

Final Project Report

200586

<https://github.com/maysa92/cloudMusicPackage>

Table of Contents

- Overview -----2
- Code and Branch Coverage -----3
- Mutant Testing -----4-7

Overview (_test.js)

- This project is to test a Netease API in Javascript.
- There are 4 test cases in my project for two main scenarios:
 - Song Searching (line 7 to line 30)
 - Lyrics Catching (line 34 to line 65)
- In Branch Coverage there are 2 branches: if an error occurs or not. Test cases are designed based on 2 core concepts: whether a successful connection is performed, and whether correct information is caught.
- Variables are thrown into 'createRequest' to see whether it can handle it precisely.

Code and Branch Coverage

- To generate an automated report of code and branch coverage in Mocha, I needed an additional library called istanbul.
- Figure 1 shows the full testing report.

```
> cloundmusicapi@0.0.0 test /Users/Jeffrey/Documents/GitHub/cloudMusicPackage/cloudMusicAPI
> nyc --reporter=text mocha
```

the searching

- ✓ Response Status Codes should be 200 (918ms)
- ✓ the name captured should be identical to the one entered (963ms)

the lyric fetching

- ✓ should not be empty (638ms)
- ✓ should have lrc (597ms)

4 passing (3s)

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	32.1	25	33.33	32.91	
crypto.js	17.07	100	0	17.95	... 57,58,59,60,61
util.js	47.5	25	50	47.5	... 6,78,84,90,111

Figure 1. the coverage report created by Istanbul after the test execution.

Mutant Testing

- Step 1:

A new branch “mutant 2” was created on Github for mutant testing.

- Step 2:

One syntactic element was changed, then the test suite was executed.

- Step 3:

Another syntactic element was then changed before the test suite was re-executed.

- Step 4:

2 separate testing reports were generated (see next 2 slides).

Notes:

The mutant created in Step 2 is to change the way the method utilized in the Song Searching function.

The extra statements added in Step 3 is to change the way the method “GET” is utilized by aborting the request.

Figure 2. the Step-2 report

```
the searching
  1) Response Status Codes should be 200
  2) the name captured should be identical to the one entered

the lyric fetching
  ✓ should not be empty (538ms)
  ✓ should have lrc (538ms)

2 passing (2s)
2 failing

1) the searching
   Response Status Codes should be 200:

   AssertionError [ERR_ASSERTION]: The expression evaluated to a falsy value:

   assert(rscore == 200)

   + expected - actual

   -false
   +true

   at /Users/Jeffrey/Documents/GitHub/cloudMusicPackage/cloudMusicAPI/mutant2/test/routers.test.js:14:9
   at processTicksAndRejections (internal/process/task_queues.js:93:5)

2) the searching
   the name captured should be identical to the one entered:
   TypeError: Cannot read property 'songs' of undefined
   at /Users/Jeffrey/Documents/GitHub/cloudMusicPackage/cloudMusicAPI/mutant2/test/routers.test.js:23:42
   at processTicksAndRejections (internal/process/task_queues.js:93:5)
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	30.86	12.5	33.33	31.65	
crypto.js	17.07	100	0	17.95	... 57,58,59,60,61
util.js	45	12.5	50	45	... ,84,90,111,122

npm ERR! Test failed. See above for more details.

```
120   }
121   if (method == 'PUT') { //change from POST to PUT
122     http_client.write(data)
123   }
124   http_client.end()
125 })
126 }
127 module.exports = {
128   createWebAPIRequest,
129   createRequest
130 }
131 |
```

Figure 3. the Step-2 mutant

Figure 4. the Step-3 report

```
1) the searching
   Response Status Codes should be 200:

   AssertionError [ERR_ASSERTION]: The expression evaluated to a falsy value:

   assert(rscode == 200)

   + expected - actual

   -false
   +true

   at /Users/Jeffrey/Documents/GitHub/cloudMusicPackage/cloudMusicAPI/mutant2/test/routers.test.js:15:9
   at processTicksAndRejections (internal/process/task_queues.js:93:5)

2) the searching
   the name captured should be identical to the one entered:
   TypeError: Cannot read property 'songs' of undefined
   at /Users/Jeffrey/Documents/GitHub/cloudMusicPackage/cloudMusicAPI/mutant2/test/routers.test.js:24:42
   at processTicksAndRejections (internal/process/task_queues.js:93:5)

3) the lyric fetching
   should not be empty:
   Error: Timeout of 2000ms exceeded. For async tests and hooks, ensure "done()" is called; if returning a
   mise, ensure it resolves. (/Users/Jeffrey/Documents/GitHub/cloudMusicPackage/cloudMusicAPI/mutant2/test/route
   test.js)
   at listOnTimeout (internal/timers.js:531:17)
   at processTimers (internal/timers.js:475:7)

4) the lyric fetching
   should have lrc:
   Error: Timeout of 2000ms exceeded. For async tests and hooks, ensure "done()" is called; if returning a
   mise, ensure it resolves. (/Users/Jeffrey/Documents/GitHub/cloudMusicPackage/cloudMusicAPI/mutant2/test/route
   test.js)
   at listOnTimeout (internal/timers.js:531:17)
   at processTimers (internal/timers.js:475:7)
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	32.53	30	33.33	33.33	
crypto.js	17.07	100	0	17.95	... 57,58,59,60,61
util.js	47.62	30	50	47.62	... ,84,90,111,122

```
120   )
121   if (method == 'PUT') { //change from POST to PUT
122     http_client.write(data)
123   }
124   //add extra wrong statements
125   if(method == 'GET') { You, a few seconds ago
126     http_client.abort()
127   }
128   http_client.end()
129 }
130 }
131 module.exports = {
132   createWebAPIRequest,
133   createRequest
134 }
135
```

Figure 5. the Step-3 mutants