TESTING DEVELOPMENT STORIES

The purpose of this document is to show evidence that the final solution meets the following user stories relating to the test environment.

TSV-U085 – As a test environment I should run unit tests

TSV-U086 – As a test environment I should have integration tests

TSV-U087 – As a test environment I should be able to stub data as to not make external data requests

TSV-U088 – As a test environment I should use assertions for the unit tests

TSV-U089 – As a test environment I should show coverage test results

TSV-U090 – As a test environment I should check dependencies are up-to-date

TSV-U091 – As a test environment I should check code quality with automated code reviews

TSV-U092 – As a test environment I should do headless browser testing and HTTP service testing

EXAMPLE UNIT TESTS

The following unit tests are taken from the *sentiment-analysis* module

test/utils.test.coffee

```
expect = require('chai').expect

process.env.NODE_ENV = 'test'

sentimentAnalysis = require('../index')._private


describe 'doesWordExist will return boolean weather word exists', ()->
  doesWordExist = sentimentAnalysis.doesWordExist

  it 'should return a boolean value', ()->
    expect(doesWordExist('coffee')).to.be.a('boolean')
    expect(doesWordExist('mocha')).to.be.a('boolean')
    expect(doesWordExist('java')).to.be.a('boolean')

  it 'should return true for words that exist', () ->
    expect(doesWordExist('woo')).to.be.true
    expect(doesWordExist('alive')).to.be.true
    expect(doesWordExist('awesome')).to.be.true
    expect(doesWordExist('anger')).to.be.true
    expect(doesWordExist('bright')).to.be.true
    expect(doesWordExist('love')).to.be.true
    expect(doesWordExist('easy')).to.be.true
    expect(doesWordExist('drunk')).to.be.true
    expect(doesWordExist('dumb')).to.be.true
    expect(doesWordExist('hacked')).to.be.true
    expect(doesWordExist('important')).to.be.true
    expect(doesWordExist('hug')).to.be.true
    expect(doesWordExist('itchy')).to.be.true
    expect(doesWordExist('laugh')).to.be.true
    expect(doesWordExist('stupid')).to.be.true
```

```coffeescript
    expect(doesWordExist('bomb')).to.be.true

  it 'should return false for words that do not exist', () ->
    expect(doesWordExist('hello')).to.be.false
    expect(doesWordExist('world')).to.be.false
    expect(doesWordExist('everything')).to.be.false
    expect(doesWordExist('is')).to.be.false
    expect(doesWordExist('stupidness')).to.be.false
    expect(doesWordExist('acid')).to.be.false
    expect(doesWordExist('dinosaurs')).to.be.false
    expect(doesWordExist('laptop')).to.be.false
    expect(doesWordExist('pepsi')).to.be.false
    expect(doesWordExist('lorem')).to.be.false
    expect(doesWordExist('ipsum')).to.be.false
    expect(doesWordExist('squashed')).to.be.false
    expect(doesWordExist('watson')).to.be.false
    expect(doesWordExist('brain')).to.be.false

  it 'should not throw an error with funny values', ()->
    expect(doesWordExist(1)).to.be.a('boolean')
    expect(doesWordExist([])).to.be.a('boolean')
    expect(doesWordExist(true)).to.be.a('boolean')
    expect(doesWordExist(undefined)).to.be.a('boolean')
    expect(doesWordExist(1)).to.be.false
    expect(doesWordExist([])).to.be.false
    expect(doesWordExist(undefined)).to.be.false



describe 'getScoreOfWord method return a sentiment score for that word', ()->
  getScoreOfWord = sentimentAnalysis.getScoreOfWord

  it 'should return an integer', ()->
    expect(getScoreOfWord('amazing')).to.be.a('number')
    expect(getScoreOfWord('warm')).to.be.a('number')
    expect(getScoreOfWord('yummy')).to.be.a('number')

  it 'should be in a range of -5 to + 5', () ->
    expect(getScoreOfWord('nice')).to.be.above(-5).to.be.below(5)
    expect(getScoreOfWord('good')).to.be.below(5).to.be.below(5)
    expect(getScoreOfWord('great')).to.be.above(-5).to.be.below(5)
    expect(getScoreOfWord('awesome')).to.be.above(-5).to.be.below(5)

  it 'should return 0 if word doesn\'t exist, rather than crashing', ()->
    expect(getScoreOfWord('batman')).equal(0)
    expect(getScoreOfWord('superman')).equal(0)
    expect(getScoreOfWord('spiderman')).equal(0)
    expect(getScoreOfWord('pepperpig')).equal(0)

  it 'should return 0 if passed multiple words at a time that don\'t exist', ()->
    expect(getScoreOfWord('type error')).equal(0)
    expect(getScoreOfWord('everything is stupid')).equal(0)
    expect(getScoreOfWord('dinosaurs are awesome')).equal(0)

  it 'should return actual positive score for positive words that exist', ()->
    expect(getScoreOfWord('united')).equal(1)
    expect(getScoreOfWord('unstoppable')).equal(2)
    expect(getScoreOfWord('excited')).equal(3)
    expect(getScoreOfWord('win')).equal(4)
    expect(getScoreOfWord('outstanding')).equal(5)

  it 'should return actual negative score for negative words that exist', ()->
    expect(getScoreOfWord('fight')).equal(-1)
    expect(getScoreOfWord('fails')).equal(-2)
    expect(getScoreOfWord('evil')).equal(-3)
    expect(getScoreOfWord('fraud')).equal(-4)
    expect(getScoreOfWord('twat')).equal(-5)
  it 'should return 0 for neutral words that exist', ()->
    expect(getScoreOfWord('some kind')).equal(0)
    # There is only 1 neutral result in the AFINN word list!

describe 'getWordsInSentence will transform a sentence into a clean array', ()->
  getWordsInSentence = sentimentAnalysis.getWordsInSentence

  it 'Should correctly turn a sentence into an array', ()->
    expect(getWordsInSentence('hello world')).eql(['hello', 'world'])
    expect(getWordsInSentence('this is a longer sentence'))
    .eql(['this', 'is', 'a', 'longer', 'sentence'])

  it 'Should normalise case', ()->
    expect(getWordsInSentence('HeLlO wOrLd')).eql(['hello', 'world'])
    expect(getWordsInSentence('JAVASCRIPT')).eql(['javascript'])

  it 'Should remove dupplicates', ()->
    expect(getWordsInSentence('foo foo bar foo'))
    .eql(['foo', 'bar'])
  expect(getWordsInSentence('foo foo BAR Foo bAr foO bar foo'))
  .eql(['foo', 'bar', ])

  it 'Should remove blanks', ()->
    expect(getWordsInSentence('space       blank        '))
    .eql(['space', 'blank'])

  it 'Should remove special characters', ()->
    expect(getWordsInSentence('foo ! ^&*^&^%^%&^^&%%^bar$$%^'))
    .eql(['foo', 'bar'])

describe 'removeDupplicates should remove dupplicates from an array', () ->
  removeDupplicates = sentimentAnalysis.removeDuplicates

  it 'should remove duplicates', () ->
    expect(removeDupplicates(['hello', 'world', 'hello', 'hello']))
    .eql(['hello', 'world'])

describe 'scaleScore should ensure the score is within the valid range', () ->
  scaleScore = sentimentAnalysis.scaleScore

  it 'should not be below -1', () ->
    expect(scaleScore(-1.2)).to.be.above(-1.01)
    expect(scaleScore(-38.8)).to.be.above(-1.01)
    expect(scaleScore(1.2)).to.be.above(-1.01)

  it 'should not be above +1', () ->
    expect(scaleScore(4.5)).to.be.below(1.01)
    expect(scaleScore(42)).to.be.below(1.01)
    expect(scaleScore(-1.2)).to.be.below(1.01)

  it 'should have 1 or 2 decimal places', () ->
    expect(scaleScore(1)).to.be.within(-1,+1);
    expect(scaleScore(-1)).to.be.within(-1,+1);
    expect(scaleScore(0)).to.be.within(-1,+1);
    expect(scaleScore(10)).to.be.within(-1,+1);
    expect(scaleScore(-1)).to.be.within(-1,+1);
    expect(scaleScore(-1.01)).to.be.within(-1,+1);
    expect(scaleScore(+1.0001)).to.be.within(-1,+1);
    expect(scaleScore(999999)).to.be.within(-1,+1);
    expect(scaleScore(-999999)).to.be.within(-1,+1);
    expect(scaleScore(-0)).to.be.within(-1,+1);
    expect(scaleScore(+0)).to.be.within(-1,+1);
    expect(scaleScore(42)).to.be.within(-1,+1);
    expect(scaleScore(3.1415926535897932)).to.be.within(-1,+1);
    expect(scaleScore(-273.15)).to.be.within(-1,+1);
```

test/main.test.coffee

```
expect = require('chai').expect

process.env.NODE_ENV = 'test'

sentimentAnalysis = require('../index').main


describe 'Check the modules basic functionality', ()->

  it 'should return an integer', () ->
    expect(sentimentAnalysis('lorem ipsum dolor seit amet'))
    .to.be.a('number')
    expect(sentimentAnalysis('foo bar')).to.not.be.undefined;

  it 'Should return the correct sentiment value for negative sentences', () ->
    expect(sentimentAnalysis('I hate everything, everything is stupid')).equal(-
0.5)
    expect(sentimentAnalysis('London is gloomy today because of all the
smog')).equal(-0.4)
    expect(sentimentAnalysis('He was captured and put into slavery')).equal(-0.3)
    expect(sentimentAnalysis('Windows is very unstable')).equal(-0.2)
    expect(sentimentAnalysis('The slug was tired, he felt slugish')).equal(-0.2)

  it 'Should return the correct sentiment value for positive sentences', () ->
    expect(sentimentAnalysis('Today is a wonderful amazing awesome day')).equal(1)
    expect(sentimentAnalysis('I am so grateful for all the presents, thank
you!')).equal(0.5)

  it 'Should not return a score greater than 1 of smaller than -1', () ->
    expect(sentimentAnalysis('happy happy amazing awesome cool'))
    .to.be.above(-1.1).to.be.below(1.1)
    expect(sentimentAnalysis('crap crap crap crap'))
    .to.be.above(-1.1).to.be.below(1.1)


  it 'Should be able to cope with weird inputs and never crash', ()->
```

TEST CONFIGURATION

test/mocha.opts

```
--compilers coffee:coffee-script/register
--reporter spec
```

.travis.yml

```
language: node_js
node_js:
  - "0.12"
  - "0.10"

before_script:
    - "npm i -g mocha"
```

Running the tests

All tests can be run by running the command ‘npm test’ or ‘gulp test’

The gulp task which runs all tests is as follows:

```
/* Run unit tests and generate coverage report */
gulp.task('test', function (cb) {
    gulp.src(['./index.js'])
        .pipe(istanbul())
        .pipe(istanbul.hookRequire())
        .on('finish', function () {
            gulp.src('./test/**/*.coffee', {read: false})
                .pipe(mocha({ reporter: 'spec' }))
                .pipe(istanbul.writeReports({reporters: ['text-summary', 'lcov']}))
                //.pipe(istanbul.enforceThresholds({ thresholds: { global: 90 } }))
                .on('end', cb);
        });
});
```

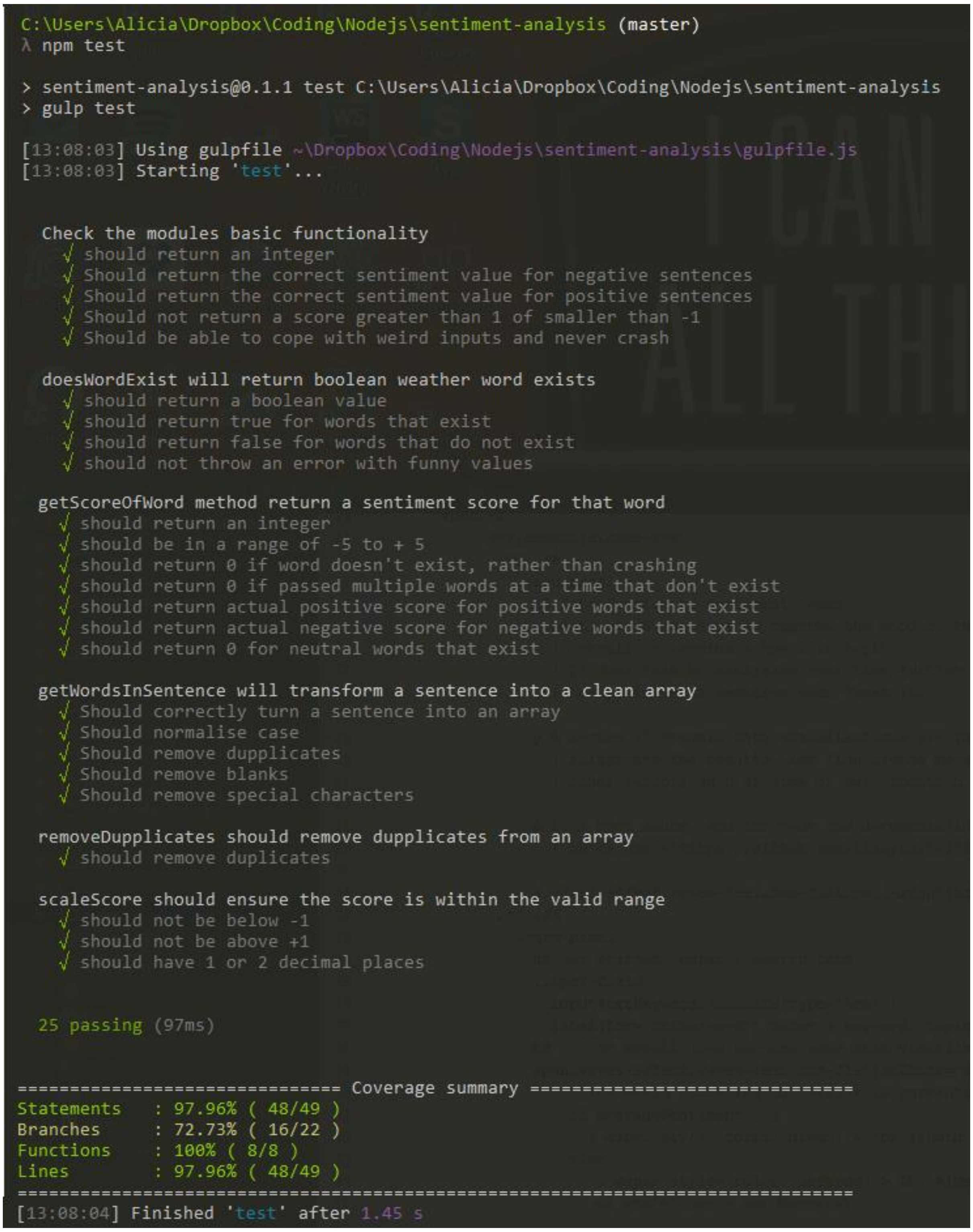
Exporting appropriate methods

If developing in the test environment (as opposed to production), we also export private methods, which allows them to be unit tested. For example:

```
if process.env.NODE_ENV == 'test'
  module.exports =
    main: analyseSentence
    _private:
      scaleScore: scaleScore
      doesWordExist: doesWordExist
      getScoreOfWord: getScoreOfWord
      removeDuplicates: removeDuplicates
      getWordsInSentence: getWordsInSentence
```

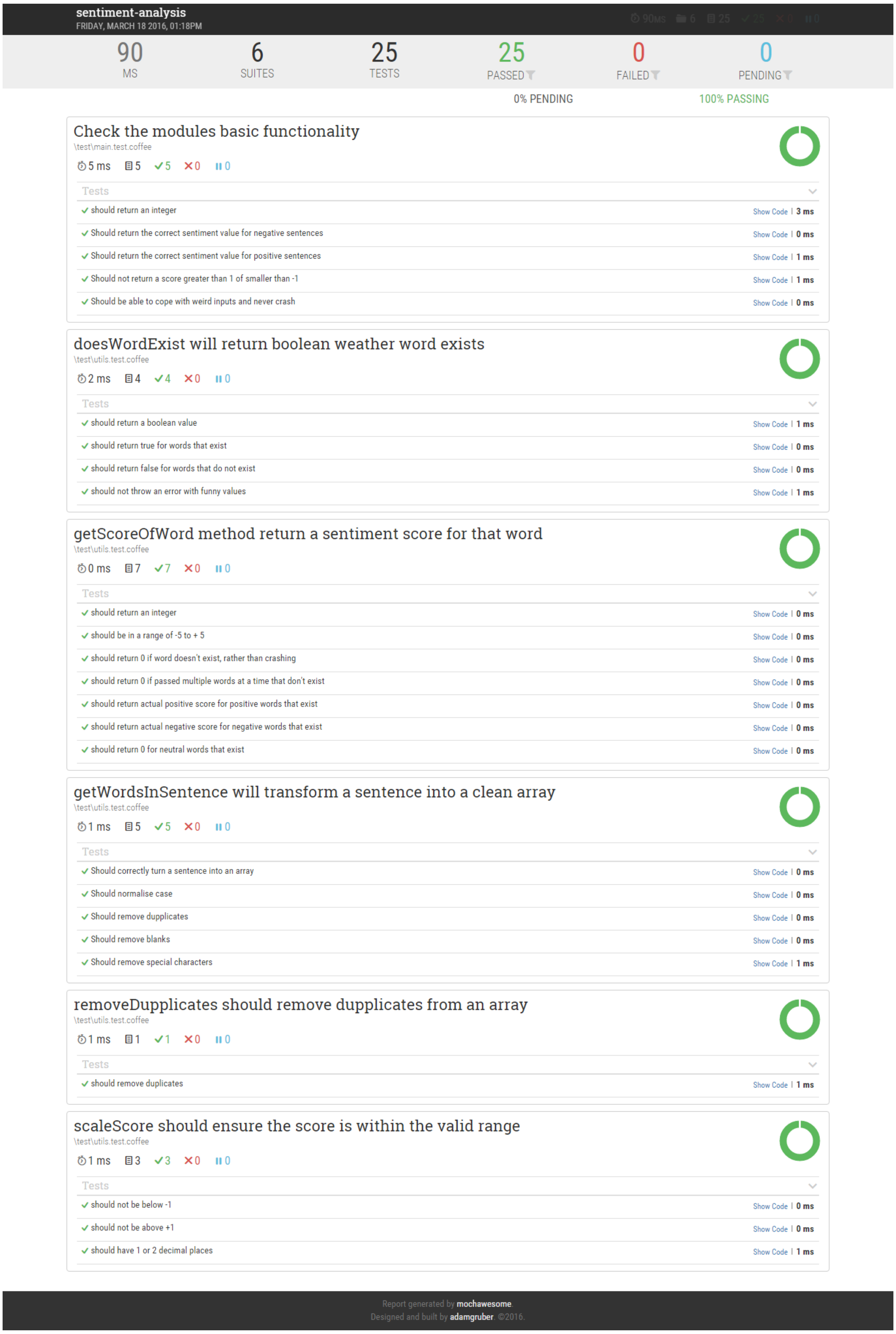
TEST RESULTS

Example console output printed after tests are run



```
C:\Users\Alicia\Dropbox\Coding\Nodejs\sentiment-analysis (master)
λ npm test

> sentiment-analysis@0.1.1 test C:\Users\Alicia\Dropbox\Coding\Nodejs\sentiment-analysis
> gulp test

[13:08:03] Using gulpfile ~\Dropbox\Coding\Nodejs\sentiment-analysis\gulpfile.js
[13:08:03] Starting 'test'...


  Check the modules basic functionality
    √ should return an integer
    √ Should return the correct sentiment value for negative sentences
    √ Should return the correct sentiment value for positive sentences
    √ Should not return a score greater than 1 of smaller than -1
    √ Should be able to cope with weird inputs and never crash

  doesWordExist will return boolean weather word exists
    √ should return a boolean value
    √ should return true for words that exist
    √ should return false for words that do not exist
    √ should not throw an error with funny values

  getScoreOfWord method return a sentiment score for that word
    √ should return an integer
    √ should be in a range of -5 to + 5
    √ should return 0 if word doesn't exist, rather than crashing
    √ should return 0 if passed multiple words at a time that don't exist
    √ should return actual positive score for positive words that exist
    √ should return actual negative score for negative words that exist
    √ should return 0 for neutral words that exist

  getWordsInSentence will transform a sentence into a clean array
    √ Should correctly turn a sentence into an array
    √ Should normalise case
    √ Should remove dupplicates
    √ Should remove blanks
    √ Should remove special characters

  removeDupplicates should remove dupplicates from an array
    √ should remove duplicates

  scaleScore should ensure the score is within the valid range
    √ should not be below -1
    √ should not be above +1
    √ should have 1 or 2 decimal places


  25 passing (97ms)


=============================== Coverage summary ===============================
Statements   : 97.96% ( 48/49 )
Branches     : 72.73% ( 16/22 )
Functions    : 100% ( 8/8 )
Lines        : 97.96% ( 48/49 )
================================================================================
[13:08:04] Finished 'test' after 1.45 s
```

Example of the visual report generated for unit test results



sentiment-analysis
FRIDAY, MARCH 18 2016, 01:18PM

90	6	25	25	0	0
MS	SUITES	TESTS	PASSED	FAILED	PENDING

0% PENDING   100% PASSING

Check the modules basic functionality
\test\main.test.coffee
5 ms 5 ✓5 ✕0 0

Tests	
✓ should return an integer	Show Code | 3 ms
✓ Should return the correct sentiment value for negative sentences	Show Code | 0 ms
✓ Should return the correct sentiment value for positive sentences	Show Code | 1 ms
✓ Should not return a score greater than 1 of smaller than -1	Show Code | 1 ms
✓ Should be able to cope with weird inputs and never crash	Show Code | 0 ms

doesWordExist will return boolean weather word exists
\test\utils.test.coffee
2 ms 4 ✓4 ✕0 0

Tests	
✓ should return a boolean value	Show Code | 1 ms
✓ should return true for words that exist	Show Code | 0 ms
✓ should return false for words that do not exist	Show Code | 0 ms
✓ should not throw an error with funny values	Show Code | 1 ms

getScoreOfWord method return a sentiment score for that word
\test\utils.test.coffee
0 ms 7 ✓7 ✕0 0

Tests	
✓ should return an integer	Show Code | 0 ms
✓ should be in a range of -5 to + 5	Show Code | 0 ms
✓ should return 0 if word doesn't exist, rather than crashing	Show Code | 0 ms
✓ should return 0 if passed multiple words at a time that don't exist	Show Code | 0 ms
✓ should return actual positive score for positive words that exist	Show Code | 0 ms
✓ should return actual negative score for negative words that exist	Show Code | 0 ms
✓ should return 0 for neutral words that exist	Show Code | 0 ms

getWordsInSentence will transform a sentence into a clean array
\test\utils.test.coffee
1 ms 5 ✓5 ✕0 0

Tests	
✓ Should correctly turn a sentence into an array	Show Code | 0 ms
✓ Should normalise case	Show Code | 0 ms
✓ Should remove dupplicates	Show Code | 0 ms
✓ Should remove blanks	Show Code | 0 ms
✓ Should remove special characters	Show Code | 1 ms

removeDupplicates should remove dupplicates from an array
\test\utils.test.coffee
1 ms 1 ✓1 ✕0 0

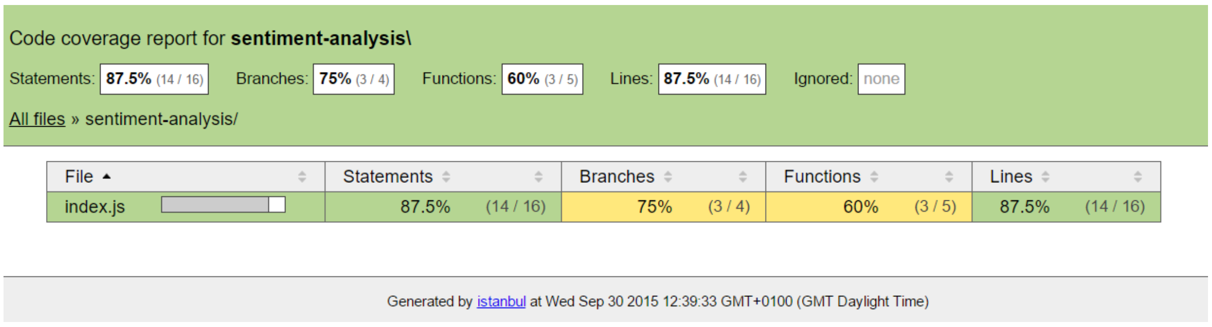
Tests	
✓ should remove duplicates	Show Code | 1 ms

scaleScore should ensure the score is within the valid range
\test\utils.test.coffee
1 ms 3 ✓3 ✕0 0

Tests	
✓ should not be below -1	Show Code | 0 ms
✓ should not be above +1	Show Code | 0 ms
✓ should have 1 or 2 decimal places	Show Code | 1 ms

Report generated by **mochawesome**.
Designed and built by **adamgruber**. ©2016.

Example of the visual report generated for coverage results



Code coverage report for **sentiment-analysis**

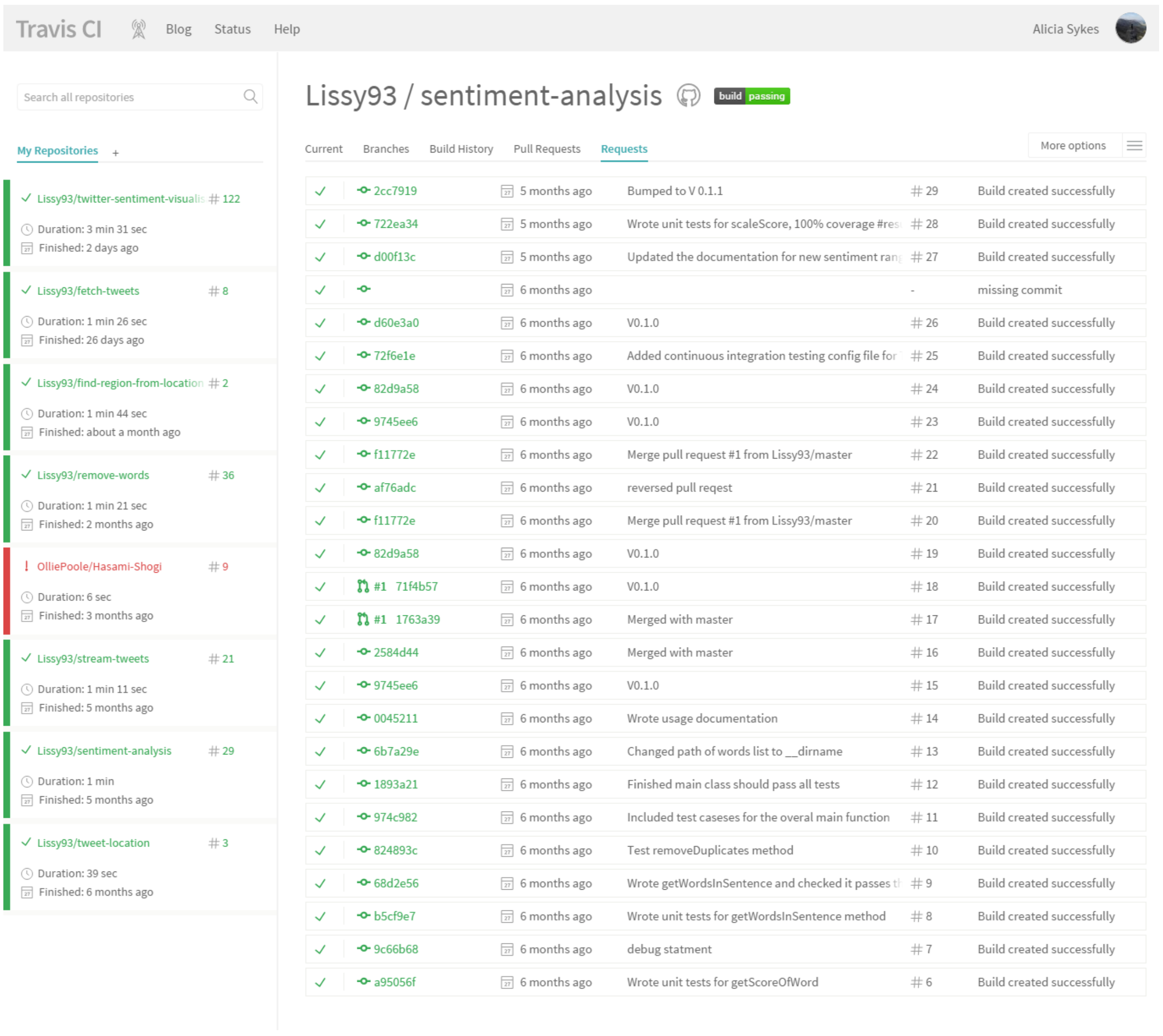
Statements: **87.5%** (14 / 16) Branches: **75%** (3 / 4) Functions: **60%** (3 / 5) Lines: **87.5%** (14 / 16) Ignored: none

All files » sentiment-analysis/

File		Statements		Branches		Functions		Lines	
index.js		87.5%	(14 / 16)	75%	(3 / 4)	60%	(3 / 5)	87.5%	(14 / 16)

Generated by istanbul at Wed Sep 30 2015 12:39:33 GMT+0100 (GMT Daylight Time)

Example of the report generated for automated continuous integration testing



✓ Lissy93/twitter-sentiment-visuali # 122
Duration: 3 min 31 sec
Finished: 2 days ago

✓ Lissy93/fetch-tweets # 8
Duration: 1 min 26 sec
Finished: 26 days ago

✓ Lissy93/find-region-from-locatio # 2
Duration: 1 min 44 sec
Finished: about a month ago

✓ Lissy93/remove-words # 36
Duration: 1 min 21 sec
Finished: 2 months ago

! OlliePoole/Hasami-Shogi # 9
Duration: 6 sec
Finished: 3 months ago

✓ Lissy93/stream-tweets # 21
Duration: 1 min 11 sec
Finished: 5 months ago

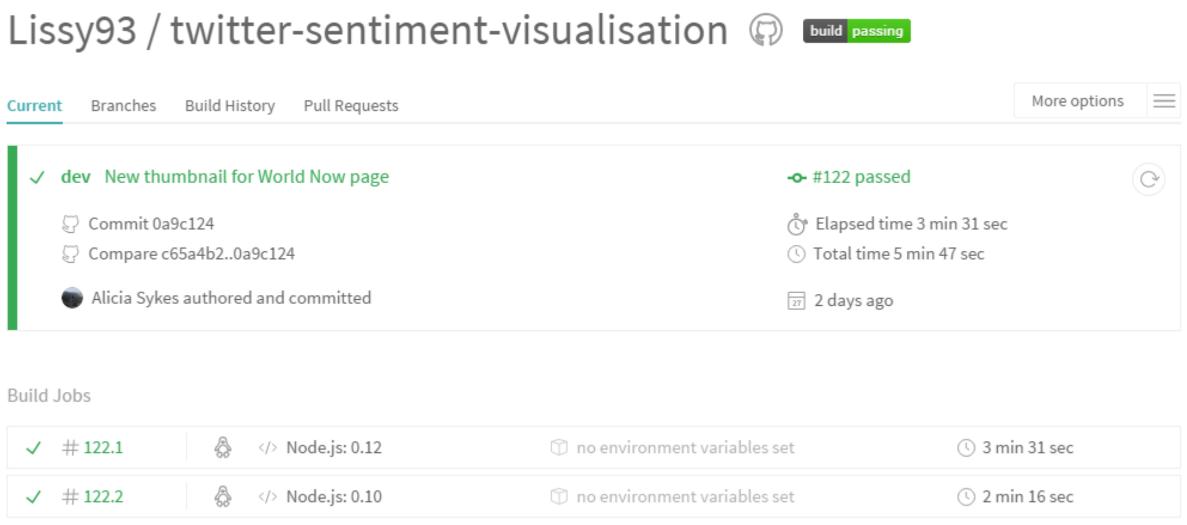
✓ Lissy93/sentiment-analysis # 29
Duration: 1 min
Finished: 5 months ago

✓ Lissy93/tweet-location # 3
Duration: 39 sec
Finished: 6 months ago

Lissy93 / sentiment-analysis build passing

✓	2cc7919	5 months ago	Bumped to V 0.1.1	# 29	Build created successfully
✓	722ea34	5 months ago	Wrote unit tests for scaleScore, 100% coverage #res	# 28	Build created successfully
✓	d00f13c	5 months ago	Updated the documentation for new sentiment ran	# 27	Build created successfully
✓		6 months ago		-	missing commit
✓	d60e3a0	6 months ago	V0.1.0	# 26	Build created successfully
✓	72f6e1e	6 months ago	Added continuous integration testing config file for	# 25	Build created successfully
✓	82d9a58	6 months ago	V0.1.0	# 24	Build created successfully
✓	9745ee6	6 months ago	V0.1.0	# 23	Build created successfully
✓	f11772e	6 months ago	Merge pull request #1 from Lissy93/master	# 22	Build created successfully
✓	af76adc	6 months ago	reversed pull reqest	# 21	Build created successfully
✓	f11772e	6 months ago	Merge pull request #1 from Lissy93/master	# 20	Build created successfully
✓	82d9a58	6 months ago	V0.1.0	# 19	Build created successfully
✓	#1 71f4b57	6 months ago	V0.1.0	# 18	Build created successfully
✓	#1 1763a39	6 months ago	Merged with master	# 17	Build created successfully
✓	2584d44	6 months ago	Merged with master	# 16	Build created successfully
✓	9745ee6	6 months ago	V0.1.0	# 15	Build created successfully
✓	0045211	6 months ago	Wrote usage documentation	# 14	Build created successfully
✓	6b7a29e	6 months ago	Changed path of words list to __dirname	# 13	Build created successfully
✓	1893a21	6 months ago	Finished main class should pass all tests	# 12	Build created successfully
✓	974c982	6 months ago	Included test caseses for the overal main function	# 11	Build created successfully
✓	824893c	6 months ago	Test removeDuplicates method	# 10	Build created successfully
✓	68d2e56	6 months ago	Wrote getWordsInSentence and checked it passes t	# 9	Build created successfully
✓	b5cf9e7	6 months ago	Wrote unit tests for getWordsInSentence method	# 8	Build created successfully
✓	9c66b68	6 months ago	debug statment	# 7	Build created successfully
✓	a95056f	6 months ago	Wrote unit tests for getScoreOfWord	# 6	Build created successfully

Example of summary of build status


Lissy93 / twitter-sentiment-visualisation build passing

Current Branches Build History Pull Requests More options

✓ dev New thumbnail for World Now page #122 passed

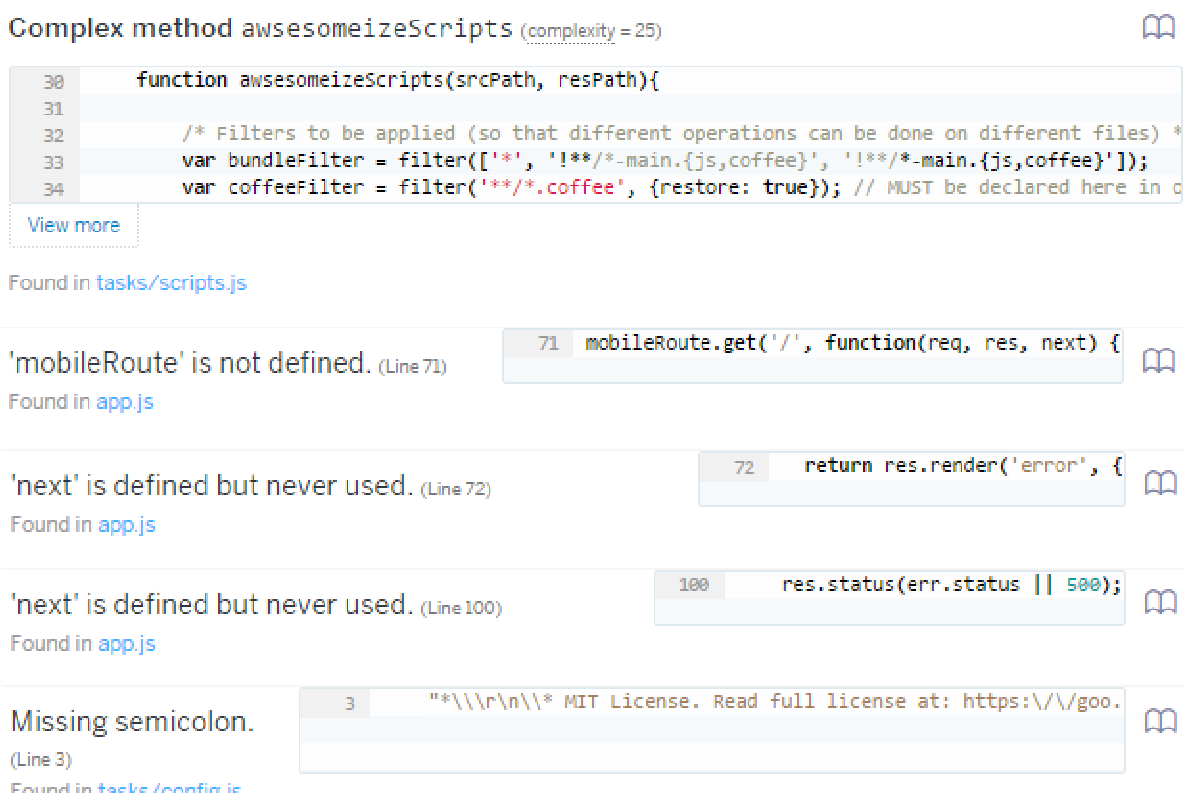
Commit 0a9c124
Compare c65a4b2..0a9c124
Alicia Sykes authored and committed

Elapsed time 3 min 31 sec
Total time 5 min 47 sec
2 days ago

Build Jobs

✓	# 122.1	Node.js: 0.12	no environment variables set	3 min 31 sec
✓	# 122.2	Node.js: 0.10	no environment variables set	2 min 16 sec

Example of automated code review reports


Complex method awsesomeizeScripts (complexity = 25)

```
function awsesomeizeScripts(srcPath, resPath){

    /* Filters to be applied (so that different operations can be done on different files) *
    var bundleFilter = filter(['*', '!**/*-main.{js,coffee}', '!**/*-main.{js,coffee}']);
    var coffeeFilter = filter('**/*.coffee', {restore: true}); // MUST be declared here in c
```

View more

Found in tasks/scripts.js

'mobileRoute' is not defined. (Line 71)
Found in app.js

```
71  mobileRoute.get('/', function(req, res, next) {
```

'next' is defined but never used. (Line 72)
Found in app.js

```
72      return res.render('error', {
```

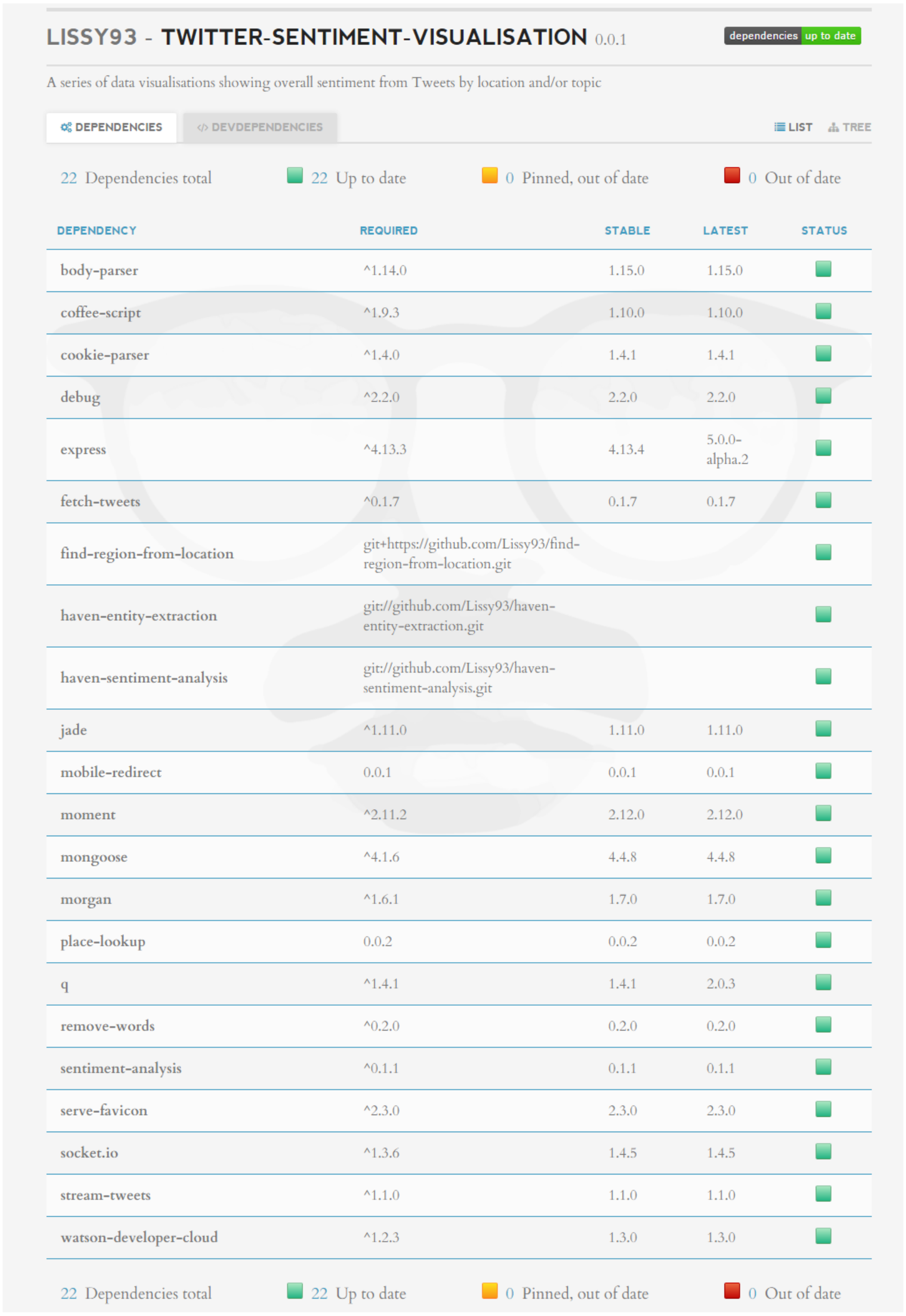
'next' is defined but never used. (Line 100)
Found in app.js

```
100     res.status(err.status || 500);
```

Missing semicolon.
(Line 3)
Found in tasks/config.js

```
3      "*\\\r\n\\* MIT License. Read full license at: https:\/\/goo.
```

Example of automated dependency checking report



LISSY93 - **TWITTER-SENTIMENT-VISUALISATION** 0.0.1

dependencies up to date

A series of data visualisations showing overall sentiment from Tweets by location and/or topic

DEPENDENCIES | DEVDEPENDENCIES | LIST | TREE

22 Dependencies total 🟩 22 Up to date 🟨 0 Pinned, out of date 🟥 0 Out of date

DEPENDENCY	REQUIRED	STABLE	LATEST	STATUS
body-parser	^1.14.0	1.15.0	1.15.0	🟩
coffee-script	^1.9.3	1.10.0	1.10.0	🟩
cookie-parser	^1.4.0	1.4.1	1.4.1	🟩
debug	^2.2.0	2.2.0	2.2.0	🟩
express	^4.13.3	4.13.4	5.0.0-alpha.2	🟩
fetch-tweets	^0.1.7	0.1.7	0.1.7	🟩
find-region-from-location	git+https://github.com/Lissy93/find-region-from-location.git			🟩
haven-entity-extraction	git://github.com/Lissy93/haven-entity-extraction.git			🟩
haven-sentiment-analysis	git://github.com/Lissy93/haven-sentiment-analysis.git			🟩
jade	^1.11.0	1.11.0	1.11.0	🟩
mobile-redirect	0.0.1	0.0.1	0.0.1	🟩
moment	^2.11.2	2.12.0	2.12.0	🟩
mongoose	^4.1.6	4.4.8	4.4.8	🟩
morgan	^1.6.1	1.7.0	1.7.0	🟩
place-lookup	0.0.2	0.0.2	0.0.2	🟩
q	^1.4.1	1.4.1	2.0.3	🟩
remove-words	^0.2.0	0.2.0	0.2.0	🟩
sentiment-analysis	^0.1.1	0.1.1	0.1.1	🟩
serve-favicon	^2.3.0	2.3.0	2.3.0	🟩
socket.io	^1.3.6	1.4.5	1.4.5	🟩
stream-tweets	^1.1.0	1.1.0	1.1.0	🟩
watson-developer-cloud	^1.2.3	1.3.0	1.3.0	🟩

22 Dependencies total 🟩 22 Up to date 🟨 0 Pinned, out of date 🟥 0 Out of date