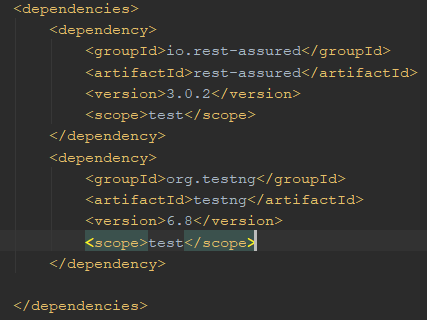
To test the given assignment, I have used RestAssured, TestNG, JAVA with IntelliJ IDEA as editor.

To have RESTASSURED and TESTNG I have added below entries to POM.xml file of the project



By adding the dependencies, we get the library access to the RestAssured and TestNG.

To use its features, we have to import it like shown below:



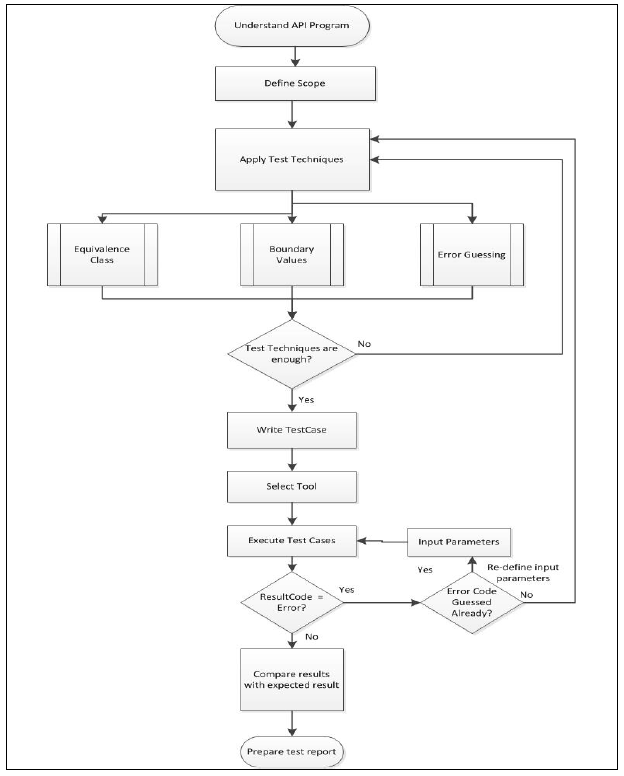
**1. How would you scale out your answer in the first task to cover an entire system? Would there be tweaks?**

API stands for Application Programming Interface, which specifies how one component should interact with the other. It consists of a set of routines, protocols and tools for building the software applications.

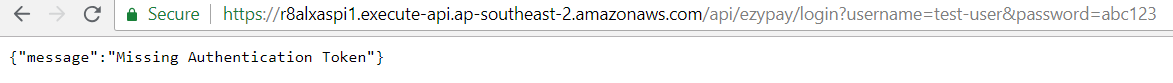
General steps involved while performing API testing are mentioned below:

* Details of API information is found in Specification Document which lists the signatures of each API function (the input parameters, the function or method name, and the return type)
* Identify the software to be used for API testing
* Add the web service call to be tested
* API call will have request and response parameters
* Prepare the inputs for the request
* Invoke the method with all provided inputs
* Analyse the output response

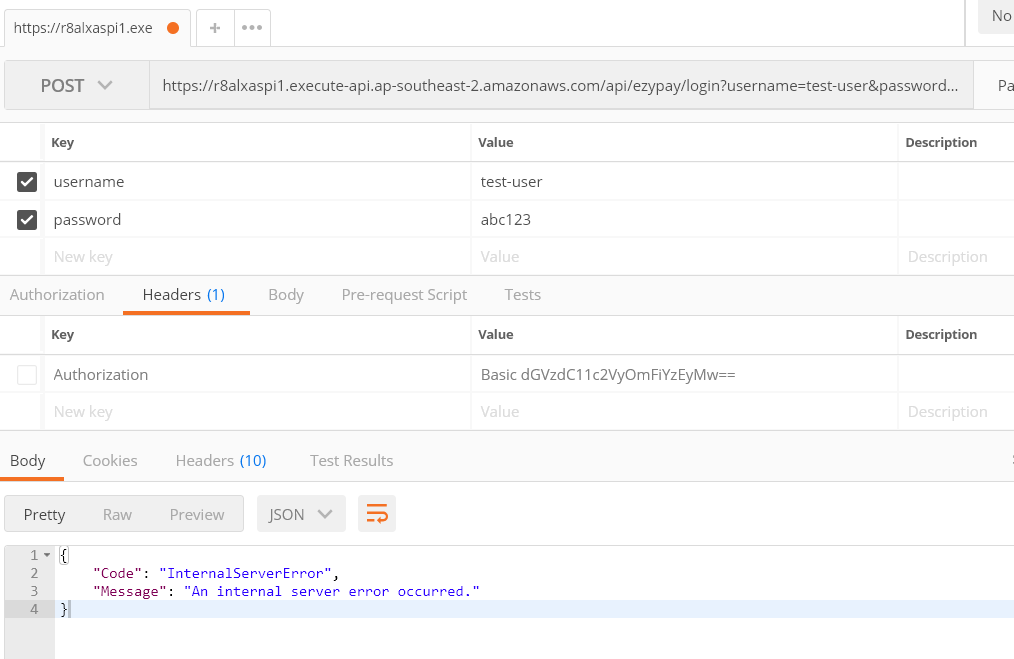
Below flow chart represents the steps to be considered for API testing:



While working on the problem statement, I found that there is a token required for the POST request to be sent.



If I keep the basic authentication with authorization token, I get below:



**2. How much do you test? What scenarios do you test?**

**Scope for Testing**

To know how the API works, it is important to understand the functionality of an API and clearly define the scope of the program.

* Apply testing techniques such as equivalence classes, boundary value analysis and error guessing and write test cases for the API.
* Input Parameters for the API need to be planned and defined appropriately.
* Execute the test cases and compare expected and actual results.
* Use the appropriate tool to test API which gives more effective results

When we understand and realize the importance of Web Services Automation, the next challenge is to select the right Test approach and Test Type before jumping into automation design and implementation.

**Test Objective**

It is important to answer the basic question “Why do we want to do Web Services Testing?” and that will help us to decide the right kind of Test we should choose to automate them. Let’s begin with few possible answers

* To validate the functional behaviour of your application / APIs
* To validate the performance aspect of your application / APIs
* To first validate the functionality and then also test the performance aspect of it.

Once the objective of the test is clear, we need to look at the technologies that can make a difference to the way we test the Web Services.

**3. What factors would you consider to ensure you get the widest end-to-end coverage, AND yet keep it easy to maintain?**

**HTTP**

* Authentication - identify and authenticate the user who accesses the API.
* Idempotent methods - GET, POST, HEAD, PUT, DELETE, OPTIONS
* status codes
* 4xx vs 5xx
* 200 OK - is a response code that indicates that the request was processed successfully.
* 202 Accepted - is a response code which indicates that the request is valid but hasn’t been completed.
* HTTP Caching
* Canonical URLs

**API Design**

* Content Negotiation
* API Versioning
* Bulk Operations
* Error Logging

**Content**

* Content Types
* Date/Time

**Security**

* SSL-HTTPS

**Documentation**