Karate Framework Interview Questions:

1. How we automate Rest Requests in Karate?

In Karate Framework we create Feature files and using Gherkin language we write our test cases. The format of Gherkin language is Given, when, Then. Where in Given we provide endpoints, parameters, headers and When we define the method type and using Then keyword we write assertions.

1. What is \* used in Karate?

In feature file we can replace any Gherkin Keywords (Given, When, Then, And) with \* .

1. What is the keyword to use call endpoint in Karate?

In Karate we use keyword url < endpoint > . Karate understands that the string after url keyword is the endpoint. In Karate we can pass url under Background keyword or we can use it in Config.js file and it will be used for all features. So we can create a variable in config file then just pass the variable name after url keyword.

1. How to pass path or resources in Karate?

In Karate we can pass each resource or services using path keyword.

1. How to define the HTTP/HTTPS method type in Karate?

Using method keyword we can pass the method type such as Get, Post, Put or Delete.

1. How to pass a JSON body in Post or Put Request ?

In Karate we use request method then using three double quotations “”” “”” . and also we can create a JSON file using .json extension and then in feature file we can read the json file using read keyword and pass it in our request.

1. How to define a variable in karate?

In karate we use def keyword to define a variable.

1. How to Print the json response or specific json node in Karate?

In Karate we can use print response or if we want to print specific json node we can store the json node in a variable then using print keyword following with json node name we can print it to console.

1. What are Assertions used in Karate?

* In Karate we can write assertions for status code using status <response code>
* To validate specific json node we can use match == json node
* To validate json schema data types we can use match == #string or #number
* To validate json node we can use match contains and provide a variable to match the response

1. How to call or execute another feature file?

In Karate we can use call keyword to call another feature. First we have to call the file and provide the path of feature file, then we can define a variable to store the response of that feature, for example we need to generate a token and pass token value as headers in our request we can create a separate feature file to head the Token generator scenario then get the token value and once we call the feature then we can store the Token in a variable and pass it as headers value in our request.

1. What is the difference between call and read keyword in karate?

Call will call another feature or file to execute. We usually use this for calling another feature like token generator or another feature that needs to run before the request.

Read will read from files, for example we can use read keyword to read Values from a java class like Data setup classes.

1. How to make a Data Driven scenario in Karate?

In Karate we can create a data driven scenario using Scenario Outline. We can pass the data from Examples and the scenario will run every time with different set of data.

1. How you do data setup in Karate?

In karate we can pass data from Feature file, if we need to generate random data to send as part of our request, we can create a java class and use Java libraries to generate random data. For example, we need to get a random email or name we can use Faker class. We create object of Faker class then use it is methods and then we can read these methods from our feature file and pass these data from feature file instead of hardcoding.

1. In Karate how you can pass value from json response to another json request. For example I need to make a call request and then use one of json response node to pass to another json body as data?

We have scenarios where we need to post a request and then geta the account Id value to pass in another request. So what we do first we run the scenario and then we create a new variable using def keyword to store the accountID like { \* def accountIT = response.accountIT } then in our another request can use this variable accountIT as input data and make a request.

1. How to validate a JSON response ?

In Karate we can use JSONObject class of java which implements Hashmap as key and value. So we can write expected values in values and josn nodes in keys. So on comparison we can call the method from java class and then write assertions to validate each json node.

Karate Framework: Karate Framework is based on Gherkin Language, and we use Feature file to write our API Testing scenarios.

In Karate Framework we have feature file where we write our scenario and scenario outline and also in background keyword, we can put repeated steps. Each scenario will be independent from another scenario and in each feature, we can write multiple scenarios.

In Cucumber Gherkin language we can replace any Gherkin keyword with \* for example:

Given url ‘https://jsonplaceholder.typicode.com’

Is the same as

\* url ‘https://jsonplaceholder.typicode.com’

We can use \* instead of Given, When, And, Then in Karate we will use \* for some step definitions where we need to define a value or we can use it instead of Given .

url: This keyword is used to pass the url with Given keyword

path: this keyword is used to pass the resource path

method: this keyword is used to pass HTTP/HTTPS methods such as GET, Post, PUT, Delete

request: this keyword is used to send json body in POST and PUT methods.

Note: in Karate if we need to pass JSON body we need it send it inside the three double quotation. “”” “””

status: this keyword is used to check response status code such as 200,201, 404, 500

def: this keyword is used to define a variable and assign value from response or request and scope is inside the scenario.

For example, \* def <variable Name> = <variable Value>

print: This keyword is used to print the response or value it is same as System.out.println() in java

response: this keyword is used to store json response and we can get access to response array and response json object

match == : these keywords are used to match the json response node values based on exact match and it is part of assertion.

match contains : these keywords are used to match the json response node values based on contains keywords.

Call : This keyword is used to call another feature file and we can execute a feature file before our scenario as a pre request. For example if we need to generate a token before each request then we can create a feature file and write scenario there then call that feature from another feature file in background keyword.

Read: this will read the response of called feature file and it can be used after call and callonce.

Data Setup in Karate: in Karate we can create a java class and use Faker library to generate random data based on our need, and we can call the Java class with Java.type(path to java class file <package.class>) then we can define a variable and use each method as variable for example: in java class we can create a static String method to generate a random email and then we can call it from feature file by defining a variable then call the method name.

\* def email = datagenerator.getEmail()

If we send a Post method and Put Method we need to Validate Response against Request, and Request against Database.

If we send a get method we need to check Response against Database.

If we send a delete method we need to check the response against Database, to make sure data is removed.

In Karate we can send a JSON payload or body from Feature file with keyword “request” following with three double “”” payload””” and also we can send from JSON file. The file should have .json extension and we can store the JSON values inside the file.

In Karate we have an important file named “karate-config.js”. this is a JavaScript file extension. In this file we can define the url of webservice, so we don’t need to write it everytime. We use call the name of variable after Given url method, Then karate will use this as url to make a call to api. We can send Headers from this file so we don’t need to write headers every time we create feature files and scenarios. Below is code for calling Token generator feature file and passing headers.

var accessToken = karate.callSingle('TokenGenerator.feature', config).token

karate.configure('headers', {Authorization: 'Bearer ' + accessToken})

In karate we can validate JSON response Schema, it means we can validate what is data type of each field in json response body. For example for validating a number or Int we can write:

And match response.data.primaryPerson.email != '#number'

It means we are writing an assertion to email data type should not be a number.

We can use JSONObject class to pass values to JSON nodes. \* A JSON object. Key value pairs are unordered. JSONObject supports \* java.util.Map interface.

In karate we can set values to json nodes using ‘set’ keywords.