**REST API Mocha Test Runner Approach: Parallel Execution**

In Mocha Test Framework, we do have 2 approaches to run the test specs parallelly.

1. **Mocha-parallel-tests**

* mocha-parallel-tests executes **each of your test files in a separate process.**
* doesn't require you to write the code in a different way or use some specific APIs
* just run your tests with mocha-parallel-tests instead of mocha and you will see the difference.
* running I/O-heavy tests with mocha-parallel-tests can be a good thing to try if you want to keep your mocha tests and make them a bit faster. However, the price that you pay here is the RAM which is allocated for each subprocess.

Limitations :

1. Global Variables can’t be shared across test suites.
2. Not All mocha CLI options are supported in mocha-parallel-tests.
3. Parallel Limit : mocha-parallel-tests CLI executable has its own --max-parallel option which is the amount of tests executed at the same time. By default it's equal to the number of logical CPI cores (os.cpus().length) on your computer but you can also specify your own number or set it to 0, which means that all test files will be started executing at the same time. However this is not recommended especially on machines with low number of CPUs and big number of tests executed.

Observations :

Prerequisities :

1. Sample test suites per ts file.
2. Install npm : install --save-dev mocha-parallel-tests

Test Suites ran parallelly per ts. In a ts file, individual tests along with before and after hooks ran sequentially.

Took More time as compared to mocha.parallel Tool.

1. **mocha.parallel**

* is a tool which allows you to run mocha tests in parallel. while mocha-parallel-tests is more like a wrapper on top of mocha which is running all tests in subprocesses.
* Uncaught exceptions are associated with the spec that threw them, despite them all running at the same time. So debugging doesn't need to be too difficult!
* mocha.parallel is using one process, specific API to run tests and doesn't provide "true" parallelism, while mocha-parallel-tests forks subprocesses and is a mocha extension

Limitations :

1. Optimum for LightWeight Tests (less I/O).
2. Existing Test Code needed changes.
3. Global Variables can’t be shared across test suites.

Observations :

Prerequisities :

1. Sample test suites per ts file.
2. Install npm : npm install --save-dev mocha.parallel

Test Suites ran parallelly per ts. In a ts file, individual tests along with before and after hooks also ran parallely (coz of this we can’t use this our current framework).

Took less time as compared to mocha-parallel-tests wrapper.

**Reference Links :**

1. <https://github.com/mocha-parallel/mocha-parallel-tests/wiki/Comparison-with-mocha.parallel>
2. <https://github.com/danielstjules/mocha.parallel>
3. <https://github.com/mocha-parallel/mocha-parallel-tests>
4. <https://github.com/mocha-parallel/mocha-parallel-tests/wiki/Limitations>