1. **Rest Assured API Test – GET request:**

*// <groupId>io.rest-assured</groupId>------> <artifactId>rest-assured</artifactId>*

@Test  
**public void** justRestAssuredLib1(){  
RestAssured.*baseURI*=**"http://restapi.demoqa.com/utilities/weather/city"**;  
Response responseGet=*get*(**"/Hyderabad"**);  
responseGet.getBody().prettyPrint();  
String body=responseGet.getBody().asString();

}

@Test  
**public void** justRestAssuredLib2(){  
  
 RestAssured.*baseURI*=**"http://restapi.demoqa.com/utilities/weather/city"**;  
 Response response=*with*().given().get(**"/Hyderabad"**);String body=response.body().asString();  
 System.***out***.println(body);  
}

@Test  
**public void** justRestAssuredLib5() {  
 String uri=**"http://restapi.demoqa.com/utilities/weather"**;  
 String path=**"/city"**;  
 Response response=*with*().given().baseUri(uri).and().basePath(path).get(**"/Hyderabad"**).then().extract().response();  
 System.***out***.println(response.getBody().asString());  
}

@Test  
**public void** justRestAssuredLib3(){  
 RestAssured.*baseURI*=**"http://restapi.demoqa.com/utilities/weather/city"**;  
 Response response=*with*().given().get(**"/Hyderabad"**).then().extract().response();  
 String body=response.getBody().asString();  
 System.***out***.println(body);  
}

@Test  
**public void** justRestAssuredLib4() {  
 RestAssured.*baseURI* = **"http://restapi.demoqa.com/utilities/weather/city"**;  
 Response response = *with*().given().get(**"/Delhi"**).then().extract().response();  
 String body=response.body().asString();  
 **int** sCode = response.statusCode();  
 *assertThat*(**"Wrong Status code, please investigate"**,  
 sCode, *equalTo*(200));  
 JsonPath.*config*=**new** JsonPathConfig();  
 JsonPath jp=**new** JsonPath(body);  
 String city=jp.get(**"City"**);  
 *assertThat*(**"Wrong city returned, please investigate"**,  
 city,*equalTo*(**"Delhi"**));  
}

@Test  
**public void** justRestAssuredLib6() {  
  
 String uri=**"http://restapi.demoqa.com/utilities/weather"**;  
 String path=**"/city"**;  
 Response response=*with*().given().baseUri(uri).and().basePath(path).get(**"/Hyderabad"**).then().extract().response();  
 String body=response.getBody().asString();  
  
 **int** sCode=response.getStatusCode();  
 System.***out***.println(**"Status Code : "**+sCode);  
 String sLine=response.getStatusLine();  
 System.***out***.println(**"Status Line : "**+sLine);  
 String ct=response.getContentType();  
 System.***out***.println(**"Content Type : "**+ct);  
 Headers header= response.getHeaders();  
 System.***out***.println(**"All Headers content : "**+header);  
  
 JsonPath jp=**new** JsonPath(body);  
 String city=jp.get(**"City"**);  
 System.***out***.println(**"City Node USING JsonPath : "** +city);  
 Map<String, Object> allNodes=jp.getMap(**"$"**);  
 System.***out***.println(**"Number properties in json object : "**+ city +**" : "**+allNodes.size());  
 Set<String> allKeys=allNodes.keySet();  
 **for**(String key: allKeys){  
 System.***out***.println(**"Keys : "**+ key + **" : Values : "**+allNodes.get(key));  
 }  
  
 System.***out***.println(**" Just to compare response >>>>>>>> : "**);  
 response.getBody().prettyPrint();  
}

@@@@@@@@@@@@@@@@@@@@ **Using parsing the JSON body returned as Json Array** @@@@@@@@@@@@

@Test  
**public void** allCountries() **throws** ParseException {  
 String uri = **"https://restcountries-v1.p.rapidapi.com"**;  
 String path = **"/all"**;  
  
 Map<String, String> headers = **new** HashMap<>();  
 headers.put(**"X-RapidAPI-Host"**, **"restcountries-v1.p.rapidapi.com"**);  
 headers.put(**"X-RapidAPI-Key"**, **"8bcb167f50msh8c2f0f0c3daa81ep162e14jsn12daa5cf7d03"**);  
  
 Response response = *with*().given().baseUri(uri)  
 .and().basePath(path)  
 .and().headers(headers).get()  
 .then().extract().response();  
  
 String body = response.getBody().asString();  
  
 *// First way - - - to get the number of countries* JsonPath.*config* = **new** JsonPathConfig();  
 JsonPath jp = **new** JsonPath(body);  
 List<String> allCountries = jp.get(**"name"**);  
 System.***out***.println(**"Number of countries : "** + allCountries.size());  
  
 *// Second way - - - to get the number of countries* List<String> list = jp.get(**"$"**);  
 System.***out***.println(**"USING JSONPATH : "** + list.size());  
  
 *// Third way - - - using googlecode json-simple* JSONParser jsonParser = **new** JSONParser();  
 JSONArray jsonArray = **null**;  
 jsonArray = (JSONArray) jsonParser.parse(body);  
 **if** (jsonArray != **null**) {  
 System.***out***.println(**"Json array using Simple json Library : "** + jsonArray.size());  
 }  
 JSONObject jsonObject = **null**;  
 **if** (jsonArray != **null**) {  
 jsonObject = (JSONObject) jsonParser.parse(jsonArray.get(0).toString());  
 }  
  
 System.***out***.println(**"Content of the first object in response body returned in JSON ARRAY : "** + jsonArray.get(0));  
 System.***out***.println(**"json object for index 0 >>>>>>>>>>. "** + jsonObject.get(**"name"**));  
 System.***out***.println(**"json object for index 0 >>>>>>>>>>. "** + jsonObject.get(**"topLevelDomain"**));  
 System.***out***.println(**"json object for index 0 >>>>>>>>>>. "** + jsonObject.get(**"alpha2Code"**));  
 System.***out***.println(**"json object for index 0 >>>>>>>>>>. "** + jsonObject.get(**"alpha3Code"**));  
 System.***out***.println(**"json object for index 0 >>>>>>>>>>. "** + jsonObject.get(**"callingCodes"**));  
 System.***out***.println(**"json object for index 0 >>>>>>>>>>. "** + jsonObject.get(**"capital"**));  
 System.***out***.println(**"json object for index 0 >>>>>>>>>>. "** + jsonObject.get(**"altSpellings"**));  
 System.***out***.println(**"json object for index 0 >>>>>>>>>>. "** + jsonObject.get(**"region"**));  
 System.***out***.println(**"json object for index 0 >>>>>>>>>>. "** + jsonObject.get(**"subregion"**));  
 System.***out***.println(**"json object for index 0 >>>>>>>>>>. "** + jsonObject.get(**"borders"**));  
}

**// Using the Properties file**

@Test  
**public void** allCountries() **throws** IOException {

*// loading properties and initialize BufferedReader using FileReader* Properties properties = **new** Properties();  
 BufferedReader reader = **new** BufferedReader(**new** FileReader(**"./test.properties"**));  
 properties.load(reader);  
 reader.close();  
  
 Map<String, String> headers = **new** HashMap<>();  
 headers.put(**"X-RapidAPI-Host"**, properties.getProperty(**"host"**));  
 headers.put(**"X-RapidAPI-Key"**, properties.getProperty(**"key"**));  
 String uri = properties.getProperty(**"uri"**);  
 String path = properties.getProperty(**"path"**);  
  
 Response response = *with*().when().baseUri(uri).basePath(path).and().headers(headers).get().then().extract().response();  
  
 *assertThat*(**"Wrong status code returned, Please investigate"**,  
 response.getStatusCode(), *equalTo*(200));  
   
 *assertThat*(**"Response body is Empty, Please investigate"**,  
 response.getBody(), *notNullValue*());  
  
}

**// Using the Properties file and Serenity Session Variables**

@Test  
**public void** usingSerenitySessionVariable() **throws** IOException {  
  
 Properties properties = **new** Properties();  
 BufferedReader reader = **new** BufferedReader(**new** FileReader(**"./test.properties"**));  
 properties.load(reader);  
 reader.close();  
  
 String uri = properties.getProperty(**"uri"**);  
 String path = properties.getProperty(**"path"**);  
  
 Serenity.*setSessionVariable*(**"myUrl"**).to(uri);  
 Serenity.*setSessionVariable*(**"myPath"**).to(path);  
  
 Map<String, String> headers = **new** HashMap<>();  
 headers.put(**"X-RapidAPI-Host"**, properties.getProperty(**"host"**));  
 headers.put(**"X-RapidAPI-Key"**, properties.getProperty(**"key"**));  
  
 String serenityUri=Serenity.*sessionVariableCalled*(**"myUrl"**);  
 String serenityPath=Serenity.*sessionVariableCalled*(**"myPath"**);  
  
 Response response=*with*().when().baseUri(serenityUri).basePath(serenityPath).headers(headers).get().then().extract().response();  
  
 response.getBody().prettyPrint();  
}

**// Using the Properties file and Validatable Response**

@Test  
**public void** testWithVR() **throws** IOException {  
  
 Properties properties = **new** Properties();  
 BufferedReader bufferedReader = **new** BufferedReader(**new** FileReader(**"./test.properties"**));  
 properties.load(bufferedReader);  
 bufferedReader.close();  
  
 **final** String uri = properties.getProperty(**"uri"**);  
 **final** String path = properties.getProperty(**"path"**);  
  
 **final** String host = properties.getProperty(**"host"**);  
 **final** String key = properties.getProperty(**"key"**);  
  
 Map<String, String> headers = **new** HashMap<>();  
 headers.put(**"X-RapidAPI-Host"**, host);  
 headers.put(**"X-RapidAPI-Key"**, key);  
  
 ValidatableResponse vr = *with*().when().baseUri(uri).basePath(path).headers(headers).get()  
 .then();  
  
 *assertThat*(**"Wrong status code returned, Please investigate"**,  
 vr.extract().response().getStatusCode(), *equalTo*(200));  
  
 *assertThat*(**"Response body is empty, Please investigate"**,  
 vr.extract().response().getBody(), *notNullValue*());

*// Or --- To assert Count of countries* String str=vr.extract().response().getBody().asString();  
  
 JsonPath jsonPath=**new** JsonPath(str);  
 List<String> allCountries=jsonPath.get(**"name"**);  
  
 *assertThat*(**"Number of countries returned is incorrect, Please investigate"**,  
 allCountries.size(), *equalTo*(250));

}

**// Using the Properties file and Extractable Response**

@Test  
**public void** testWithEr() **throws** IOException {  
  
 Properties properties = **new** Properties();  
 BufferedReader bufferedReader = **new** BufferedReader(**new** FileReader(**"./test.properties"**));  
 properties.load(bufferedReader);  
 bufferedReader.close();  
  
 **final** String uri = properties.getProperty(**"uri"**);  
 **final** String path = properties.getProperty(**"path"**);  
  
 **final** String host = properties.getProperty(**"host"**);  
 **final** String key = properties.getProperty(**"key"**);  
  
 Map<String, String> headers = **new** HashMap<>();  
 headers.put(**"X-RapidAPI-Host"**, host);  
 headers.put(**"X-RapidAPI-Key"**, key);  
  
 ExtractableResponse er = *with*().given().baseUri(uri).basePath(path).headers(headers).when().get()  
 .then().extract();  
  
  
 *assertThat*(**"Wrong status code, Please investigate"**,  
 er.statusCode(), *equalTo*(200));  
  
  
 Headers allHeaders = er.response().getHeaders();  
  
 *assertThat*(**"Wrong headers returned, Please investigate"**,  
 allHeaders, *notNullValue*());  
  
 String str = er.response().getBody().asString();  
 JsonPath jsonPath = **new** JsonPath(str);  
  
 List<String> allCountries = jsonPath.get(**"name"**);  
  
 Assert.*assertThat*(**"Number of countries returned is incorrect, Please investigate"**,  
 allCountries.size(), *equalTo*(250));  
  
}