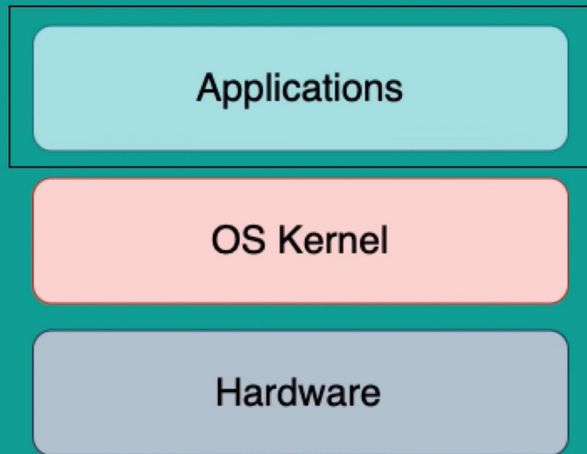
=====

- Decoupling container from storage
- Share volume (storage/data) among different containers
- Attach volume to container
- On deleting container volume does not delete

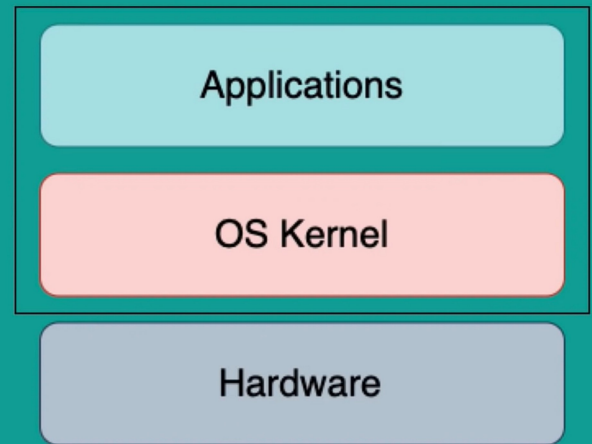
Commands

```
docker run --name MyJenkins1 -v myvol1:/var/jenkins_home -p 8080:8080 -p 50000:50000 jenkins
docker run --name MyJenkins2 -v myvol1:/var/jenkins_home -p 9090:8080 -p 60000:50000 jenkins
docker run --name MyJenkins3 -v /Users/raghav/Desktop/Jenkins_Home:/var/jenkins_home -p 9191:8080 -p 40000:50000 jenkins(This is example of bind mounts)
```

Docker



VM



Difference between container and image
Container is a running env for an image
It has virtual file system
Port binded: talk to application running inside a container
Application image: postgres, mongo etc..

Aws Tutorial(Naveen automation)

Terminologies:

Region->Geo Location(EU,USA)

AZ -->Availability zones(Building/data centre)

VPC-->Private cloud

subnetsSubnetwork under the same VPC

AMI->Amazon machine image(OS+Diff app/software)

Security groups->Firewall rules for AWS instance

IAM -(Identify Access mangement)user/permission roles

- 1)Open aws console
- 2)Launch instance and select community AMIs which has java docker,git installed by default
- 3)If you would like to add a tcp add port number eg 4444(used for selenium grid)
- 4)Ssh into instance `ssh -i /path/my-key-pair.pem my-instance-user-name@my-instance-public-dns-name`

`Sudo yum remove java`

To install java refer open jdk installation