NodeJS service pull request checklists

Upon every pull request, any reviewer should perform the following checklists, before merging the request.

Perform Unit Test and Integration Test

A valid project must have spec and integration test file for every pull request. And the following scripts must be available on the package.json.

1. validate, this script should check for eslint validity
2. test:unit, should parse all *.spec.js files for unit testing
3. test:integration, should parse all *.integration.test.js files for integration testing
4. test, should parse all *.spec.js and *.integration.test.js files for performing both unit and integration test.

A reviewer should run the following command:

```
$ yarn validate
$ yarn test
```

All lint rules and all tests must pass, if there's error that is not generated intentionally by the unit test even though the tests passes, the test considered failed, and request must be rejected.

The test performed must have code coverage reports in 2 formats:

1. lcov file
2. text report

The code coverage must at least reach 90% of all statements and branching. Developer can use BDD or TDD.

Manual test and fuzz the feature

After the eslint rules, unit tests and integration tests passes. Reviewer must perform manual test of the feature, by cloning the repo and perform various abusive tests like for example:

- Set numeric parameter to negative minimum number and positive maximum number, e.g. -999999999999999, 999999999999999
- Set string parameter to massive characters count, for example put string of 1000-2000 characters into the parameter.
- Set the appropriate parameters value to reflect the ideal response.

After putting those parameters, reviewer must see if the response is a suitable response based on parameter or not, if not then request must be rejected.

Example of suitable response, for example, we have this api:

POST: http://localhost:5000/api/v1/demo

params: type: numeric (0-9), name: string

Then we give the **type** parameter to 99999999, or -9999999999. The response should say something like validation error or invalid parameter.

And if we give the **name** parameter to string of more than 512, the response should say something like validation error, because the service did the name validation before processing the logic.

Sonarlint rules manual check

Review should take a look on rules on sonarlint to check on the pull requests if there's any rule broken.

The whole rules can be check on this https://rules.sonarsource.com/javascript.

Approval

The following git branch needs to be on PR:

- `develop`, developer can create PR with at least one reviewer.
- `master`, developer can create PR with 3 reviewers (1 main reviewer, 2 support reviewers).
- other branches, developer don't need any PR for other than 2 branches.

Once all the required steps completed, reviewer can then approve the pull request, where the lead reviewer then merge, after at least 2 reviews approval.