

SQA<sup>®</sup>  
—————  
DAYS #27

# **TestCafe. Выжать все из функционального web-тестирования**

Елена Дикарева, DevExpress



# Обо мне

- Работала над несколькими E2E-фреймворками
- Senior developer & project manager
- Более 8 лет работаю над TestCafe

# О чем я расскажу?

- Современные инструменты для функционального тестирования
- Продвинутое возможности фреймворков на примере TestCafe

# Что было раньше?

- Selenium
- Selenium-based решения
- Протокол WebDriver

# Недостатки WebDriver

– Сложная настройка окружения на CI

# Недостатки WebDriver

- Сложная настройка окружения на CI
- **Нестабильные, мигающие тесты**

# Недостатки WebDriver

- Сложная настройка окружения на CI
- Нестабильные, мигающие тесты
- **Проблемы с очисткой состояние между тестовыми прогонами**

# Недостатки WebDriver

- Сложная настройка окружения на CI
- Нестабильные, мигающие тесты
- Проблемы с очисткой состояние между тестовыми прогонами
- **Сложности доступа к элементам тестируемой страницы**

# Современные фреймворки

- Внедряются в тестовые страницы
- Минимум дополнительных настроек
- Более стабильные тесты
- Доступ к элементам тестируемой страницы
- JavaScript

# Требования к новому фреймворку

- Простая установка, работает без дополнительного ПО
- Запуск тестов локально и удаленно в разных окружениях и браузерах
- Встроенный механизм поддержки стабильности тестов
- Не требуется освоение новых языков



github.com

Search or jump to... Pull requests Issues Marketplace Explore

DevExpress / testcafe Unwatch releases 187 Star 8.4k Fork 554

<> Code Issues 605 Pull requests 8 Actions Projects 5 Wiki Security Insights

master 11 branches 302 tags Go to file Add file Code

AlexSkorkin [docs] Add new contributors (#5505)	72c8c3f 2 days ago	2,324 commits
.github [docs] Add the Pull Request Template (#4759)		7 months ago
.md-lint [docs] Merge Documentation 2.0 (#5051)		4 months ago
@types Rewrite the browser/connection/remotes-queue.js file to TypeScript (...)		9 months ago
bin Provide info about emulated device in User-Agent (closes #3290) (#5...		3 months ago
docker Fix docker tests (#5392)		last month
docs [docs] DevOps post - add links to CI topics		7 days ago
examples Fix incorrect indentation in the BitBucket config		10 months ago

**About**  
A Node.js tool to automate end-to-end web testing.  
[devexpress.github.io/testcafe/](https://devexpress.github.io/testcafe/)

testcafe end-to-end-testing test-automation test-framework functional-testing e2e testing-tools test browser typescript

Readme MIT License

# Free & Open source



# Cross browsers



# Cross platform (Win, macOS, Linux)



# Plugin free



# Для любых сайтов



# Подходит для сложных сценариев



# **Установка и 'Hello word' тест**

```
~ admin$ npm install -g testcafe
```

```
~ admin$ testcafe --list-browsers
```

```
~ admin$ testcafe --list-browsers  
firefox  
chrome  
edge  
opera  
safari
```

```
fixture('A basic example that illustrate how to use TestCafe API')
  .page('http://devexpress.github.io/testcafe/example/');

test('A simple test with form sending', async t => {
  await t
    .typeText('#developer-name', 'Peter Parker')
    .click('input[name="re-using"]')
    .click('#macos')
    .click(Selector('#submit-button'));
});
```

```
~ admin$ testcafe chrome my-test.js
```

JS basic.js

```

sqa-days > JS basic.js > ...
1  import { Selector } from 'testcafe';
2
3  fixture('A basic example that illustrate how to use TestCafe API')
4    .page('http://devexpress.github.io/testcafe/example/');
5
6  test('A simple test with form sending', async t => {
7    await t
8      .typeText('#developer-name', 'Peter Parker')
9      .click('input[name="re-using"]')
10     .click('#macos')
11     .click(Selector('#submit-button'));
12  });

```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

2: zsh

→ sqa-days



```
fixture('A basic example that illustrate how to use TestCafe API')
  .page('http://devexpress.github.io/testcafe/example/');

test('A simple test with form sending', async t => {
  await t
    .typeText('#developer-name', 'Peter Parker')
    .click('input[name="re-using"]')
    .click('#macos')
    .click(Selector('#submit-button'));
});
```

```
fixture('A basic example that illustrate how to use TestCafe API')
  .page('http://devexpress.github.io/testcafe/example/');

test('A simple test with form sending', async t => {
  await t
    .typeText('#developer-name', 'Peter Parker')
    .click('input[name="re-using"]')
    .click('#macos')
    .click(Selector('#submit-button'));
});
```

```
fixture('A basic example that illustrate how to use TestCafe API')
  .page('http://devexpress.github.io/testcafe/example/');

test('A simple test with form sending', async t => {
  await t
    .typeText('#developer-name', 'Peter Parker')
    .click('input[name="re-using"']')
    .click('#macos')
    .click(Selector('#submit-button'));
});
```

```
fixture('A basic example that illustrate how to use TestCafe API')
  .page('http://devexpress.github.io/testcafe/example/');

test('A simple test with form sending', async t => {
  await t
    .typeText('#developer-name', 'Peter Parker')
    .click('input[name="re-using"]')
    .click('#macos')
    .click(Selector('#submit-button'));
});
```





wrong.js



```
fixture('Login page')
  .page('http://testcafe/login');

test('Login user', async t => {
  await t
    .typeText('.name', 'User name')
    .typeText('.password', 'password')
    .click('#login');
});

fixture('Documentation')
  .page('http://testcafe/docs');

test('Getting started', async t => {
  await t.click('h1');
});
```





login-page.js



```
fixture('Login page')  
  .page('http://testcafe/login');  
test('Login user', async t => {  
  ...  
})
```



documentation.js



```
fixture('Documentation')  
  .page('http://testcafe/docs');  
test('Getting started', async t => {  
  ...  
})
```





wrong.js



```
fixture('Landing page');
```

```
test('Logging', async t => {});
```

```
...
```

```
test('Subscription', async t => {});
```

```
...
```

```
test('Main menu', async t => {});
```

```
...
```





logging.js



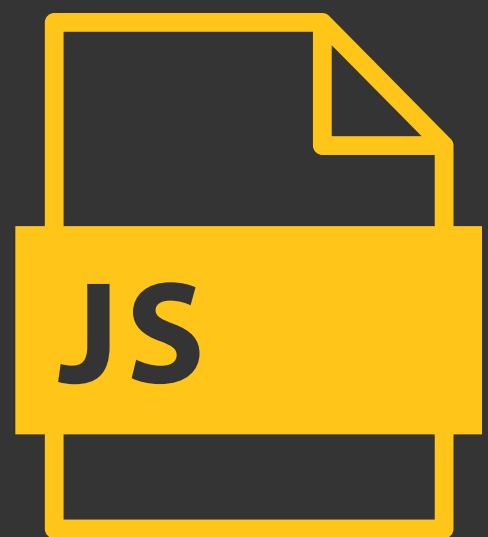
```
fixture('Logging functionality');  
test(...)  
test(...)
```



main-menu.js



```
fixture('Main menu functionality');  
test(...)  
test(...)
```



subscription.js



```
fixture('Subscription functionality');  
test(...)  
test(...)
```



```
import { setDataBase, resetDataBase }  
  from './helpers';  
  
fixture('Setting environment')  
  .before(async () => {  
    await setDataBase();  
  })  
  .after(async () => {  
    await resetDataBase();  
  });
```



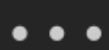
```
import { login, logout } from './helpers';

fixture('Setting environment')
  .beforeEach(async () => {
    await login('user-name', 'password');
  })
  .afterEach(async () => {
    await logout();
  });
```





EXPLORER



> OPEN EDITORS



✓ PROJECT

> src

✓ tests

> data

> helpers

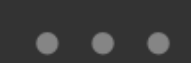
> page\_model

✓ test\_group1

JS test.js

✓ test\_group2

JS test.js



> OUTLINE

> NPM SCRIPTS



# Короткие тесты



# Селекторы

```
import { Selector } from 'testcafe';
```

```
const myButton = Selector('.my-button');
```

```
// CSS  
Selector( 'div.my-button:nth(2)' );
```

```
// TestCafe  
Selector( 'div' ).filter( '.my-button' ).nth(1);
```

```
Selector('label').nth(4).parent(3).child('ul');
```



Поиск по DOM-иерархии

```
Selector('label').withText('Login:')
```



Поиск по тексту

```
const element = Selector('#element-id');  
// → Promise
```

```
await elementSnapshot = await element;  
// → DOM Node State
```

// привязаны к разметке

```
Selector('body').find('div')  
    .nth(5).find('p').nth(3)
```



//слишком общие

```
Selector('div > button')
```



// изменяющиеся параметры

```
Selector(  
  '[style*="background-color: red"]'  
)
```



// читабельные

```
Selector('div')  
  .find('pre')  
  .nextSibling(-1)
```



```
Selector('#topContainer')  
  .find('.child')  
  .withText('Add item')
```



// с точки зрения пользователя

```
Selector('form')  
  .find('[name="btn-foo-123"]')
```



```
Selector('button')  
  .withText('Submit')
```



// кастомные атрибуты

```
Selector(' [data-testid="populate"] ')
```



```
import { Selector } from 'testcafe';

class PageModel {
  constructor() {
    this.nameInput = Selector('#developer-name');
    this.submitButton = Selector('#submit-button');
  }
}

export const pageModel = new PageModel();
```

```
import { pageModel } from './page-model';

test('A test with PageModel', async t => {
  await t
    .typeText(pageModel.nameInput, 'Peter Parker');
    .click(pageModel.submitButton);
});
```

# Селекторы

**React**

**Vue.js**

**Angular**

```
import { ReactSelector } from
'testcafe-react-selectors';

const el = ReactSelector('LoginBtn');
```

# **Автоматическое ожидание**

```
await t.click(selector('#my-button'));
```

```
import { Selector } from 'testcafe';

fixture `Auto waiting for an element`
  .page `https://js.devexpress.com/Demos/WidgetsGallery/Demo/
LoadPanel/Overview/jQuery/Light/`;

test('Loading data test', async t => {
  await t
    .switchToIframe('#demoFrame')
    .click(Selector('span').withText('Load Data'))
    .click(Selector('b').withText('jheart@dx-email.com'));
});
```

JS wait-for-an-element.js X

sqa-days > JS wait-for-an-element.js > ...

```
1 import { Selector } from 'testcafe';
2
3 fixture `Auto waiting for an element`
4   .page `https://js.devexpress.com/Demos/WidgetsGallery/Demo/LoadPanel/Overview/jQuery/Light/`;
5
6 test('Loading data test', async t => {
7   await t
8     .switchToIframe('#demoFrame')
9     .click(Selector('span').withText('Load Data'))
10    .click(Selector('b').withText('jheart@dx-email.com'));
11 });
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

2: zsh + [ ] [ ] [ ] [ ] [ ]

→ sqa-days █



# Автоматические ожидания в TestCafe

- Появление элемента
- Видимость элемента

# Автоматические ожидания в TestCafe

- Появление элемента
- Видимость элемента
- Загрузка старницы
- Выполнение XHR запросов
- Загрузка iframes
- Перевычисление ассершенов

```
// Константы
await t
    .expect('actual').toEqual('expected');

// Promise
await t
    .expect(selector('input').value).toEqual('expected');
```

```
await t
```

```
  .expect(element.tagName).toEqual('input')  
  .expect(element.id).toEqual('element-id')  
  .expect(element.value).toEqual('foo')  
  .expect(element.getAttribute('data-test')).toEqual('test1')  
  
  .expect(element.exists).ok()  
  .expect(element.visible).ok(a)  
  .expect(element.count).toEqual(1)  
  .expect(element.hasClass('bar')).ok();
```

```
import { Selector } from 'testcafe';

fixture `Auto recalculation for an assertion`
  .page `https://js.devexpress.com/Demos/WidgetsGallery/
    Demo/ProgressBar/Overview/jQuery/Light/`;

test('Start progress indicator test', async t => {
  await t
    .switchToIframe('#demoFrame')
    .click(Selector('span').withText('Start progress'))
    .expect(Selector('.dx-progressbar-status').textContent)
      .eql("Loading: 100%", {timeout: 15000});
});
```

JS wait-for-an-assertion.js X

sqa-days > JS wait-for-an-assertion.js > ...

```
1 import { Selector } from 'testcafe';
2
3 fixture `Auto recalculation for an assertion`
4   .page `https://js.devexpress.com/Demos/WidgetsGallery/Demo/ProgressBar/Overview/jQuery/Light/`;
5
6 test('Start progress indicator test', async t => {
7   await t
8     .switchToIframe('#demoFrame')
9     .click(Selector('span').withText('Start progress'))
10    .expect(Selector('.dx-progressbar-status').textContent).eql('Loading: 100%', { timeout: 15000 });
11 });
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

2: zsh

→ sqa-days



```
const value = await Selector('input').value;  
await t.expect(value).eq('temp');
```



```
await t
    .expect(Selector('input').value).eq('temp');
```



## Установка таймаута ожидания элемента

```
// в экшене  
await t.click(Selector('#elementId', { timeout: 500 }));  
  
// CLI аргумент  
testcafe chrome test.js --selector-timeout 500
```



# **Продвинутая эмуляция пользовательских действий**

# Поддержка сложных сценариев

- iframes
- Cross-domain
- Uploading
- Downloading
- Multi-windows

```
//Uploading файлов
```

```
await t
```

```
.setFilesToUpload( '#upload-input', 'test.png' )
```

```
// Отправка файлов на сервер
```

```
await t
```

```
    .setFilesToUpload('#upload-input', 'test.png')  
    .click('#upload-button');
```



JS upload.js ×

sqa-days &gt; JS upload.js &gt; ...

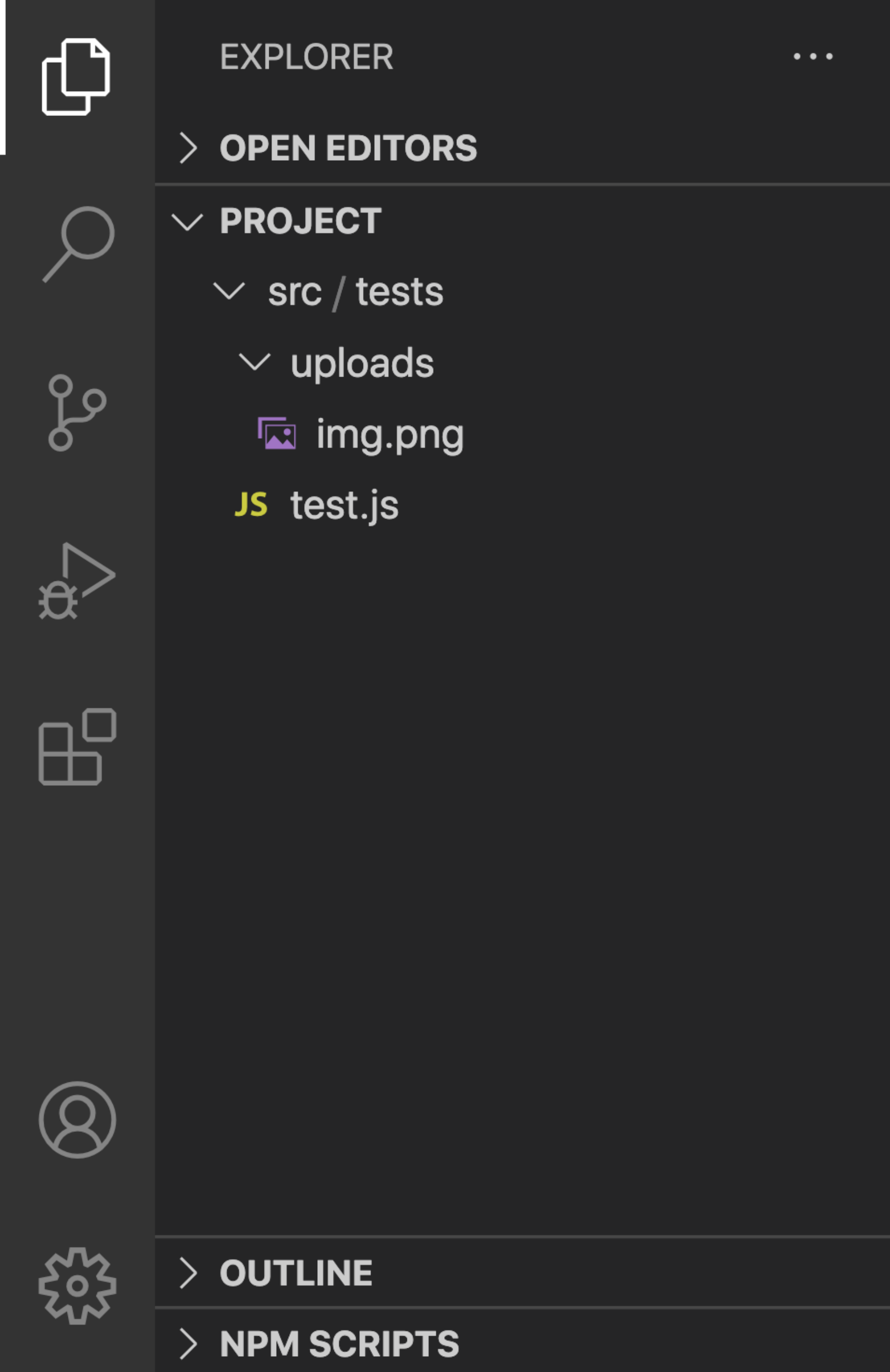
```
1 import { Selector } from 'testcafe';
2
3 fixture('Upload')
4   .page('https://demos.devexpress.com/ASPFileManagerAndUploadDemos/UploadControl/UploadAndSubmit.aspx');
5
6 test('Should upload a new file', async t => {
7   await t
8     //Задаем описание загружаемого файла для демо
9     .typeText('#DemoArea input[type="text"]', 'New File')
10
11     //Загружаем файл new_file.txt
12     .setFilesToUpload(Selector('input[type="file"]').nth(1), './text.txt')
13
14     //Сабмитим форму
15     .click(Selector('span').withText('SUBMIT'))
16
17     //Проверяем что файл был загружен
18     .expect(Selector('#DemoArea td').withText('text.txt').exists).ok();
19 });
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

2: zsh

→ sqa-days █

I



EXPLORER

> OPEN EDITORS

✓ PROJECT

- ✓ src / tests
  - ✓ uploads
    - img.png
    - JS** test.js

> OUTLINE

> NPM SCRIPTS



```
await t
  .setFilesToUpload('#upload-input', [
    './uploads/1.jpg',
    './uploads/2.jpg',
    './uploads/3.jpg'
  ])
```



```
//start downloading process  
await t.click('a');  
  
//check downloaded files  
const isDownloaded = await waitForFileDownload(downloadedFilePath);  
await t.expect(isDownloaded).ok();
```

```
//start downloading process  
await t.click('a');  
  
//check downloaded files  
const isDownloaded = await waitForFileDownload(downloadedFilePath);  
await t.expect(isDownloaded).ok();
```

```
import { t } from 'testcafe';
import fs from 'fs';
import path from 'path';
import os from 'os';

async function waitForFileDownload (path) {
  for (let i = 0; i < 10; i++) {
    if (fs.existsSync(path))
      return true;

    await t.wait(500);
  }

  return false;
}
```

# Слишком большие файлы в тестах



```
fixture `Uploading and Download tests`  
  .page('./index.html')  
  .afterEach(async t => {  
    //clean each uploaded/downloaded file  
  })  
  .after(async () => {  
    //clean all files after all tests  
  });
```



# Тестирование iframes

- Сторонние компоненты
- Сторонние сервисы
- Виджеты платежных систем
- WYSIWYG редакторы

```
await t
    .switchToIframe('#iframeId');
```

```
await t
  .switchToIframe('#iframeId')
  .switchToIframe('#childIframeId');
```

```
await t  
    .switchToMainWindow();
```

# Тестирование функционала в сторонних iframes

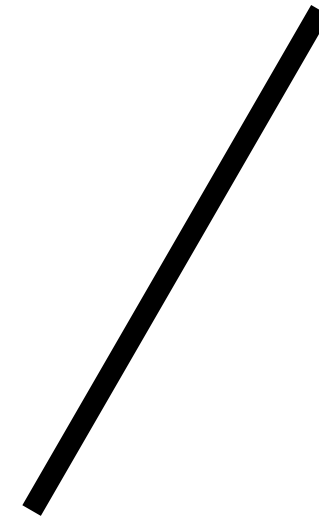


```
await t
  .switchToIframe( '#cross-domain' )
  .click( 'button' )
```



# **Выполнение кода в браузере и клиентские функции**

**Серверный процесс**



**Клиентский процесс**

## Серверный процесс

Тест исполняется на сервере;

## Клиентский процесс

Экшены, ассершены и селекторы элементов выполняются в браузере.

```
import { ClientFunction } from 'testcafe';  
  
const getClientData = ClientFunction(() => {  
    return 'some data';  
});
```

```
const getUrl = ClientFunction(() => {  
    return window.location.href;  
});
```

```
const table = Selector('#customers');

const getFnText = ClientFunction((row, col) => {
  return getTable().rows[row].cells[col].innerText;
},
  { dependencies: { getTable: table } }
);

await t.expect(getFnText(1, 1)).eq('Maria Anders');
```

```
const table = Selector('#customers');

const getFnText = ClientFunction((row, col) => {
  return getTable().rows[row].cells[col].innerText;
},
  { dependencies: { getTable: table } }
);

await t.expect(getFnText(1, 1)).eq('Maria Anders');
```

```
const table = Selector('#customers');

const getFnText = ClientFunction((row, col) => {
  return getTable().rows[row].cells[col].innerText;
},
  { dependencies: { getTable: table } }
);

await t.expect(getFnText(1, 1)).eq('Maria Anders');
```

```
const getSalesAmount = ClientFunction(() => {
  const grid      = document.querySelector('.dx-datagrid-rowsview');
  const rowCount  = grid.querySelectorAll('.dx-data-row').length;
  const sales     = grid.querySelectorAll('td:nth-child(3)');
  const customers = grid.querySelectorAll('td:nth-child(7)');

  const array = [];

  for (let i = 0; i < rowCount; i++) {
    array.push({
      sales: sales[i].textContent,
      customer: customers[i].textContent
    });
  }

  return array;
});
```

```
await t
  .expect(getSalesAmount()).toEqual([
    { sales: '$6,370', customer: 'Renewable Supplies' },
    { sales: '$4,530', customer: 'Apollo Inc' },
    { sales: '$1,110', customer: 'Johnson & Assoc' },
    { sales: '$6,600', customer: 'Global Services' },
    { sales: '$2,830', customer: 'Health Plus Inc' },
    { sales: '$6,770', customer: 'Gemini Stores' },
    { sales: '$1,460', customer: 'Discovery Systems' }
  ]);
```

# Инъекция кода в тестируемую страницу.

```
import { ClientFunction } from 'testcafe';

fixture('Check the a11y')
  .page('https://sqadays.com/en/index')
  .clientScripts([
    './node_modules/axe-core/axe.min.js'
  ]);

test("Shouldn't have any a11y issues", async t => {
  const getViolation = ClientFunction(() => axe.run());

  const { violations } = await getViolation();
  const axeErrors      = violations.map(violation => violation.description);

  console.log('Errors:', axeErrors);

  await t.expect(axeErrors.length).eq(0);
});
```

JS axe.js

```

sqa-days > JS axe.js > ...
1  import { ClientFunction } from 'testcafe';
2
3  fixture('Check the a11y')
4    .page('https://sqadays.com/en/index')
5    .clientScripts([
6      './node_modules/axe-core/axe.min.js'
7    ]);
8
9
10 test("Shouldn't have any a11y issues", async t => {
11   const getViolation = ClientFunction(() => axe.run());
12
13   const { violations } = await getViolation();
14   const axeErrors      = violations.map(violation => violation.description);
15
16   console.log('Errors:', axeErrors);
17
18   await t.expect(axeErrors.length).eql(0);
19 });

```



OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 2: zsh

→ sqa-days



```
const getUrl      = ClientFunction(() => window.location.href);  
const currentLocation = await getUrl();  
await t.expect(currentLocation).eql('http://example.com');
```



```
const getUrl = ClientFunction(() => window.location.href);  
await t  
  .expect(getUrl()).eq('http://example.com');
```



```
class FormPage {
    constructor() {
        // ...
    }

    clearForm() {
        return ClientFunction(() => {
            document.getElementById('main-form').reset();
        });
    }
}
```



```
import { formPage } from './page-model';  
await formPage.clearForm();
```



```
const checkAnimationDone = ClientFunction(() => {  
    //check something on the page  
});  
  
//wait for animation complete  
await t  
    .expect(checkAnimationDone())  
    .ok({ timeout: 50000 });
```



# **Тестирование аторизации**

## **User Roles**

```
import { Role } from 'testcafe';

const userRole = Role('http://example.com/login', async t => {
  await t
    .typeText(loginInput, 'login')
    .typeText(passwordInput, 'password')
    .click(loginButton);
});
```

```
import { Role } from 'testcafe';

const userRole = Role('http://example.com/login', async t => {
  await t
    .typeText(loginInput, 'login')
    .typeText(passwordInput, 'password')
    .click(loginButton);
});
```

```
const userRole = Role(...);  
await t.useRole(userRole);
```

# Автоматическая авторизация

- Browser cookies
- Storages

```
test('Logged user can buy a product', async t => {
  await t.useRole(userRole);

  await t
    .click(pageHeader.myAccountIcon)
    .click(pageHeader.downloadsItem)

    .expect(myDownloadsPage.licenseList.visible).ok()
    .click(myDownloadsPage.newLicenseLink);
});

test('Logged user can manage their account', async t => {
  await t.useRole(userRole);

  await t
    .click(pageHeader.myAccountIcon)
    .click(pageHeader.manageProfileItem)

    .expect(manageAccountPage.accountInfo.visible).ok()
    .click(manageAccountPage.personalInfoUpdateLink);
});
```

JS roles.js

sqa-days &gt; JS roles.js &gt; ...

```
1 import { Role } from 'testcafe';
2 import { pageHeader, loginPage, manageAccountPage, myDownloadsPage } from './utils/page_model';
3 import credentials from './utils/credentials';
4
5 fixture `Login functionality`
6   .page `https://www.devexpress.com`;
7
8 const userRole = Role('https://www.devexpress.com/MyAccount/LogIn/', async t => {
9   await t
10     .typeText(loginPage.loginInput, credentials.user1.login, { paste: true })
11     .pressKey('tab')
12     .typeText(loginPage.passwordInput, credentials.user1.password, { paste: true })
13     .click(loginPage.termOfUseCheckbox)
14     .click(loginPage.loginButton);
15 });
16
17 test('Logged user can buy a product', async t => {
18   await t.useRole(userRole);
19 }
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

2: zsh

→ sqa-days

# Храните роли в отдельно



```
Role('http://example.com/login', async t => {  
  await t  
    .click('a').withText('Go to login page')  
  
  // ...  
  
  .click('a').withText('Open products');  
});
```



```
test('Role using', async t => {  
  await t.click('a').withText('Go to login page');  
  await t.useRole(userRole);  
  await t.navigateTo('http://example/products');  
  
  //test actions  
});
```



```
const userRole = Role('http://example.com/login',  
  async t => {  
    //login process  
  });
```

```
test('Role using', async t => {  
  await t.useRole(userRole);  
  
  //test actions  
});
```



```
fixture `Login functionality used in a hook`  
  .page `https://www.devexpress.com/`  
  .beforeEach(async t => {  
    await t.useRole(userRole);  
  });
```



# Обработка HTTP-запросов

# Обработка HTTP-запросов

- Логирование
- Подмена

# Логирование HTTP-запросов

- Запрос ресурса
- Изменения форматов ответов сервера

```
import { RequestLogger } from 'testcafe';  
const logger = RequestLogger();
```

```
fixture `Request Logger`  
  .requestHooks(logger);  
  
test  
  .requestHooks(logger)  
  ('Test requests logging', async t => {  
  
    ...  
  
  });
```

# Методы RequestLogger

- count
- contains

```
logger.contains({ request, response }) => {  
    return request.url.indexOf('favicon.ico') > 0 &&  
        request.method === 'get' &&  
        response.statusCode === 200;  
});
```

JS logger.js

sqa-days > JS logger.js > ...

```

1  import { RequestLogger } from 'testcafe';
2
3  const logger = RequestLogger();
4
5  fixture `Request Logger`
6    .page `http://devexpress.github.io/testcafe/example`;
7
8  test
9    .requestHooks(logger)
10   ('Test requests logging', async t => {
11     console.log(logger.requests);
12     console.log('Request count:' + logger.requests.length);
13
14     await t
15       .expect(logger.contains(({ request, response }) => {
16         return request.url.indexOf('favicon.ico') > 0 &&
17           request.method === 'get' &&
18           response.statusCode === 200;
19       })).ok();
20   });

```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

2: zsh + [ ] [ ] [ ] [ ] [ ]

→ sqa-days

I

```
await t
    .expect(await logger.contains(...)).ok();
```



```
const hasRequest = await logger.contains(...);  
await t.expect(hasRequest).ok();
```



```
await t.expect(logger.contains(...)).ok();
```



# Сценарии применения подмены HTTP-запросов

- Подключение к сторонним сервисам;
- Разрозненные форматы ответов с сервера;
- Выполнение тестирования на ранних этапах;
- Долгие вычисления;
- Тестирование ошибок;
- Соккрытие чувствительных данных.

```
import { RequestMock } from 'testcafe';
import weatherData from './weather.json';

const mock = RequestMock()
  .onRequestTo(new RegExp(
    'https://api.openweathermap.org/data/2.5/weather'
  ))
  .respond(weatherData, 200, {
    'Access-Control-Allow-Origin': '*'
  });
```

```
import { RequestMock } from 'testcafe';
import weatherData from './weather.json';

const mock = RequestMock()
  .onRequestTo(new RegExp(
    'https://api.openweathermap.org/data/2.5/weather'
  ))
  .respond(weatherData, 200, {
    'Access-Control-Allow-Origin': '*'
  });
```

```
import { RequestMock } from 'testcafe';
import weatherData from './weather.json';

const mock = RequestMock()
  .onRequestTo(new RegExp(
    'https://api.openweathermap.org/data/2.5/weather'
  ))
  .respond(weatherData, 200, {
    'Access-Control-Allow-Origin': '*'
  });
```

```
import { RequestMock } from 'testcafe';
import weatherData from './weather.json';

const mock = RequestMock()
  .onRequestTo(new RegExp(
    'https://api.openweathermap.org/data/2.5/weather'
  ))
  .respond(weatherData, 200, {
    'Access-Control-Allow-Origin': '*'
  });
```

```
fixture `Request Mock`  
  .requestHooks(mock);  
  
test  
  .requestHooks(mock)  
  ('Test requests mocking', async t => {  
  
    ...  
  
  });
```

JS mock.js

```

sqa-days > JS mock.js > ...
1  import { RequestMock, Selector } from 'testcafe';
2
3  import weatherData from './weather.json';
4
5  const mock = RequestMock()
6    .onRequestTo(new RegExp('https://api.openweathermap.org/data/2.5/weather'))
7    .respond(weatherData, 200, { 'Access-Control-Allow-Origin': '*' });
8
9  fixture `Request Mock`
10   .page('https://rawfour.github.io/weather-app-react/')
11   .requestHooks(mock);
12
13  test('Test request mocking', async t => {
14    const temperatureElement = Selector('h3');
15    const descriptionElement = Selector('h4').nth(1);
16
17    await t
18      .typeText('input[type=text]', 'New York')
19      .pressKey('enter')
20      .expect(temperatureElement.textContent).contains('30')
21      .expect(descriptionElement.textContent).eql('Clear sky')
22      .wait(5000);
23  });

```



OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

1: zsh + [ ] [ ] [ ] [ ] [ ]

→ sqa-days



# Злоупотребление моками



- Эмуляция ошибок
- Обращение к нестабильному ресурсу
- Подмена долгих вычислений



# **Отладка и обработка ошибок**

# Отслеживание ошибок

- JS ошибки на странице
- Uncaught errors and unhandled promise rejections  
в серверном коде

report-examples

1: bash

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

Using locally installed version of TestCafe.  
Running tests in:  
- Chrome 86.0.4240.111 / macOS 10.15.7

JS error handling  
\* Click on the button

1) A JavaScript error occurred on "[file:///Users/afilin/examples/report-examples/8\\_debugging/errors-handling/raise\\_error.html](file:///Users/afilin/examples/report-examples/8_debugging/errors-handling/raise_error.html)".  
Repeat test actions in the browser and check the console for errors.  
If you see this error, it means that the tested website caused it. You can fix it or disable tracking JavaScript errors in TestCafe. To do the latter, enable the "--skip-js-errors" option.  
If this error does not occur, please write a new issue at:  
["https://github.com/DevExpress/testcafe/issues/new?template=bug-report.md"](https://github.com/DevExpress/testcafe/issues/new?template=bug-report.md).

JavaScript error details:  
Error: JS error on the page  
at HTMLDivElement.<anonymous> (file:///Users/afilin/examples/report-examples/8\_debugging/errors-handling/raise\_error.html:32:15)  
at HTMLDivElement.dispatchEvent (http://192.168.0.103:53337/hammerhead.js:7:19465)  
at e.t.\_raiseDispatchEvent (http://192.168.0.103:53337/hammerhead.js:8:21347)  
at e.t.\_dispatchMouseEvent (http://192.168.0.103:53337/hammerhead.js:8:19595)  
at e.t.\_dispatchMouseRelatedEvents (http://192.168.0.103:53337/hammerhead.js:8:18282)  
at l (http://192.168.0.103:53337/hammerhead.js:8:12524)  
at e.t.\_simulateEvent (http://192.168.0.103:53337/hammerhead.js:8:12985)  
at e.t.mousedown (http://192.168.0.103:53337/hammerhead.js:8:22218)  
at http://192.168.0.103:53337/testcafe-automation.js:1:37441  
at a (http://192.168.0.103:53337/hammerhead.js:5:11133)

Browser: Chrome 86.0.4240.111 / macOS 10.15.7

```
1 | fixture `JS error handling`  
2 |   .page `./raise_error.html`;  
3 |  
4 | test('Click on the button', async t => {  
> 5 |   await t.click('#button');  
6 | });  
7 |
```

at <anonymous> (/Users/afilin/examples/report-examples/8\_debugging/errors-handling/js\_errors.js:5:13)  
at <anonymous> (/Users/afilin/examples/report-examples/8\_debugging/errors-handling/js\_errors.js:4:1)

1/1 failed (0s)  
Alexeys-MacBook-Pro:errors-handling afilin\$

main 0 0 0 Go Live

```
testcafe chrome test.js --skip-js-errors
```

```
testcafe chrome test.js -e
```

```
testcafe chrome test.js --skip-uncaught-errors
```

```
testcafe chrome test.js -u
```

```
testcafe chrome test.js --live  
testcafe chrome test.js -L
```

```
live-mode.js — test  
JS live-mode.js X  
sqa-days > JS live-mode.js > test('Editing test in live-mode') callback  
1 import { Selector } from 'testcafe';  
2  
3 fixture('Live mode')  
4   .page('http://devexpress.github.io/testcafe/example/');  
5  
6 const nameInput = Selector('#developer-name');  
7  
8 test('Editing test in live-mode', async t => {  
9   await t  
10    .typeText(nameInput, 'Peter Parker')  
11    .expect(nameInput.value).eql('Peter');  
12 });
```

OUTPUT TERMINAL ... 2: zsh + [ ] ^ x

→ sqa-days █

I



**Ctrl+S** – stops test run

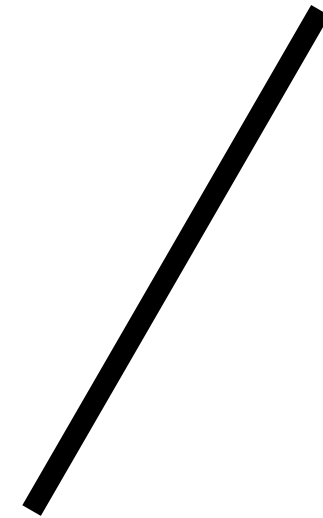
**Ctrl+R** – restarts test run

**Ctrl+W** – on/off the file watcher

**Ctrl+C** – terminates the process



**Серверный debug**



**Клиентский debug**

```
// Клиентская отладка
```

```
await t.debug();
```

# Клиентский debug-режим

The screenshot displays a web browser window with a form. The form contains a text input field labeled "Your name:" and a "Populate" button. The browser's developer tools are open, showing the DOM tree with the "input#populate" element selected. The "Styles" panel shows the "input:matches([type='button'])" rule. The top bar indicates "Debugging test... Which features are important to you?" and "Safari 13.1.2 / macOS 10.15.6".

```
testcafe chrome test.js --debug-mode  
testcafe chrome test.js -d
```

```
testcafe chrome test.js --debug-on-fail
```

# Серверная отладка

- Chrome DevTools
- IDE (VSCode, WebStorm, Sublime text, etc.)

```
test('Server debug', async t => {  
    debugger;  
});
```

```
testcafe --inspect-brk chrome test.js
```

# Debug в Chrome DevTools

The image shows a screenshot of the Chrome DevTools interface. The top window is titled "server\_debug.js — test". The main editor shows the following JavaScript code:

```
1 import { Selector } from 'testcafe';
2
3 fixture `A set of examples that illustrate how to use TestCafe`
4   .page `http://devexpress.github.io/testcafe/example/`;
5
6 test('How to press a specified key (Press Key user action)', async t => {
7   await t.typeText('#developer-name', 'Peter Parker');
8
9   const initialValue = await Selector('#developer-name').value();
10
11   debugger;
12
13   await t
14     .pressKey('home right . delete delete delete delete')
15     .expect(Selector('#developer-name').value).notEql(initialValue)
16     .expect(Selector('#developer-name').value).eql('P. Parker');
17 });
```

The DevTools interface includes a top navigation bar with tabs for Profiler, Console, Sources, Memory, axe, Adblock Plus, and Hints. The Sources panel shows the file "server\_debug.js? [sm]" with a breakpoint at line 11. The right-hand side of the interface shows the "Debugger paused" status and a Call Stack with the following entries:

- (anonymous) server\_debug.js? [sm]:11
- step asyncToGenerator.js:17
- (anonymous) asyncToGenerator.js:28
- \$\$testcafe\_test\_run\$\$tn0q9TQXr\$\$ VM1948:7
- \$\$testcafe\_test\_run\$\$tn0q9TQXr\$\$ VM1950:7
- Promise.then (async) (anonymous) test-run-tracker.js:26

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "type": "node",
      "protocol": "inspector",
      "request": "launch",
      "name": "Launch test files with TestCafe",
      "program": "${workspaceRoot}/node_modules/testcafe/bin/testcafe.js",
      "args": [
        "chrome",
        "${relativeFile}"
      ],
      "console": "integratedTerminal",
      "cwd": "${workspaceRoot}"
    }
  ]
}
```



```
1 | import { Selector } from 'testcafe';
2 |
3 | fixture('Server debugging')
4 |   .page('http://devexpress.github.io/testcafe/example/');
5 |
6 | test('Test with a debugger statement', async t => {
7 |   await t.typeText('#developer-name', 'Peter Parker');
8 |
9 |   const initialValue = await Selector('#developer-name').value;
10 |
11 |   debugger;
12 |
13 |   await t
14 |     .pressKey('home right . delete delete delete delete')
15 |     .expect(Selector('#developer-name').value).notEq(initialValue)
16 |     .expect(Selector('#developer-name').value).eq('P. Parker');
17 | });
```

**Скриншоты и видео**

```
await t.takeScreenshot();
```

```
await t.takeElementScreenshot('#elementId');
```

```
await t.takeScreenshot({ fullPage: true })
```

```
testcafe chrome test.js -s takeOnFails=true
```

```
testcafe chrome test.js --video reports/screen-captures
```

```
testcafe chrome test.js --video videos --video-options failedOnly=true
```

```
testcafe chrome test.js --video videos --video-options singleFile=true
```

```
testcafe chrome test.js --speed 0.5
```

```
await t.typeText(  
    '#developer-name', 'Peter', { speed: 0.5 }  
);
```

```
await t.setTestSpeed(0.5);
```



```
testcafe firefox,edge tests/**/*-test.js  
-s takeOnFails=true --video reports/videos
```



```
testcafe chrome tests --stop-on-first-fail  
testcafe chrome tests --sf
```



```
testcafe all tests --quarantine-mode  
testcafe all tests -q
```



# **Параллелизация тестов**

```
testcafe chrome,firefox test.js
```

```
testcafe chrome tests/*.js -c 3
```

JS concurrency.js x

sqa-days > JS concurrency.js > test('Add and Remove ToDo Items') callback

```

1  import { Selector } from 'testcafe';
2
3  fixture `Add/Remove ToDo Items`
4    .page `http://todomvc.com/examples/vue/`;
5
6  async function addToDoItems (t) {
7    await t
8      .typeText('.new-todo', 'Buy some milk')
9      .pressKey('enter')
10     .typeText('.new-todo', 'See the doctor')
11     .pressKey('enter')
12  }
13
14  test('Add and Remove ToDo Items', async t => {
15    const todoItem = Selector('ul.todo-list').find('li');
16    const removeButton = Selector('button.destroy');
17
18    await addToDoItems(t);
19  }

```



OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

2: zsh + [ ] ^ x

→ sqa-days

I

- Атомарные тесты
- Учитывайте ресурсы машины



**Запуск на мобильных  
устройствах и облачных  
сервисах**

```
await t.resizeWindow(500, 200);

await t.resizeWindowToFitDevice('iPhone 8',
  { portraitOrientation: true }
);

await t.maximizeWindow();
```

```
testcafe "chrome:emulation:device=iphone X" test.js
```

```
testcafe
```

```
"chrome:emulation:width=100;height=200;  
orientation=vertical;" test.js
```

JS mobile-emulation.js x

sqa-days > JS mobile-emulation.js > ...

```

1 fixture('Mobile emulation')
2   .page('https://devexpress.github.io/testcafe/documentation/getting-started/');
3
4 test('Test for a mobile emulation', async t => {
5   await t
6     .click('.cookie-notice .close-icon')
7     .wait(5000);
8 });

```



OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

2: zsh + [ ] [ ] ^ x

→ sqa-days █



```
admin$ testcafe remote:2 test.js
Connecting 2 remote browser(s)...
Navigate to the following URL from each remote browser.
Connect URL: http://192.168.0.103:54707/browser/connect
```

```
admin$ testcafe remote:2 test.js --qr-code
```

```
Connecting 2 remote browser(s)...
```

```
Navigate to the following URL from each remote browser.
```

```
You can either enter the URL or scan the QR-code.
```

```
Connect URL: http://192.168.0.103:54707/browser/connect
```

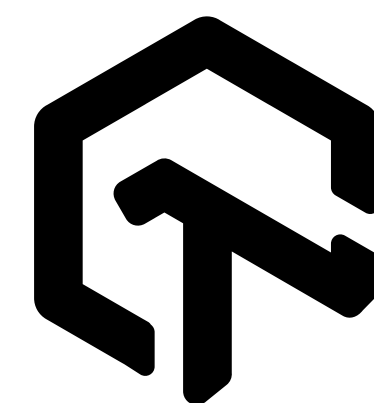




**BrowserStack**



**SAUCE***LABS*



**LAMBDATEST**

```
npm install  
testcafe-browser-provider-browserstack
```

File Edit Selection View Go Run Terminal Help test.js - browser-stack - Visual Studio Code

EXPLORER  
> OPEN EDITORS  
BROWSER-STACK  
JS test.js U

```
JS test.js > ...
1 import { Selector } from 'testcafe';
2
3 fixture('A set of examples that illustrate how to use TestCafe API')
4   .page('http://devexpress.github.io/testcafe/example/');
5
6 test('A simple test on our example page', async t => {
7   const interfaceSelect = Selector('#preferred-interface');
8
9   await t
10     .click(Selector('label').withText('MacOS'))
11     .typeText(Selector('#developer-name'), 'Peter Parker')
12     .click(interfaceSelect)
13     .click(interfaceSelect.find('option').withText('Both'))
14     .click(Selector('#submit-button'))
15     .expect(Selector('#article-header').textContent).eql('Thank you, Peter Parker!');
16 });
17
```

TERMINAL DEBUG CONSOLE PROBLEMS OUTPUT 2: powershell

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS C:\GitHub\report-examples\10\_mobile\_and\_cloud\browser-stack> █

main\*+ Ln 1, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

# Содержание фермы реальных мобильных устройств



- Resize экшены
- Emulation
- Облачные платформы



# Итоги



- Бесплатный, Open Source фреймворк
- Cross browsers
- Cross platform (Win, macOS, Linux)
- Plugin free
- Подходит для любых сайтов
- Подходит для сложных сценариев

## Ссылка на опрос



<https://testcafe.io>

[elena.dikareva@devexpress.com](mailto:elena.dikareva@devexpress.com)

[testcafeteam@devexpress.com](mailto:testcafeteam@devexpress.com)