***Task Given:***

*There are three resources wherein one is dependent on the other. One is the manufacturer API that results in the manufacturer codes which is an input to the main type API. Further the main type API returns some main type codes and both the manufacturer code and the main type code helps in producing the output for the built in API.*

***Automated tests written and their explanation:***

1. ***Scenario #1*** *is written that tests whether the* ***Manufacturer API*** *is returning the different manufacturer codes with the key and locale passed as a query parameter. This tests that this API is working as expected and we also test that the API is returning a status code of 200. This is tested as a happy path for the manufacturer API.*

***Observation:*** *Actual Result matches with the expected result****.***

1. ***Scenario #2:*** *In this scenario**we remove the authentication key from the manufacturer API(****wa\_key*** *in this case) and test that after removing this we get a response of 401(Unauthorized response).*

***Observation:*** *Actual Result matches with the expected result****.***

1. ***Scenario #3(Bug Identified):*** *In this scenario we remove the locale from the Manufacturer API. Since this API accepts only two parameters(one of which is the authentication key and the other is the locale) it is evident that without a locale being passed as a query parameter to this API it shouldn’t work. According to my observation if locale is a mandatory parameter it should either return a* ***Bad Request****(error 400) or an* ***Internal Server Error****(error 500****).*** *However this API also returns the desired response even without appending the locale as a query parameter and gives a status of 200.*

***Observation:*** *Actual Result doesn’t match with the expected result****.***

1. ***Scenario #4:*** *In this scenario we hit**the Main type API without any manufacturer code and test the Main type API response. In this case the response should be either a* ***Bad request(****error 400****)*** *or an* ***Internal server error****(error 500). The API returns a bad request with error 400 and an error message saying that “Required String parameter ‘manufacturer’ is not present.” which is expected.*

***Observation:*** *Actual Result matches with the expected result****.***

1. ***Scenario #5(Bug Identified)****: For this scenario we test the Main type API without any locale and observe if the API is returning a* ***Bad request****(****error 400****) or an* ***Internal server error(error 500)****.However just like the Manufacturer API this API also returns the desired response even without appending the locale as a query parameter and returns a status of 200.*

***Observation:*** *Actual Result doesn’t match with the expected result.*

1. ***Scenario #6:*** *In this scenario we try to identify the response of the Main type API without the authentication key as a query parameter. The response in this case should return an unauthorized response. This API response exactly gives the response as expected.*

***Observation:*** *Actual Result matches with the expected result****.***

1. ***Scenario #7(Bug Identified)****:**In**this scenario we pass a two digit manufacturer code to the main type API. If the manufacturer code is not valid then we should be getting either an error 400 or an error 500 for the main type API since the main type API is dependent on the manufacturer API response. However this API is returning an empty wkda object which according to me is not desired.*

***Observation:*** *Actual Result doesn’t match with the expected result.*

1. ***Scenario #8(Bug Identified):*** *This scenario is almost identical to* ***Scenario #7.*** *However it was tested for atleast one time to check if an invalid three digit manufacturer code being passed as a query parameter to the Main type API it returns an error 400 or error 500 for the main type response. In this case as well the API returns an empty wkda object which is not desired.*

***Observation:*** *Actual Result doesn’t match with the expected result.*

1. ***Scenario #9****: In this scenario we tried hitting the built in API without an authentication key and with all other parameters intact. This should result in an unauthorized response(error 401). We exactly get the same result for it.*

***Observation:*** *Actual Result matches with the expected result****.***

1. ***Scenario #10(Bug Identified):*** *For this scenario we hit the built in API without locale. In this case as other API responses it should return a**Bad request****(error 400)*** *or an**Internal server error****(error 500).*** *However this API is also returning a valid wkda response with the desired parameters.*

***Observation:*** *Actual Result doesn’t match with the expected result.*

1. ***Scenario #11:*** *In this scenario we hit the built in API without manufacturer code. The response here should be either a* ***Bad request(****error 400****)*** *or an* ***Internal server error****(error 500). In this case the API returns a bad request with error 400 and an error message saying that “Required String parameter ‘manufacturer’ is not present.” This one suffices the fact since if the manufacturer code is not present or invalid it should give a bad request.*

***Observation:*** *Actual Result matches with the expected result****.***

1. ***Scenario #12****: For this scenario we hit the built in API without main type code. Here the response should be either a* ***Bad request(****error 400****)*** *or an* ***Internal server error****(error 500). In this case this API returns a bad request with error 400 and an error message saying that “Required String parameter ‘main type is not present.” This one suffices the fact since if the manufacturer code is not present or invalid it should give this response.*

***Observation:*** *Actual Result matches with the expected result****.***

1. ***Scenario #13(Bug Identified):*** *In this scenario we test the built in API with an invalid two digit main type code. If the main type dropdown is an invalid selection from the API level it should either give a Bad request****(error 400)*** *or an Internal server error(****error 500).*** *However the API in this case returns an empty wkda.*

***Observation:*** *Actual Result doesn’t match with the expected result.*

1. ***Scenario #14:*** *For this scenario**we test the built in API without the manufacturer code. In this situation the expected response should be a bad request****(error 400).*** *The response matches with the expectation.*

***Observation:*** *Actual Result matches with the expected result****.***

1. ***Scenario #15:*** *In this scenario we test the manufacturer API with an invalid key. This response is returning forbidden(403) which is expected.*

***Observation:*** *Actual Result matches with the expected result****.***

1. ***Scenario #16:*** *For this scenario the main type API is tested with an invalid key. The response is 403 as expected.*

***Observation:*** *Actual Result matches with the expected result****.***

1. ***Scenario #17:*** *This scenario tests the built in API with an invalid key. We get an expected response of 403.*

***Observation:*** *Actual Result matches with the expected result****.***

*18.* ***Scenario #18:*** *This scenario end to end tests all the responses. All the possible combination of manufacturer code and main type codes are passed to the built in API to test that the built in API doesn’t break for any combination. In addition to that all manufacturer code has been passed to the main type API in order to test that the main type API is not breaking for any manufacturer code.*

***Observation:*** *Actual Result matches with the expected result****.***

***Advantages Of Automating the APIs:***

*If we test on an average each API tests when running in automation is taking around 30milliseconds to 40milliseconds except the end to end test. If this would have been done manually then it would have taken a greater time. Specifically for the end to end test we are testing all possible combinations of manufacturer code and main type code which is passing more than 500 responses in just 600 seconds. If we were to test this end to end test manually then it would have been a day’s work. There would have been a scope of data being missed out which might lead to a bug.*

***Buggy Scenarios:***

*Scenario# (3,5,7,8,10,13) are identified as buggy scenarios. The explanation as to why they are identified as buggy are explained explicitly.*

***An extra observation:***

*When the authentication key is removed we get a 401 response and when the authentication key is tampered we get a 403 response. Is this really necessary?*