Create new JSON file

Add content to file

Declare variables – def

Reading Json from file

In Background: Def expectedOutput = read(‘result.json’)

* And match response == expectedOutput
* If you have multiple arrays then expectedOutput[0]

“#ignore” – will not validate response

Verifying field is present = response.data.last\_name != null

To confirm its not there field

Create variable and then check

* Def job\_code = response.data.job\_code
* And match job\_code == null

Read feature into another feature file \* def feature = read (‘featurefile.feature’)

To perform action on that feature file = \* def result = call feature

Then print ‘result…’, result – will print result

Make post call:

Feature:

Background:

Scenario: Create user details

Given URL

And request ‘JSON Payload’

And headers {Content-Type: ‘application/json’, Accept: ‘application/json’}

When method post

Then status 201

To pass parameters = Add param delay = 3

Data required to be generated token:

And form field grant\_type, client\_id, etc.

And header Authorization = ‘Bearer’ + accessToken (what you saved as variable)

Call read to read another file

To pass value from one file to another: #(email)

* Def authToken = currentFile.accessToken (variable from another file)

JS in karate (Function in karate)

* Def myFun = function(){return ‘hello’}
* \* def return = call myFun
* Then print ‘return---‘, return
* Def myFun1 = “””
* Function(){
* Return ‘myData’}
* “””
* \* def return2 = call myFun1
* Then print return2
* Def data = response.data

Example: function (arg) {return arg.length} or args[0].id

Example: function (arg) {for(i=0; i<arg.length; i++){if(arg[i].id==9){return arg[i]}}

* Def #ofUsers = call fun data

Mistake

POST should be after parameters, headers, URL etc.

Keep space after \* always

Single space before = character

Data Driven

Scenario Outline create user details

And request {“name”:<name>}

Examples:

|Name | salary| age|

|test |50,000|35|

To compare value from one to another

First make post call and save the value into variable: \*def result = response

Then make get call and to assertion:

And match.response.data contains {id: ‘#(result.data.id)’}

For CSV file:

Example:

|read(‘../data/inputData.csv’)|