



## programming for testers



testing with JavaScript, Mocha and Chai

Erik Haartmans

Martijn Thiele

# Agenda



- Finalizing laptop setup
- Create test with JavaScript
  - Using Mocha and Chai
- Create RESTful API test
- Create Selenium webdriver test

# Required, otherwise you can't start ...



- Hardware
  - Laptop with wifi
- Installed software
  - Visual Studio Code
  - Node.js  $\geq 8.6$
  - Chrome browser  $\geq v60$



- Slack account



# Workshop will be at high speed!



- Because we only have 2 hours
- It's not a complete course
- We want to cover 2 topics
  - RESTful API testing
  - Selenium webdriver testing
- We use **github gists** during the workshop

# Steps to be taken now ...



- Connect to wifi
  - SSID:
  - UN:
  - PW:

# create test development folder

- e.g. d:\testing\testnet-pft



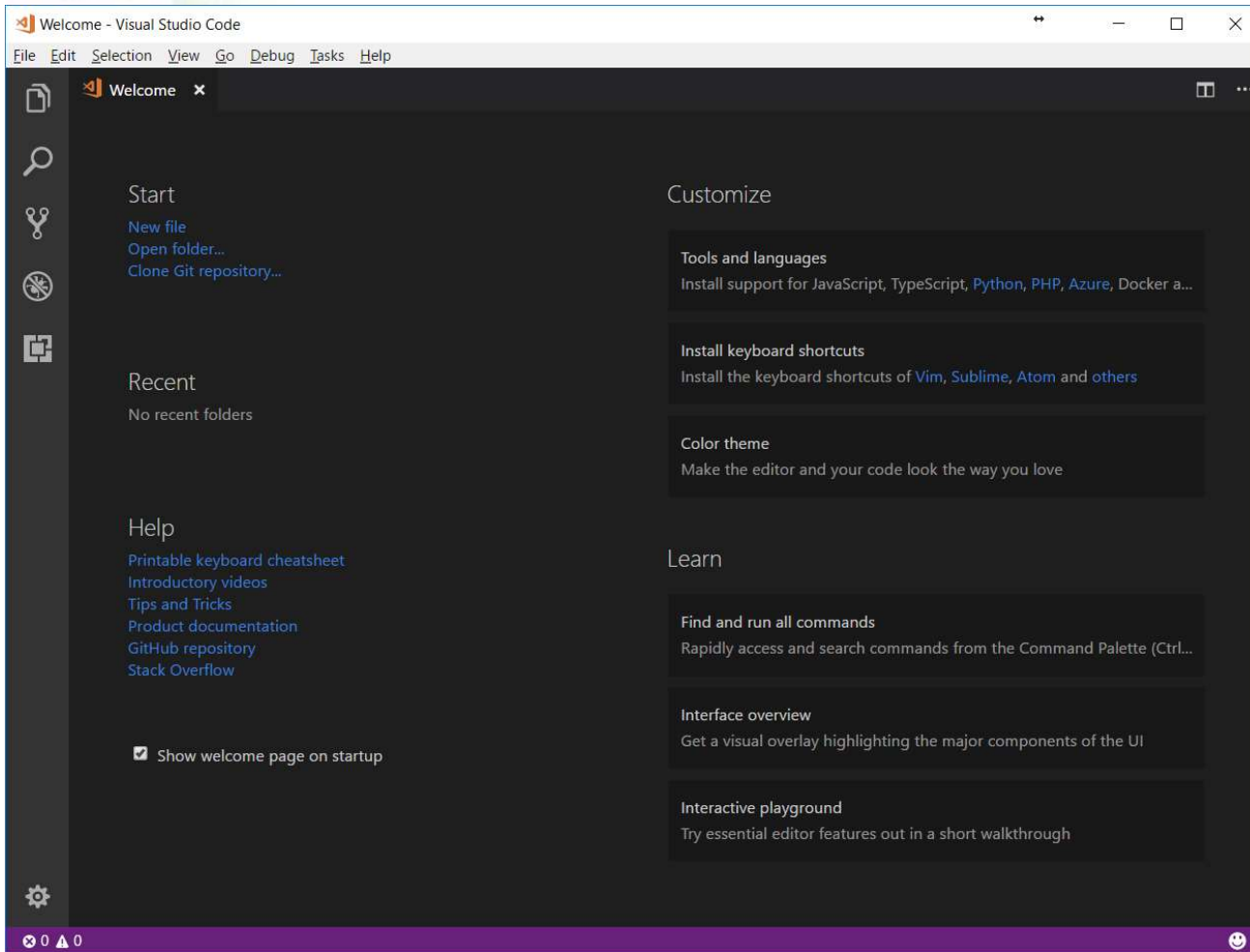
# Start Visual Studio Code



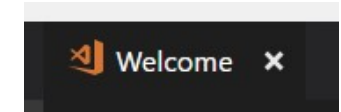
Visual Studio Code

Bureaublad-app

# Start Visual Studio Code



- Start menu...
  - File / Open folder
- Browse to ...
  - `d:\testing\testnet-pft`
  - and select this folder
- Close 'Welcome' tab





# Open terminal



Show All Commands    Ctrl+Shift+P

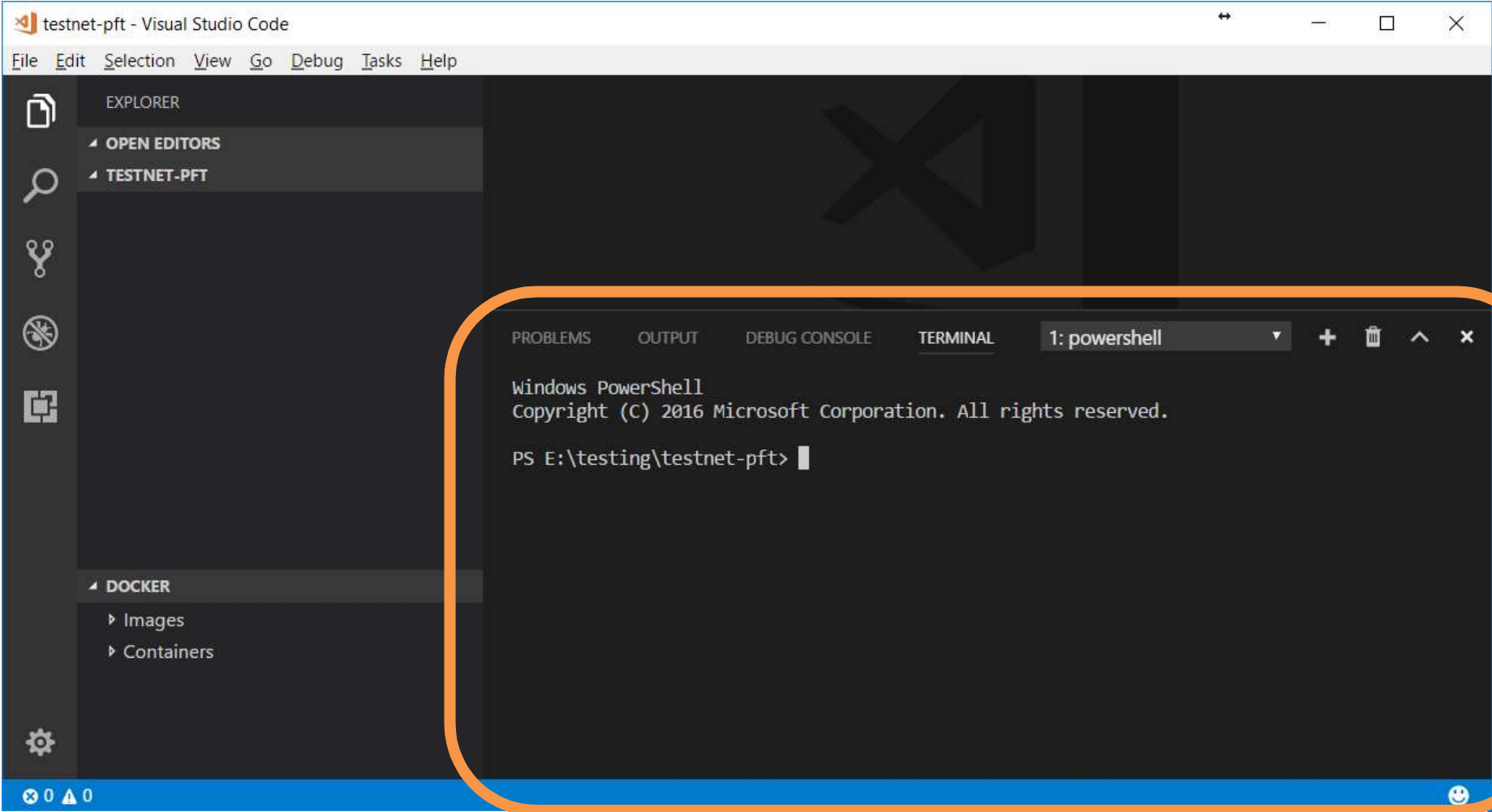
Go to File    Ctrl+P

Find in Files    Ctrl+Shift+F

Start Debugging    F5

**Toggle Terminal    Ctrl+`**

# Open terminal



# Check versions



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Windows PowerShell  
Copyright (c) 2016 Microsoft Corporation. All rights reserved.  
  
PS D:\testing\testnet-pft> node -v  
v8.9.0  
PS D:\testing\testnet-pft> npm -v  
5.5.1  
PS D:\testing\testnet-pft> █
```

- node -v  
– v8.9.0
- npm -v  
– 5.5.1

# Init project



- `npm init`
  - package name: (testnet-pft) <enter>
  - version: (1.0.0) <enter>
  - description: testproject<enter>
  - entry point: (index.js) <enter>
  - test command: mocha <enter>
  - git repository: <enter>
  - keywords: <enter>
  - author: <your name> <enter>
  - license: (ISC) <enter>
  - Is this ok? (yes) <enter>

A screenshot of a terminal window with a dark background. At the top, there are four tabs: 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'TERMINAL' tab is active. The terminal shows the command prompt 'PS D:\testing\testnet-pft>' followed by the command 'npm init' with a cursor at the end.

# package.json



```
{ } package.json x
1  {
2    "name": "testnet-pft",
3    "version": "1.0.0",
4    "description": "testproject",
5    "main": "index.js",
6    "scripts": {
7      "test": "mocha"
8    },
9    "author": "Erik Haartmans",
10   "license": "ISC"
11 }
12
```

# Install npm packages



- `npm install -g mocha`
- `npm install --save-dev mocha`
- `npm install --save-dev chai`

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

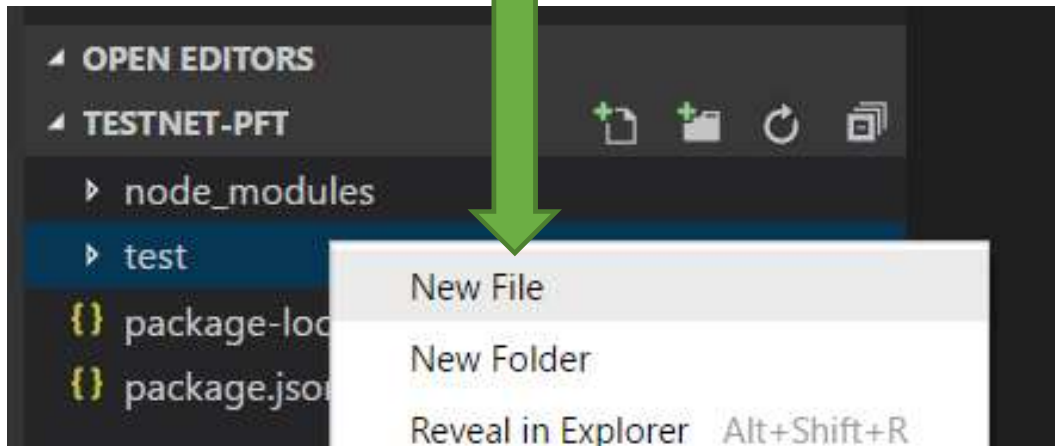
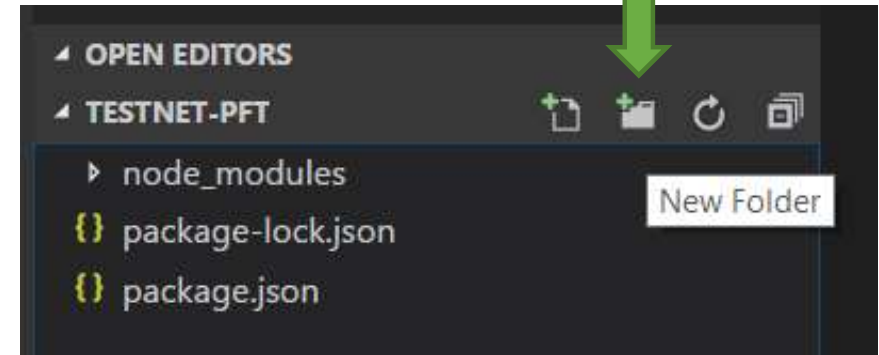
PS D:\testing\testnet-pft> npm install -g mocha
C:\Users\Erik\AppData\Roaming\npm\mocha -> C:\Users\Erik\AppData\Roaming\npm\node_modules\mocha\bin\mocha
C:\Users\Erik\AppData\Roaming\npm\_mocha -> C:\Users\Erik\AppData\Roaming\npm\node_modules\mocha\bin\_mocha
+ mocha@4.0.1
updated 1 package in 1.811s
PS D:\testing\testnet-pft> npm install --save-dev mocha
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN testnet-pft@1.0.0 No repository field.

+ mocha@4.0.1
added 24 packages in 2.8s
PS D:\testing\testnet-pft> npm install --save-dev chai
npm WARN testnet-pft@1.0.0 No repository field.

+ chai@4.1.2
added 7 packages in 2.268s
PS D:\testing\testnet-pft> █
```

# Create test folder

- Click new folder icon next to project name in VSC
  - create folder 'test'
- Right click on 'test' folder
  - 'new file' => 'my-first-test.js'



# Add test code



```
JS my-first-test.js x
1 |
```





## Add test code

- Go to: <https://gist.github.com/erikhpoltq>
- View: pft-01-my-first-test.js
- Click on 'Raw'



`pft-01-my-first-test.js`

Raw

```
1 describe('my first test suite', () => {
2
3   it('my first test', () => {
4
5   })
6
7 })
```

- Copy text from browser and paste into my-first-test.js

# Add test code



```
JS my-first-test.js ×  
1 describe('my first test suite', () => {  
2  
3     it('my first test', () => {  
4  
5     })  
6  
7 })  
8 |
```

- Save code!

# Run test

- In terminal
  - npm test



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\testing\testnet-pft> npm test
```

# Run test

- 1 test passing!

```
JS my-first-test.js x
1 describe('my first test suite', () => {
2
3     it('my first test', () => {
4
5     })
6
7 })
8

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS D:\testing\testnet-pft> npm test

> testnet-pft@1.0.0 test D:\testing\testnet-pft
> mocha

my first test suite
  ✓ my first test

1 passing (18ms)

PS D:\testing\testnet-pft>
```



# Run test

- In terminal
  - mocha



```
JS my-first-test.js x
1 describe('my first test suite', () => {
2
3     it('my first test', () => {
4
5     })
6
7 })
8
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

```
PS D:\testing\testnet-pft> mocha

my first test suite
  ✓ my first test

1 passing (25ms)

PS D:\testing\testnet-pft> 
```

# Add assertion



- Above describe:
  - `const chai = require('chai');`
  - `chai.should();`
- In test
  - `const x = 5;`
  - `x.should.equal(4);`
- **Save code!**

```
JS my-first-test.js x
1  const chai = require('chai');
2  chai.should();
3
4  describe('my first test suite', () => {
5
6      it('my first test', () => {
7          const x = 5;
8          x.should.equal(4);
9      })
10
11 })
12
```

# Run test again

- In terminal
  - npm test

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS D:\testing\testnet-pft> npm test

> testnet-pft@1.0.0 test D:\testing\testnet-pft
> mocha

my first test suite
  1) my first test

0 passing (14ms)
1 failing

1) my first test suite
   my first test:

   AssertionError: expected 5 to equal 4
   + expected - actual

   -5
   +4

   at Context.it (test\my-first-test.js:8:22)

npm ERR! Test failed.  See above for more details.
PS D:\testing\testnet-pft> █
```



# Adjust test



```
JS my-first-test.js x
1  const chai = require('chai');
2  chai.should();
3
4  describe('my first test suite', () => {
5
6      it('my first test', () => {
7          const x = 5;
8          x.should.equal(5);
9      })
10
11 })
12
```

- Save code!



# Run test again

- In terminal
  - npm test

```
JS my-first-test.js x
1  const chai = require('chai');
2  chai.should();
3
4  describe('my first test suite', () => {
5
6      it('my first test', () => {
7          const x = 5;
8          x.should.equal(5);
9      })
10
11  })
12
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

```
PS D:\testing\testnet-pft> npm test
> testnet-pft@1.0.0 test D:\testing\testnet-pft
> mocha

my first test suite
  ✓ my first test

1 passing (3ms)

PS D:\testing\testnet-pft> █
```

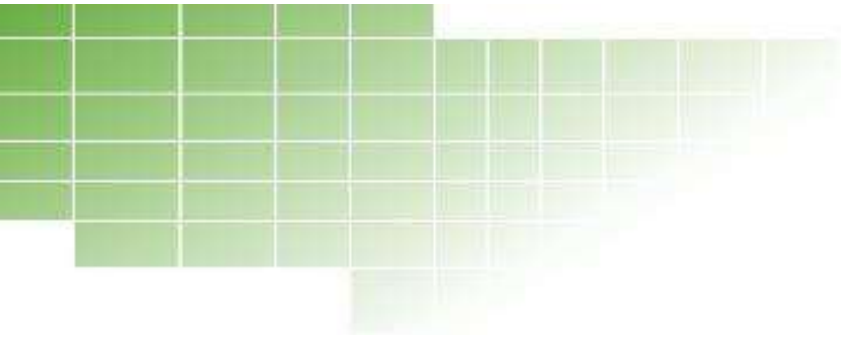


# We have created our first test!



- Using:
  - JavaScript
  - Mocha
  - Chai



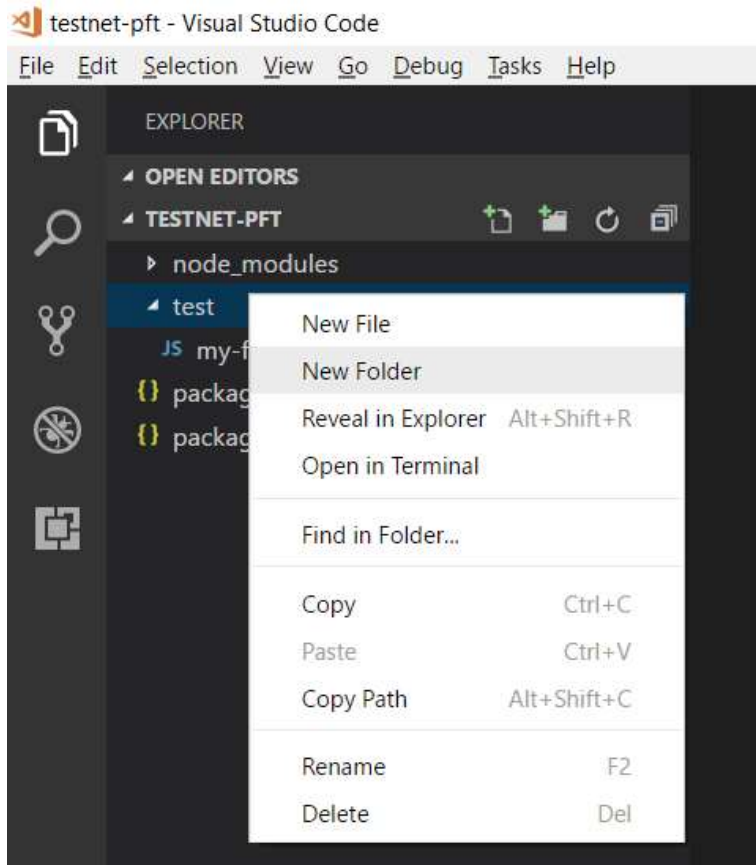


## RESTful API testing

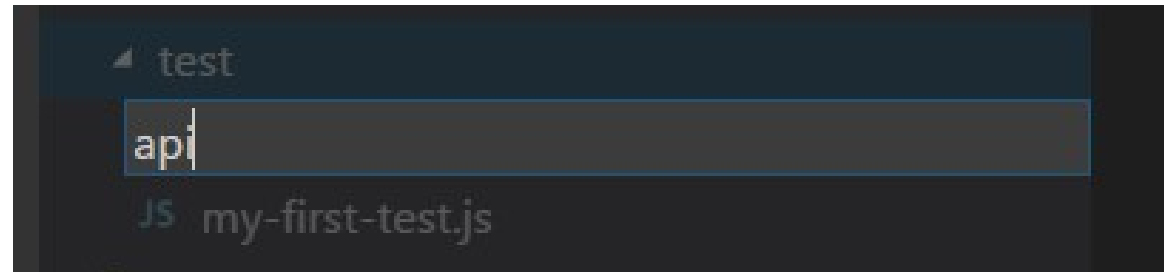
# Create new 'api' folder



- Add new folder 'api' in 'test' folder



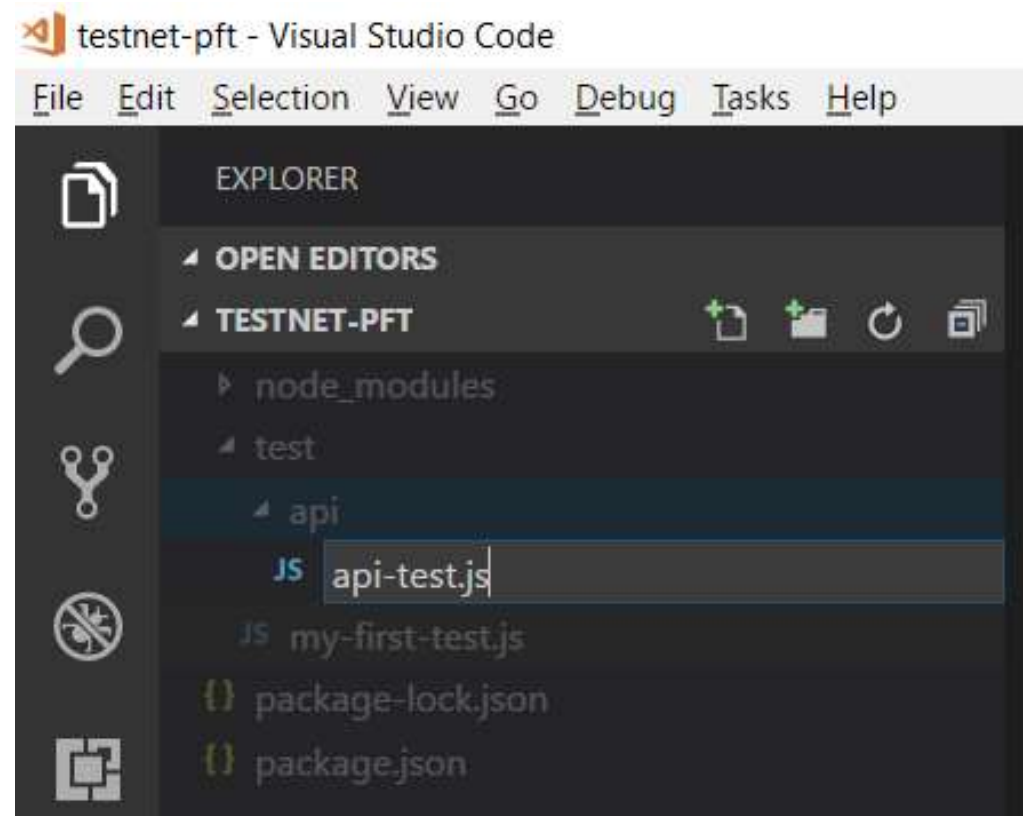
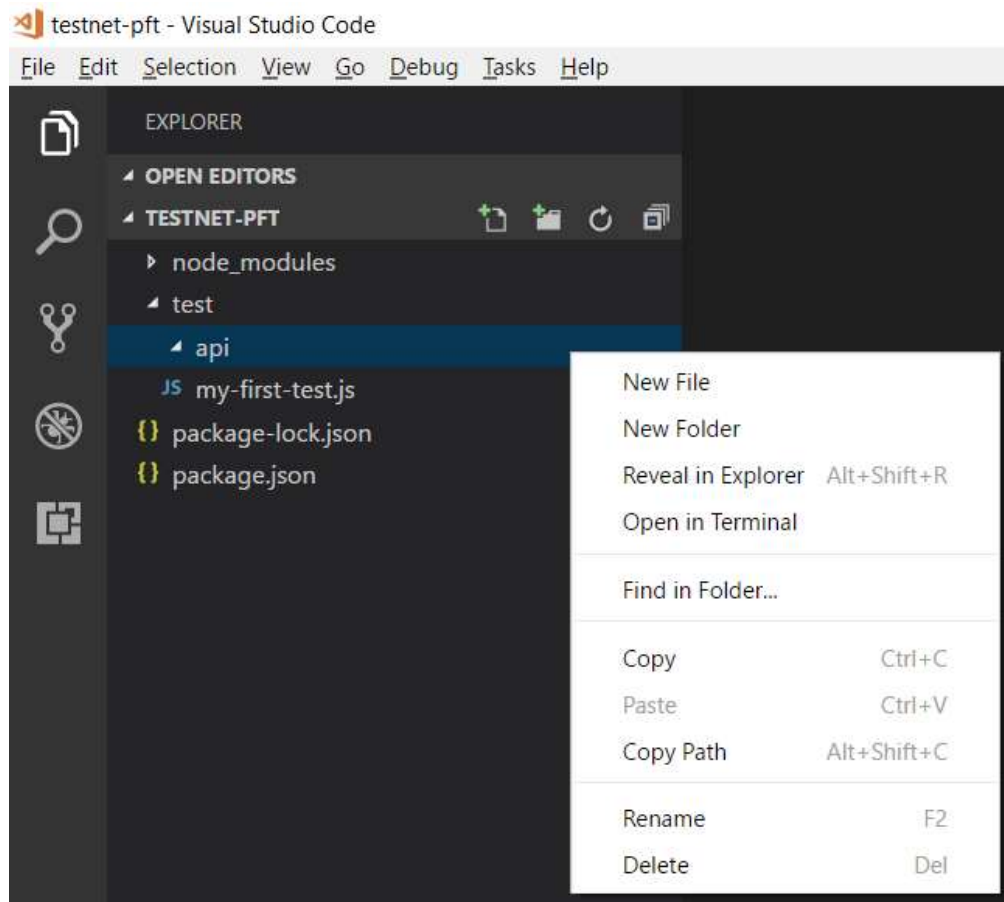
- Right click on 'test' folder



# Add testfile to 'api' folder



- Add new file 'api-test.js' in 'api' folder.



# Copy test

- Copy/paste test from my-first-test
- Save code!



```
JS api-test.js x JS my-first-test.js
1  const chai = require('chai');
2  chai.should();
3
4  describe('my first test suite', () => {
5
6      it('my first test', () => {
7          const x = 5;
8          x.should.equal(5);
9      })
10
11 })
12
```

# Run tests

- In terminal
  - npm test



```
JS api-test.js x JS my-first-test.js
1  const chai = require('chai');
2  chai.should();
3
4  describe('my first test suite', () => {
5
6      it('my first test', () => {
7          const x = 5;
8          x.should.equal(5);
9      })
10
11 })
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\testing\testnet-pft> npm test
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\testing\testnet-pft> npm test
> testnet-pft@1.0.0 test D:\testing\testnet-pft
> mocha

my first test suite
  ✓ my first test

1 passing (12ms)

PS D:\testing\testnet-pft> 
```

Only 1 test!

But we have 2 tests

# mocha.opts



- Create file 'mocha.opts' in 'test' folder

```
mocha.opts x
1  --recursive
2  --timeout 30000
3  |
```

- Test again (npm test)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\testing\testnet-pft> npm test
> testnet-pft@1.0.0 test D:\testing\testnet-pft
> mocha

my first test suite
  ✓ my first test

my first test suite
  ✓ my first test

2 passing (26ms)
```



# Install supertest



- Install supertest
  - `npm install --save-dev supertest`

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS D:\testing\testnet-pft> npm install --save-dev supertest
npm WARN testnet-pft@1.0.0 No repository field.

+ supertest@3.0.0
added 22 packages in 3.518s
PS D:\testing\testnet-pft> █
```

- <https://www.npmjs.com/package/supertest>

# Create RESTful API test

- Delete code from 'api-test.js'
- Copy and paste code from gist
  - <https://gist.github.com/erikhpoltEQ>
  - pft-02-api-test.js



# Create RESTful API test



```
JS api-test.js X
1  const chai = require('chai');
2  chai.should();
3
4  const request = require('supertest')('http://api.icndb.com/jokes');
5
6  describe('chuck norris test suite', () => {
7
8      it('random api call should return http 200', () => {
9          const response = request.get('/random');
10         response.statusCode.should.equal(200, 'response status should equal 200');
11         console.log(response.body.value.joke);
12     })
13
14 })
15
```

# Run RESTful API test



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\testing\testnet-pft> npm test

> testnet-pft@1.0.0 test D:\testing\testnet-pft
> mocha

chuck norris test suite
  1) random api call should return http 200

my first test suite
  ✓ my first test

1 passing (15ms)
1 failing

1) chuck norris test suite
   random api call should return http 200:
   TypeError: Cannot read property 'should' of undefined
   at Context.it (test\api\api-test.js:10:29)

npm ERR! Test failed.  See above for more details.
PS D:\testing\testnet-pft> █
```

# JavaScript is asynchronous



```
JS api-test.js X
1  const chai = require('chai');
2  chai.should();
3
4  const request = require('supertest')('http://api.icndb.com/jokes');
5
6  describe('chuck norris test suite', () => {
7
8      it('random api call should return http 200', () => {
9          const response = request.get('/random');
10         response.statusCode.should.equal(200, 'response status should equal 200');
11         console.log(response.body.value.joke);
12     })
13
14 })
15
```

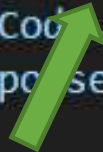

- Code is 'scheduled' => doesn't wait for answer

# Use async/await



JS api-test.js x

```
1  const chai = require('chai');
2  chai.should();
3
4  const request = require('supertest')('http://api.icndb.com/jokes');
5
6  describe('chuck norris test suite', () => {
7
8      it('random api call should return http 200', async () => {
9          const response = await request.get('/random');
10         response.statusCode.should.equal(200, 'response status should equal 200');
11         console.log(response.body.value.joke);
12     })
13
14 })
15
```



# Run RESTful API test again



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: powershell

```
PS D:\testing\testnet-pft> npm test
```

```
> testnet-pft@1.0.0 test D:\testing\testnet-pft
> mocha
```

```
  chuck norris test suite
```

```
They had to edit the first ending of 'Lone Wolf McQuade' after Chuck Norris kicked David Carradine's ass, then proceeded to barbecue and eat him.
```

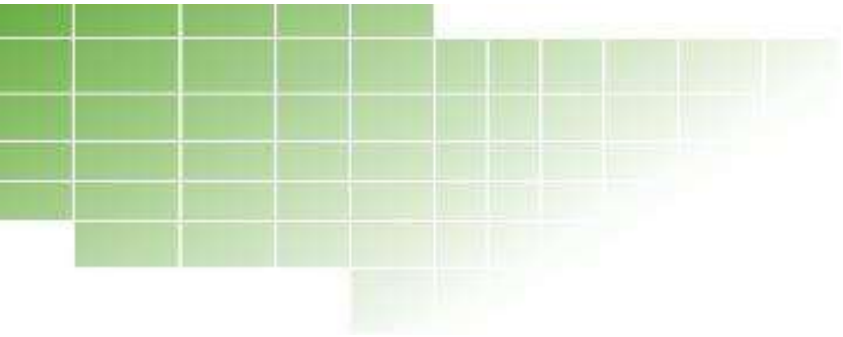
```
  ✓ random api call should return http 200 (99ms)
```

```
  my first test suite
```

```
  ✓ my first test
```

```
  2 passing (115ms)
```





## Selenium UI testing



# Install selenium & chromedriver



- `npm install --save-dev selenium-webdriver chromedriver`

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1:
PS D:\testing\testnet-pft> npm install --save-dev selenium-webdriver chromedriver
> chromedriver@2.33.2 install D:\testing\testnet-pft\node_modules\chromedriver
> node install.js

Downloading https://chromedriver.storage.googleapis.com/2.33/chromedriver_win32.zip
Saving to C:\Users\Erik\AppData\Local\Temp\chromedriver\chromedriver_win32.zip
Received 781K...
Received 1568K...
Received 2352K...
Received 3136K...
Received 3920K...
Received 4125K total.
Extracting zip contents
Copying to target path D:\testing\testnet-pft\node_modules\chromedriver\lib\chromedriver
Done. ChromeDriver binary available at D:\testing\testnet-pft\node_modules\chromedriver\lib\chromedriver\chromedriver.exe
npm WARN testnet-pft@1.0.0 No repository field.

+ chromedriver@2.33.2
+ selenium-webdriver@3.6.0
updated 2 packages in 5.532s
PS D:\testing\testnet-pft> █
```

# Create first UI test

- Create 'ui' folder in 'test' folder
- Create file 'selenium-test-01.js' in 'ui' folder
- Copy and paste code from gist
  - <https://gist.github.com/erikhpoltEQ>
  - pft-03-selenium-test-01.js



# Create first UI test



```
JS selenium-test-01.js x
1  require('chromedriver');
2  const webdriver = require('selenium-webdriver');
3
4  describe('selenium test - part 1', () => {
5
6      it('should be able to start a website', async () => {
7          let driver = new webdriver.Builder()
8              .forBrowser('chrome')
9              .build();
10         driver.manage().window().maximize();
11         await driver.navigate().to('https://techblog.polteq.com/testshop/index.php');
12         driver.sleep(5000);
13         driver.quit();
14     })
15
16 })
17 |
```

# Run test



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS D:\testing\testnet-pft> npm test

> testnet-pft@1.0.0 test D:\testing\testnet-pft
> mocha

  chuck norris test suite
  Chuck Norris knows the value of NULL, and he can sort by it too.
    ✓ random api call should return http 200 (102ms)

  my first test suite
    ✓ my first test

  selenium test - part 1
    ✓ should be able to start a website (4953ms)

3 passing (5s)
```

## Create second UI test

- Create 'selenium-test-02.js' in 'ui' folder
- Copy and paste code from gist
  - <https://gist.github.com/erikhpoltEQ>
  - pft-04-selenium-test-02.js



# Create second UI test



JS selenium-test-02.js x

```
1  require('chromedriver');
2  const webdriver = require('selenium-webdriver');
3  const By = webdriver.By;
4
5  describe('selenium test - part 2', () => {
6
7      it('should be able to start a website', async () => {
8          let driver = new webdriver.Builder()
9              .forBrowser('chrome')
10             .build();
11         driver.manage().window().maximize();
12         await driver.navigate().to('https://techblog.polteq.com/testshop/index.php');
13         let searchField = await driver.findElement(By.id('search_query_top'));
14         searchField.sendKeys('iPod' + webdriver.Key.ENTER);
15         let result = await driver.findElement(By.className('heading-counter'));
16         let resultText = await result.getText();
17         resultText.should.equal('5 results have been found.');
```



# Run test



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\testing\testnet-pft> npm test
```

```
> testnet-pft@1.0.0 test D:\testing\testnet-pft
> mocha
```

```
  chuck norris test suite
```

```
It is believed dinosaurs are extinct due to a giant meteor. That's true if you want to call Chuck Norris a giant meteor.
```

```
  ✓ random api call should return http 200 (67ms)
```

```
  my first test suite
```

```
  ✓ my first test
```

```
  selenium test - part 1
```

```
  ✓ should be able to start a website (5054ms)
```

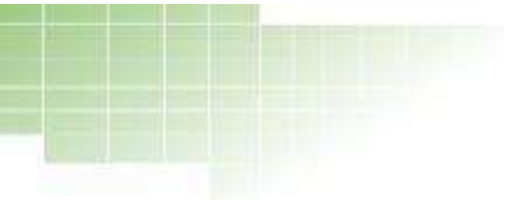
```
  selenium test - part 2
```

```
  ✓ should be able to start a website (4908ms)
```

```
 4 passing (10s)
```







## UI test with beforeEach & afterEach

- Create 'selenium-test-03.js' in 'ui' folder
- Copy and paste code from gist
  - <https://gist.github.com/erikhpoltEQ>
  - pft-05-selenium-test-03.js

# UI test with beforeEach & afterEach

```
JS selenium-test-03.js x
1  require('chromedriver');
2  const webdriver = require('selenium-webdriver');
3
4  describe('selenium test - part 3', () => {
5
6      let driver;
7
8      beforeEach(async () => {
9          driver = new webdriver.Builder()
10             .forBrowser('chrome')
11             .build();
12         driver.manage().window().maximize();
13         await driver.navigate().to('https://techblog.polteq.com/testshop/index.php');
14     })
15     afterEach(async () => {
16         await driver.quit();
17     })
18
19     it('should be able to do the first test', async () => {
20         // first test here
21     })
22     it('should be able to do the second test', async () => {
23         // second test here
24     })
25
26 })
27
```

# RESTful API test with query parameters

- <http://api.icndb.com/jokes/random?firstName=John&lastName=Doe>
- Create 'api-test-query-parameters.js' in 'api' folder
- Copy and paste code from gist
  - <https://gist.github.com/erikhpoltEQ>
  - pft-06-api-test-query-parameters.js

# RESTful API test with query parameters

```
JS api-test-query-parameters.js x
1  const chai = require('chai');
2  chai.should();
3
4  const request = require('supertest')('http://api.icndb.com/jokes');
5
6  describe('chuck norris query parameters test suite', () => {
7
8      it('should be able to use query parameters', async () => {
9
10         const response = await request.get('/random')
11             .query({
12                 firstName: 'John',
13                 lastName: 'Doe'
14             });
15
16         response.statusCode.should.equal(200, 'response status should equal 200');
17         console.log(response.body.value.joke);
18     })
19
20 })
21
```