**API exercise**

Using <https://api.spacex.land/graphql/>

* Write a test that queries the API for all launches and returns the number of launches

\* Assert status code is 200

\* Assert mission name object is not empty

\* Assert that the number of launches is greater than 0

\* Assert that that the number of ships is greater than 0

\* Assert the first stage & second stage are not null

* Write a test to update the limit parameter and returns the number of launches Ex: limit=2

\* Assert status code is 200

\* Assert mission name object is not empty

\* Assert that the number of launches is greater than 0

\* Assert that that the number of ships is greater than 0

\* Assert the first stage & second stage are not null

* Write a test to update the offset parameter and returns the number of launches Ex: offset=2

\* Assert status code is 200

\* Assert mission name object is not empty

\* Assert that the number of launches is greater than 0

\* Assert that that the number of ships is greater than 0

\* Assert the first stage & second stage are not null

Above are the requirements for the API exercise, use any of the automated testing tools to complete the exercise.

[Rest assured](https://rest-assured.io/)

[Karate](https://github.com/karatelabs/karate)

[Postman](https://www.postman.com/)

**Implement a test to verify the following user journey**

1. Login to <https://www.saucedemo.com/> using the "standard\_user" account
2. Sort the products by Price (high to low)
3. Add the cheapest & the 2nd costliest products to your basket
4. Open the basket
5. Checkout
6. Enter details and Finish the purchase

**Instructions and deliverables**

* Design and implement an extensible test framework to run and verify the above user journey. This should incorporate the best practices of the tooling you chose, and will be reviewed against extensibility and clean code.
* Writing the tests Use Java([Selenium](https://www.selenium.dev/)) or Javascript([Cypress](https://www.cypress.io/) or [Playwright](https://playwright.dev/))
* We don't expect every single aspect of the journey to be thoroughly tested, but would expect to see some set of scenarios tested
* A README.md file explaining the framework briefly and how to run the tests, what you need installed
* Pushed to GitHub for review