Web API Technical Assessment

The Movie DB project (www.themoviedb.org) is a free community built database of movie and TV meta data. TMDB also provides a free API web service that provides movie buffs with a wealth of information and details regarding their favorite movies and allows the information to be integrated with both commercial and open source media related applications. One of my personal favorites web applications and large consumer of TDMB data is the Plex Media Server https://www.plex.tv/

The APIs are all REST and return JSON data. Visit https://developers.themoviedb.org/3/getting-started for detailed documentation regarding the API. Your assignment should you choose to accept it is to pretend as though you were writing automated test cases on a new release of the TMDB API.

The requirements for the assessment are as follows:

1. Use Java (preferred) or Python and pick whatever test framework you're most familiar with. (Code generators or tools like Postman don't count)
2. Check your code into a free account on GitHub or Bitbucket and send us the link to the repo.
3. You'll need to create a free developer API key that is required to use the API.
4. Don't worry about sending us the key or checking it into a repo but instead parameterize the key so I can plug in my own key.
5. Identify at least a dozen tests you would write. Feel free to pick any endpoint.
6. Stub out all those tests with functions and using comments in the code write TODOs for the name of the tests and 1-2 sentences describing what the test will do and why
7. Implement 3-4 of the tests with actual runnable code
8. Tests should be runnable right out of the box. (i.e. I should be able clone your repo and run the tests)
9. Don't forget to include instructions on how to execute the tests!
10. Hard-coding of configuration or environment will get marks against you
11. Tests should use good OOP principles and have some abstraction in the testing layout so that they could be a base/template for implementing many automated API tests against the movie db API web service.
12. Tests should create libraries/helper classes and base classes to increase maintainability and readability
13. Bonus points are given for non-happy path test scenarios and more creative approaches