

I have just cloned enigamampc/salad

. I am following the steps in the README to start the project. Everything works fine until I run yarn start

After a few minutes when the script gets to integrationTests/01_init.spec.js

, some of the tests fail:

✓ initializes (149ms) ✓ should generate and save key/pair (1ms) ✓ initializes Sample contract (4ms) ✓ should distribute ENG tokens (1187ms) ✗ should check that 1 worker(s) and the principal node, and only them, are registered (1374ms) ✓ should check worker's stake balance is empty (89ms) ✓ should get the worker parameters for the current block (133ms) ✗ should deposit tokens in worker banks (202ms) ✓ should check workers balances have been filled ✗ should login the worker(s) (20010ms) ✗ should check that 1 worker(s), and only them, are logged in (559ms) ✓ should move forward epochSize blocks by calling dummy contract (2780ms) ✗ should get the worker parameters for the current block (1115ms) ✗ should create getTaskEncryptionKey from core (with call to P2P) (4969ms)

More detailed output:

```
client_1 | ● Init tests › should check that 1 worker(s) and the principal node, and only them, are registered client_1 | client_1 |
expect(received).toEqual(expected) // deep equality client_1 | client_1 | - Expected client_1 | + Received client_1 | client_1 |
@@ -1,7 +1,7 @@ client_1 | Array [ client_1 | - 2, client_1 | + 0, client_1 | 0, client_1 | 0, client_1 | 0, client_1 | 0, client_1 | 0,
client_1 | client_1 | 92 | workerAddress[i] = await enigma.admin.getWorkerSignerAddr(accounts[i]); client_1 | 93 | } client_1 |
> 94 | expect(workerStatuses).toEqual(arrayResults); client_1 | | ^ client_1 | 95 | }, constants.TIMEOUT_INIT); client_1 | 96 |
client_1 | 97 | it('should check worker's stake balance is empty', async () => { client_1 | client_1 | at _callee4$
(test/integrationTests/01_init.spec.js:94:28) client_1 | at tryCatch (node_modules/regenerator-runtime/runtime.js:45:40)
client_1 | at Generator.invoke [as _invoke] (node_modules/regenerator-runtime/runtime.js:271:22) client_1 | at
Generator.prototype.(anonymous function) [as next] (node_modules/regenerator-runtime/runtime.js:97:21) client_1 | at
tryCatch (node_modules/regenerator-runtime/runtime.js:45:40) client_1 | at invoke (node_modules/regenerator-
runtime/runtime.js:135:20) client_1 | at node_modules/regenerator-runtime/runtime.js:145:13 client_1 | client_1 | ● Init tests ›
should deposit tokens in worker banks client_1 | client_1 | Failed: "Returned error: VM Exception while processing
transaction: revert Unregistered worker" client_1 | client_1 | 108 | client_1 | 109 | const deposit = 900; client_1 | > 110 |
it('should deposit tokens in worker banks', async () => { client_1 | | ^ client_1 | 111 | let promises = []; client_1 | 112 | for (let i
= 0; i < nodes; i++) { client_1 | 113 | let promise = new Promise((resolve, reject) => { client_1 | client_1 | at Env.it
(node_modules/jest-jasmine2/build/jasmineAsyncInstall.js:91:24) client_1 | at Suite.
(test/integrationTests/01_init.spec.js:110:3) client_1 | at Object. (test/integrationTests/01_init.spec.js:18:1) client_1 | client_1
| ● Init tests › should login the worker(s) client_1 | client_1 | : Timeout - Async callback was not invoked within the 20000ms
timeout specified by jest.setTimeout.Timeout - Async callback was not invoked within the 20000ms timeout specified by
jest.setTimeout.Error: client_1 | client_1 | 134 | }); client_1 | 135 | client_1 | > 136 | it('should login the worker(s)', async () =>
{ client_1 | | ^ client_1 | 137 | let promises = []; client_1 | 138 | for (let i = 0; i < nodes; i++) { client_1 | 139 | let promise = new
Promise((resolve, reject) => { client_1 | client_1 | at new Spec (node_modules/jest-jasmine2/build/jasmine/Spec.js:116:22)
client_1 | at Suite. (test/integrationTests/01_init.spec.js:136:3) client_1 | at Object. (test/integrationTests/01_init.spec.js:18:1)
client_1 | client_1 | ● Init tests › should check that 1 worker(s), and only them, are logged in client_1 | client_1 |
expect(received).toEqual(expected) // deep equality client_1 | client_1 | - Expected client_1 | + Received client_1 | client_1 |
@@ -1,7 +1,7 @@ client_1 | Array [ client_1 | - 1, client_1 | + 0, client_1 | 0, client_1 | 0, client_1 | 0, client_1 | 0, client_1 | 0,
client_1 | client_1 | 158 | arrayResults[i]=1; client_1 | 159 | } client_1 | > 160 | expect(workerStatuses).toEqual(arrayResults);
client_1 | | ^ client_1 | 161 | }); client_1 | 162 | client_1 | 163 | it('should move forward epochSize blocks by calling dummy
contract', async () => { client_1 | client_1 | at _callee10$ (test/integrationTests/01_init.spec.js:160:28) client_1 | at tryCatch
(node_modules/regenerator-runtime/runtime.js:45:40) client_1 | at Generator.invoke [as _invoke]
(node_modules/regenerator-runtime/runtime.js:271:22) client_1 | at Generator.prototype.(anonymous function) [as next]
(node_modules/regenerator-runtime/runtime.js:97:21) client_1 | at tryCatch (node_modules/regenerator-
runtime/runtime.js:45:40) client_1 | at invoke (node_modules/regenerator-runtime/runtime.js:135:20) client_1 | at
node_modules/regenerator-runtime/runtime.js:145:13 client_1 | client_1 | ● Init tests › should get the worker parameters for
the current block client_1 | client_1 | expect(received).toEqual(expected) // deep equality client_1 | client_1 | - Expected
client_1 | + Received client_1 | client_1 | - Array [ client_1 | - "0x0000000000000000000000000000000000000000000000000000000000000000", client_1
| - ] client_1 | + Array [] client_1 | client_1 | 178 | await sleep(1000); client_1 | 179 | } while (!workerParams) client_1 | > 180 |
expect(workerParams.workers.sort()).toEqual(workerAddress.sort()); // they may come in a different order client_1 | | ^
client_1 | 181 | expect(workerParams.stakes).toEqual(new Array(nodes).fill(JSBI.BigInt(900 * 10 ** 8))); client_1 | 182 | });
client_1 | 183 | client_1 | client_1 | at _callee12$ (test/integrationTests/01_init.spec.js:180:41) client_1 | at tryCatch
(node_modules/regenerator-runtime/runtime.js:45:40) client_1 | at Generator.invoke [as _invoke]
(node_modules/regenerator-runtime/runtime.js:271:22) client_1 | at Generator.prototype.(anonymous function) [as next]
(node_modules/regenerator-runtime/runtime.js:97:21) client_1 | at tryCatch (node_modules/regenerator-
runtime/runtime.js:45:40) client_1 | at invoke (node_modules/regenerator-runtime/runtime.js:135:20) client_1 | at
node_modules/regenerator-runtime/runtime.js:145:13 client_1 | client_1 | ● Init tests › should create getTaskEncryptionKey
from core (with call to P2P) client_1 | client_1 | : Timeout - Async callback was not invoked within the 5000ms timeout
specified by jest.setTimeout.Timeout - Async callback was not invoked within the 5000ms timeout specified by
jest.setTimeout.Error: client_1 | client_1 | 184 | const userPubKey =
'2ea8e4cefb78efd0725ed12b23b05079a0a433cc8a656f212accf58672fee44a20cfcaa50466237273e762e49ec'+ client_1 |
185 | '912be61358d5e90bff56a53a0ed42abfe27e3'; client_1 | > 186 | it('should create getTaskEncryptionKey from core (with
```

```
call to P2P)', async () => { client_1 | | ^ client_1 | 187 | const encryptionKeyResult = await new Promise((resolve, reject) => {
client_1 | 188 | enigma.client.request('getWorkerEncryptionKey', client_1 | 189 | {workerAddress:
workerAddress[0].toLowerCase().slice(-40), userPubKey: userPubKey}, (err, response) => { client_1 | client_1 | at new Spec
(node_modules/jest-jasmine2/build/jasmine/Spec.js:116:22) client_1 | at Suite. (test/integrationTests/01_init.spec.js:186:3)
client_1 | at Object. (test/integrationTests/01_init.spec.js:18:1) client_1 | Test Suites: 1 failed, 1 total client_1 | Tests: 6 failed,
8 passed, 14 total client_1 | Snapshots: 0 total client_1 | Time: 40.263s client_1 | Ran all test suites matching
/01_init.spec.js/i. contract_1 | eth_blockNumber client_1 | Jest did not exit one second after the test run has completed.
client_1 | client_1 | This usually means that there are asynchronous operations that weren't stopped in your tests. Consider
running Jest with --detectOpenHandles to troubleshoot this issue.
```

I haven't changed anything in the project after I cloned it except for what was in README, so I am not sure what the problem is. When creating a new dummy project with discovery init

all these integration tests pass.