* When implementing the scenario, I used OOP such as abstract methods. The major challenges were refactoring the methods to create the stubs to test the API and trying to create more test to cover the coverage. The approach I used is doing test cases before coding and testing the API.
* HDD Task

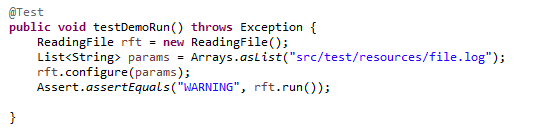
Description of the task and logic: giving the path of a log file, going through the file and save the last line, by using the indexof to find (“ : “) and get the word that should be ERROR, INFO or WARNING.

Inputs required and validation rules: input required is a file path, validation is that the file path should exist, passing a parameter in the config method should not be null and empty.

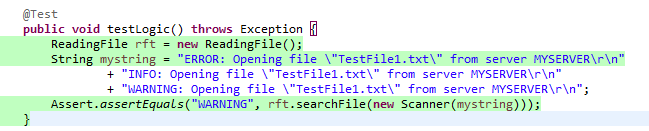
All possible outputs: INFO, WARNING or ERROR and invalid file.

Sample of inputs and relative output:

Using a HDD



Not using a HDD



* API

Description of the task and logic: giving a twitter username it will give me back a number of tweets of the last 7 days. Every day it always gives me a different number. Logic is by saving the number of tweets in an array and getting the size of the array. This is the library that I used in order to use the methods of the API



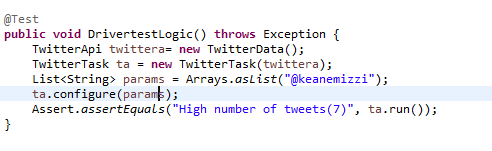
Inputs required and validation rules: a string username, validation a parameter in the config method should not be null and empty.

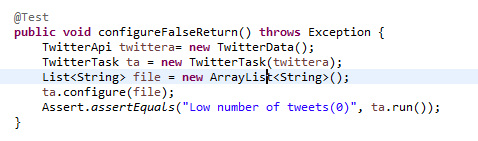
All possible outputs: If the tweets are higher or equal to 3 the output will be High number of tweets(numberOfTweet) and if less than 3 Low number of tweets(numberOfTweets).

Sample of inputs and relative output:



Using drivers:





Using a stub

