Part 3: Reflection on the automation exercise

The rest assured API automation tests should cover at least following testing methods apart from usual SDLC process.

**1. Validation Testing**

The tests that fall under validation testing are among the most important to the API’s development. They involve holistic issues about the [different types of API](https://stoplight.io/api-types/), such as:

1. Whether the correct product was made for the application
2. If the API accesses its data correctly, in the manner defined by the development team
3. If the API does what it’s supposed to do with accuracy and efficiency

Tested each individual pet endpoints with different permutations and combination to understand the functionality and behaviour. This is also performed to make sure the system is behaving as per the requirements defined

*Pro:* Validation testing is where one can expect to get a bird’s eye view of the API product. This is especially helpful in the later stages of development, where everyone’s goals should be aligning.

*Con:* For the gravity of the issues it involves, validation testing is quite difficult. Sometimes, when we find a problem during this type of test, it requires us to go back to the drawing board.

**2. Functional Testing**

 Functional testing is concerned with testing specific functions within the codebase. In other words, these tests involve the nitty-gritty of the API. Functional testing can cater to specific scenarios in regular test cases, as well as errata and edge cases (those involving extremes). This involves the end to end positive as well as negative tests.

*Pro:* Functional testing will yield a lot of purposeful insight on how the API works within its given parameters.

*Con:* This is a type of test that you can expect to do a lot of. Test purposefully so that you don’t get overwhelmed.

**3. Security Testing**

 This category includes tests that are done for security audit purposes, comprising the following:

1. Security tests that validate encryption methodologies and the API’s access control solution
2. Penetration tests that evaluate the threat vector from an outsider to the API, assuming malicious intent
3. Fuzz tests that input random noise or “fuzz” and instigate a forced crash from the API

*Pro:* All three of these are done to see what happens in the worst-case scenarios for the API. Thus, they can be the basis for appropriate security solutions.

*Con:* API development teams tend to get complacent about security testing. But we cannot assume that the finished API product will be secure 100% of the time. Testers must be thorough in dealing with potential threats to the API’s security.

**4. Load Testing**

Load testing can only happen when either specific unit tests or the whole API’s codebase has been completed. These tests are meant to see if the solutions proposed by API developers will work. To adapt theory into practice, each test is given a prescribed load, or a measurable baseline.

*Pro:* With this kind of test, we can measure your API’s performance against the regular traffic that we expect for it. We can also test against maximum traffic or an amount of traffic that is already considered an overload.

*Con:* API load testing does not simulate real users interacting with elements of your webpage. It doesn’t give you an idea of how user-friendly your application is. It doesn’t measure front-end performance or how quickly pages render in different browsers. API load testing doesn’t run client-side scripts

**Proposed testing types for the automated test**

|  |  |  |
| --- | --- | --- |
| API Call | Action | Testing Types |
| GET /pet/{id} | Get pet details By Id | Validation/Functional/Performance |
| GET /petByStatus?status={sold,available, pending} | Get pet details By status | Validation/Functional/Performance |
| GET /petByTags?tags={tagname} | Get pet details By tags | Validation/Functional/Performance |
| POST /pet | Create a new pet | Validation/Functional/Performance |
| DELETE /pet/{id} | Delete a pet by Id | Validation/Functional/Performance |
| PUT /pet | Update an existing pet | Validation/Functional/Performance |

**Note:** The endpoints automated in Task 2 covers only Validation and Functional testing types. As part of enhancement, Performance suite can be built up to test the strength of each endpoint.