**ISS API Testing Report**

**Objective**

To create a automated test suite for testing the ISS API to get the current, past, or future position of the ISS and also get TLE data on the ISS

APIs in Scope:

* satellites/[id]/positions
* satellites/[id]/tles

**Test Automation Setup**

Created a simple BDD cucumber based framework on Rest Assured(JAVA) for setting up regression testing. Each of the features are logically created based on the core functionality like GET/POST/PUT/DELETE of satellites.

Each feature has examples to cover, happy, and negative flows

**Framework Details:**

1. Runner Class - To run the tests
2. Feature files - To mention test case scenarios with examples
3. POJO Helpers - To set and get json objects
4. Report helper class - To generate extent reports
5. Step builder and helper classes - For step definition glues and reusable methods

**How to run the cases?**

Prerequisite :

* Install JAVA JDK- 11.0.6"
* Install Intellij/Eclipse- a suitable IDE
* Install GIT bash/GIT GUI to pull my repo
* If your IDE does not have MAVEN already available you can get it from marketplace
* You also need to setup cucumber on your IDE-<https://www.jetbrains.com/help/idea/enabling-cucumber-support-in-project.html>

Steps to Run the test

1. Create a folder on your local where you want to download the project
2. Clone the project from GIT with command git clone <https://github.com/madhulikamitra/wheretheissAPITesting.git>
3. Once cloned successfully open your IDE and import the project as a Maven project. Your path should be until where the pom.xml resides
4. Build with Maven-Install, once the project is imported. Wait for all dependencies to get downloaded
5. Ensure you see no errors in your project, maybe i missed some prerequisite step, so for any issues let me know :)
6. Right click and Run the test runner file under -APIAutomation/src/test/java/com/automation/hotel/runner/TestRunner.java
7. Currently all feature files are marked with “Regression” Tag, please change/comment with hash (#) symbol next to the tag if you wish to run individual features
8. After the run, the reports are present at APIAutomation/test-output/SparkReport
9. I have placed my generated report in the same folder named as “Final Report\_ISS.html”

**Techstack**

Java, Maven, Rest Assured, BDD,Extent Reports

**GIT Repo**:<https://github.com/madhulikamitra/wheretheissAPITesting>

**Assumptions**

Please pardon my silly assumptions, since satellites are a very far away topic for me :), it’s all based on some googling I did.

1. The minimum velocity should be 27000 kmph
2. The latitude should be in range of -90 and 90
3. The longitude should be in range of -180 and 180
4. The visibility could be any of 3 values (visible, eclipsed, daylight)

**Test Cases**

| S.No | Test Case | Validation | Status | Comments |
| --- | --- | --- | --- | --- |
| 1 | To perform health check of the application | To validate that via health ping | Pass |  |
| 2 | To validate we can get all details of the satellite by ID | Get all details of satellite by id and validate the model, the accuracy of details in the body params | Pass |  |
| 3 | To validate we can get the position of the satellite timestamps and units | Get the positions of the satellites by specifying current/past or future timestamps and validate all returned satellites and individual details.  Tried to get details specifying kilometers and miles | Pass |  |
| 4 | To validate we can get position of satellite with just timestamps | Get the positions of the satellites by specifying current/past or future timestamps and validate all returned satellites and individual details. | Pass |  |
| 5 | To validate we do not get details with no timestamps | When timestamps are not specified the search for positions should fail gracefully and the unit should be by default set as kilometers | Pass |  |
| 6 | To validate with up until 10 timestamps | Validated with 10 timestamps to get positions and also validated with more and it returned the data | Fail | Specify a comma delimited list of timestamps for orbital positions, limit 10 per request |
| 7 | To validate the TLE data | Validated with a valid satellite id to return all TLE details and TLE details validated individually again | Pass |  |
| 8 | To validate with incorrect satellite ID | When a special character is passed post the correct satellite id, system does not validate for wrong satellite id-https://api.wheretheiss.at/v1/satellites/25544$$$6 | Fail | Only valid is is 25544 |
| 9 | To validate methods like put, post and delete | All these methods need authorisation and are not allowed | Pass |  |
| 10 | To test if we try to use an incorrect path | When we use an incorrect path there should be an error. Error code and message validated | Pass |  |

**Defects**

1. When a special character is passed post the correct satellite id, system does not validate for wrong satellite id-<https://api.wheretheiss.at/v1/satellites/25544$$$6>(Medium)
2. The limit of timestamps should be 10 in the Get Positions requests but it accepts more.(Medium)
3. If the units is specified anything other than “kilometers” and “miles”, it accepts but then considers as kilometers only (Minor)

**PostMan Collection:**

All the requests are created and tested via postman. Collection is added in the GIT repo as well