Automation Exercise from SLACK:

Given the scope of the problem and time constraints I had, I have limited myself to doing the minimum the best way possible.
Obviously there are plenty of other improvements and better techniques that can be employed when worked on a broader time frame and execution window.

TASK 1:

FRAMEWORKS USED:

SELION (https://github.com/paypal/SeLion) (TESTNG+JUNIT)

Prerequisites

Make sure the following software is installed on your machine before executing:

1. Java JDK 1.7
2. Maven 3.2.x

Notes to execute tests:

1. EXECUTION INSTRUCTIONS:
   On unzipping the jar, from the project directory, run the below command to execute all the tests written covering the requirement provided

```
mvn clean test -DsuiteXmlFile=src/test/resources/fileOperationsSuite.xml
```

2. SUITE XML's: There are other SuiteXML Files also located in src/test/resources/ folder doing the below and can be executed as per the instruction in Step1 :
   a. uploadFilesTestSuite.xml - to run only File Upload Operation test cases.
   b. listFilesTestSuite.xml - to run only File List Operation test cases.
   c. deleteFilesTestSuite.xml - to run only File Delete Operation test cases.
   d. smokeTestsSuite.xml - to run only test cases GROUPED as "SmokeTests".

3. REPORTING: On Test Execution, navigate to the project directory and look for the index.html for a complete report on test execution

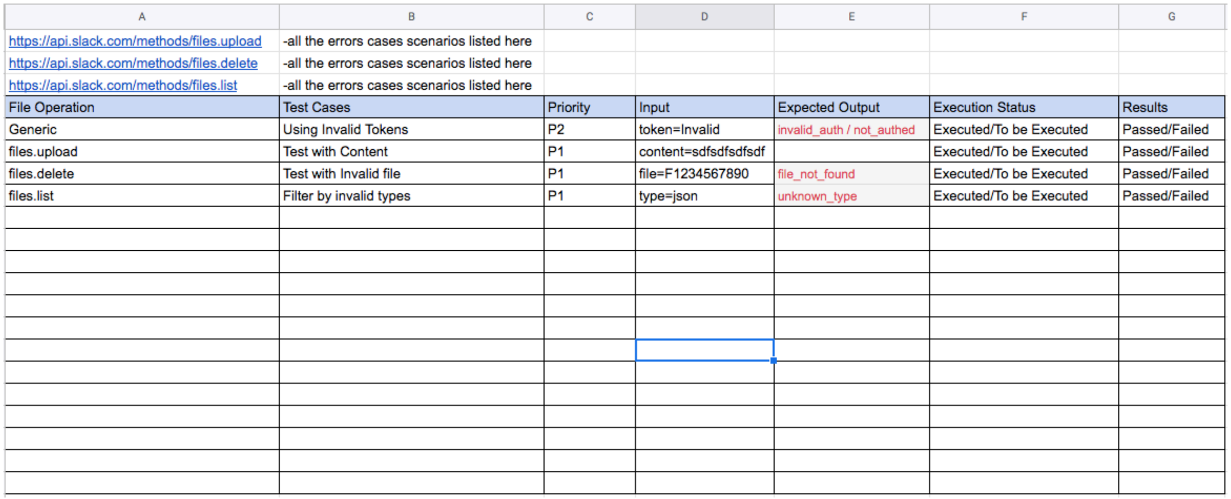
```
target/surefire-reports/RuntimeReporter/index.html
```

TASK 2:

I would envision a TO-DO List like the one below. This will have an exhaustive list of all the test cases that are referred to in the API documentation of each of the File Operation.
UPLOAD: https://api.slack.com/methods/files.upload/
DELETE: https://api.slack.com/methods/files.delete
LIST: https://api.slack.com/methods/files.list

Other Techniques or Ideas:

1. Using Data providers or similar JSON Templates (https://testng.org/doc/documentation-main.html#parameters-dataproviders) to generate all the permutations/combinations required to passes in as Request to the API operations.
2. Use Jenkins or Other Continuous Integration Pipelines to built running the above and below test depending on the need (Regression, Integration, Build Sign Off, Smoke etc.,)
3. Logging/Monitoring Systems can also be designed/built around the test system.



A	B	C	D	E	F	G
https://api.slack.com/methods/files.upload	-all the errors cases scenarios listed here					
https://api.slack.com/methods/files.delete	-all the errors cases scenarios listed here					
https://api.slack.com/methods/files.list	-all the errors cases scenarios listed here					
File Operation	Test Cases	Priority	Input	Expected Output	Execution Status	Results
Generic	Using Invalid Tokens	P2	token=Invalid	invalid_auth / not_authed	Executed/To be Executed	Passed/Failed
files.upload	Test with Content	P1	content=sdfsdfsdfsdf		Executed/To be Executed	Passed/Failed
files.delete	Test with Invalid file	P1	file=F1234567890	file_not_found	Executed/To be Executed	Passed/Failed
files.list	Filter by invalid types	P1	type=json	unknown_type	Executed/To be Executed	Passed/Failed