

# Автоматизация браузера

от вдохновения к практике

# Андрей Гурылев

12 лет как начал верстать

- **Wrike** 
  - Фронтендер
  - Тимлид
  - Скрам-мастер
- **PiterJS**
  - Ведущий
  - Менеджер



# Автоматизация браузера

- **Предыстория**
- **Простота вдохновляет**
- **Моя история**
- **Готовые утилиты**
- **Тесты и мониторинг**

# Автоматизация браузера

- **Предыстория**
- **Простота вдохновляет**
- **Моя история**
- **Готовые утилиты**
- **Тесты и мониторинг**

2009-2012



# CAR

## Система Активной Рекламы

Разместить баннер 168x60

Система активной рекламы

Главная Регистрация Новости Правила Статистика Помощь Бонусы Конкурсы

### Авторизация

Логин:   
(или WMID)

Пароль:

[Войти](#)

[Регистрация](#)

[Напомнить пароль](#)

WMLogin

Пользователей на проекте:  
**1 418 650**

Пользователей за сегодня:  
**295**

Активных за 24 часа:  
**4 306**

Выплат на сумму (руб.):  
**44 932 068.69**

[Подробнее статистика](#)

### СИСТЕМА АКТИВНОЙ РЕКЛАМЫ, РАСКРУТКА И ЗАРАБОТОК

Добро пожаловать в систему активной рекламы. Система включает широкий спектр услуг как для рекламодателя, нуждающегося в продвижении своего сайта, так и для пользователя, желающего подзаработать.

#### Рекламодавателям

- система раскрутки и продвижения сайтов, за счет повышения их посещаемости. Наша система обеспечит постоянный приток уникальных посетителей на Ваш ресурс, это будут реальные люди, которых может заинтересовать содержание и услуги, предлагаемые на Вашем сайте. Также это будет способствовать увеличению позиций Вашего сайта в различных рейтингах.

- Стоимость **1000** уникальных посещений Вашего сайта составляет от **15** рублей.
- Автоматическое пополнение счета через популярные платежные системы **WebMoney, Яндекс.Деньги, QIWI, PayPal**.
- Возможность размещения заданий любой сложности (переходы по сайту, клики, регистрации, голосования, лайки и т.п.).
- Гибкая настройка параметров таргетинга показа рекламируемого сайта (есть возможность выбрать страны для показа сайта, настроить время, дни, длительность показов, установить ограничения на максимальное количество показов в сутки, в час и др.).
- Скрытие информации о том, что посетители перешли на Ваш сайт из нашей системы, за счет замены поля referer.
- Система накопительных **скидок** на рекламу до **30%** (скидка действует только на размещение баннеров и ссылок на страницах нашего сайта).

#### Пользователям

- система активной рекламы, предоставляющая различные возможности по заработку. Получайте доход за просмотр сайтов



# Cerberos 2009-2012

Пассивный  
Стабильный  
Доход

## Что делает цербер?

Выполняет задания с сайта, суть которых такова:

1. Сменить ip, очистить cookie
2. Перейти по ссылке или найти фразу в заданном поисковике
3. Найти в поисковой выдаче сайт и перейти на него
4. Сделать N переходов по ссылкам сайта
5. Перейти по рекламе
6. На открывшемся сайте сделать N переходов по ссылкам
7. Прислать все ссылки (и ip по желанию) в отчёт
8. Быть на страницах сайта между переохдами некоторое время

Здесь максимально кратко изложены вопросы, которые прежде задавались пользователями цербера. Прежде, чем что-то спрашивать, прочитайте этот текст.

## Содержание

- [Смысл программы](#)
- [Правила](#)
- [Что делает цербер?](#)
- [Системные требования](#)
- [Первый запуск](#)
- [Ежедневный запуск](#)
- [Смена ip](#)
- [Статический IP](#)
- [Сутки это не 24 часа](#)
- [Репутация](#)
- [Сколько денег?](#)
- [Как автоматизировать работу?](#)
- [Сколько времени работает?](#)
- [Это всё легально?](#)
- [Хочу много денег и сразу](#)
- [Где можно ошибиться?](#)
- [В логе много ошибок](#)
- [Автор программы - Робин-гуд?](#)
- [Много у кого есть цербер?](#)
- [Я хочу много денег, а получаю мало](#)
- [Не всем стоит ставить цербера на авто...](#)
- [Я подключаю интернет вручную, как ав...](#)
- [Как избавиться от звуков?](#)
- [Ограниченный интернет. Сколько траф...](#)
- [Сколько всего заданий в базе?](#)
- [Цербер сделал несколько заданий и об...](#)



# Результаты

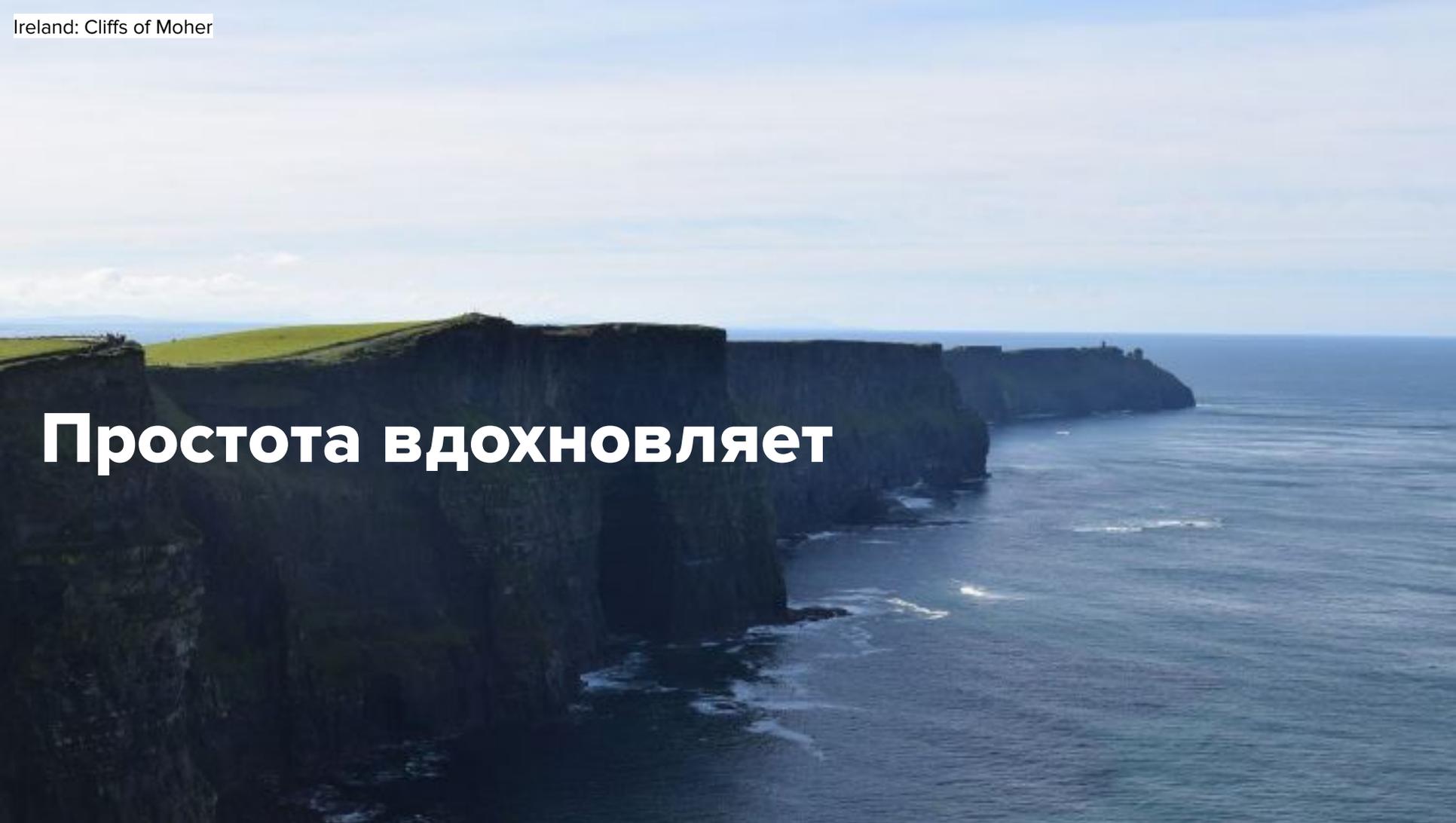
- Получил отличный бизнес-урок
- Потратил три года увлекательных вечеров
- Чутьочку поизучал Delphi и C#
- Ещё больше возненавидел IE
- Научился запускать код, которого не понимаю

**Если вам захочется автоматизировать,  
то мой доклад удался**

# Автоматизация браузера

- Предыстория
- Простота вдохновляет
- Моя история
- Готовые утилиты
- Тесты и мониторинг

**Простота вдохновляет**





```
const browser = await puppeteer.launch({
  headless: false
});
const page = await browser.newPage();
await page.setViewport({
  width: 1240,
  height: 680
});
await page.goto('https://google.com');
await page.screenshot({
  path: 'google.png'
});
await browser.close();
```

# 1. Тостер

[toster.ru/q/588520](https://toster.ru/q/588520)



pfemidi @pfemidi

Тут могло бы быть краткое описание меня



TOSTER.RU

## Можно ли получить список всех возможных тегов toster.ru?

Можно ли получить список всех возможных тегов toster.ru и если можно, то как? Чтобы подписаться на все интересующие. Просматривать все 62 (в настоящее время) страницы с тегами "По вопросам" через браузер в поисках интересных по теме конечно можно, но это та-а-а-ак долго и муторно, в виде какого-нибудь простого файла **.txt** содержащего все возможные теги скопом было бы гораздо лучше.

Вопрос задан 17 дек. 2018 - 87 просмотров



**Nomad\_77** @Nomad\_77 Куратор тегів Toster.ru

Менеджер toster.ru



Такой возможности не предусмотрено. Если круг ваших интересов ограничен определёнными темами - вы можете воспользоваться поиском, если же нет - на странице каждого тега, в правой колонке, находятся теги близкие ему. Там вы можете найти теги по смежным с вашими интересами тематикам.

Ответ написан 17 дек. 2018



knight\_t @knight\_t

```
const puppeteer = require('puppeteer');
const fs = require('fs');
const endOfLine = require('os').EOL;

const pageStart = "https://toster.ru/tags/?page=";
const selector = ".card__head-title a";
const fileName = "tags.txt";
const pages = 62;

(async () => {
  const browser = await puppeteer.launch();

  const promises = [];
  for (let i = 1; i <= pages; i++) {
    promises.push(browser.newPage().then(async page => {
      const url = pageStart + i;
      console.log(url);
    }));
  }
  await Promise.all(promises);
  console.log('done');
});
```

первый раз писал scraper, так что 99.99%, что здесь что-то костыльно, но работает =)

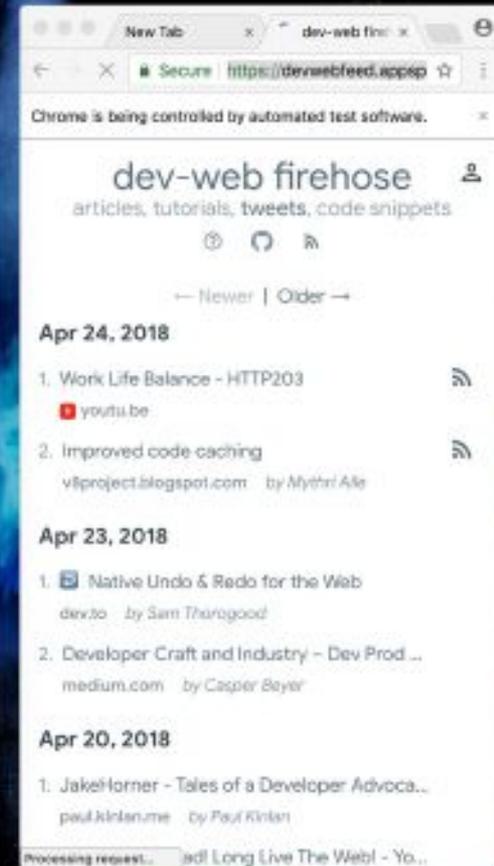
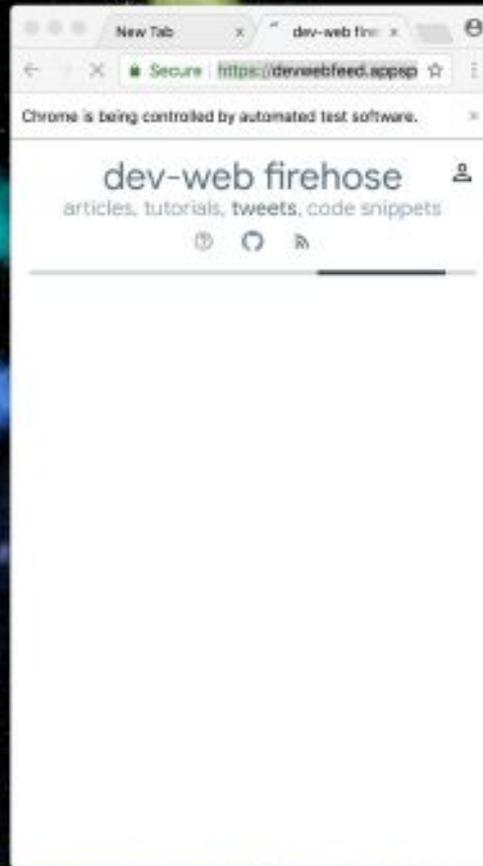
# 2. Скрипты

Eric Bidelman

[github.com/GoogleChromeLabs/puppeteer-examples](https://github.com/GoogleChromeLabs/puppeteer-examples)

# Side by side preload

Загружает две страницы рядом для визуального сравнения скорости загрузки.



[side-by-side-preload.js](https://side-by-side-preload.js)



# Code coverage

Проверяет ленивую стратегию загрузки, смотря покрытие кода CSS / JS при загрузке страницы

[code\\_coverage.js](#)

Event	JS % used	CSS % used	JS used	CSS used	Total bytes used
DOMContentLoaded	35%	55%	6.2KB (21K)	4.7KB (16K)	10.9KB/29.1KB (38K)
load	35%	55%	6.3KB (14K)	19.6KB (44K)	25.9KB/44.2KB (59K)
networkIdle	35%	55%	10.2KB (23K)	19.6KB (44K)	29.8KB/44.2KB (67K)

Event	JS % used	CSS % used	JS used	CSS used	Total bytes used
DOMContentLoaded	no usage found	-	-	-	-
load	-	-	0	33 bytes (0K)	33 bytes/12.6KB (0K)
networkIdle	-	-	0	33 bytes (0K)	33 bytes/12.6KB (0K)

Event	JS % used	CSS % used	JS used	CSS used	Total bytes used
DOMContentLoaded	no usage found	-	-	-	-
load	35%	55%	16.1KB (40K)	0	16.1KB/35.1KB (46K)
networkIdle	35%	55%	16.4KB (47K)	0	16.4KB/35.1KB (47K)

Event	JS % used	CSS % used	JS used	CSS used	Total bytes used
DOMContentLoaded	no usage found	-	-	-	-
load	35%	55%	96.1KB (45K)	0	96.1KB/211.7KB (45K)
networkIdle	35%	55%	108.9KB (51K)	0	108.9KB/211.7KB (51K)

Event	JS % used	CSS % used	JS used	CSS used	Total bytes used
DOMContentLoaded	no usage found	-	-	-	-
load	no usage found	-	-	-	-
networkIdle	35%	55%	6.3KB (35K)	0	6.3KB/17.9KB (35K)

# Verify service worker caching

Проверяет, что все ресурсы страниц закешированы для использования оффлайн при помощи service worker

```
Page: 1st load
Page: 2nd load
Requests made by https://www.chromestatus.com/features (✓ cached by sw, ✗ not cached)
https://www.chromestatus.com/features ✓
https://www.google-analytics.com/collect?v=1&_v=j666a=2897783625&t=timing&... ✗
https://www.chromestatus.com/oraha_data ✓
https://www.chromestatus.com/static/elements/features-imports.vulcanize.js ✓
https://www.chromestatus.com/static/elements/features-imports.vulcanize.htm... ✓
https://www.chromestatus.com/static/img/a2hs_icon_24px.svg ✓
https://www.chromestatus.com/static/img/ring.svg ✓
https://www.chromestatus.com/features_v2.json ✓
https://www.google-analytics.com/analytics.js ✗
https://www.chromestatus.com/static/img/chrome_logo.svg ✓
https://www.google-analytics.com/collect?v=1&_v=j666a=1750961432&t=pageview... ✗
https://www.google-analytics.com/collect?v=1&_v=j666a=1750961432&t=timing&... ✗
https://www.gstatic.com/firebasejs/4.1.3/firebase-app.js ✗
https://www.google-analytics.com/collect?v=1&_v=j666a=1750961432&t=timing&... ✗
https://www.gstatic.com/firebasejs/4.1.3/firebase-messaging.js ✗
```

[verify\\_sw\\_caching.js](https://www.chromestatus.com/features)

# 3. Добываем данные

# Егор Малькевич добывает данные



```
1 import puppeteer from 'puppeteer';
2
3 const scrape = async () => {
4   const browser = await puppeteer.launch({
5     headless: false
6   });
7   const page = await browser.newPage();
8   await page.setViewport({
9     width: 1240,
10    height: 680
11  });
12  await page.goto('https://google.com');
13  await page.screenshot({
14    path: 'google.png'
15  });
16  await browser.close();
17 }
18
19 export default () => scrape();
20
```

Example 3

# Егор Малькевич добывает данные

Доклад на HolyJS Moscow 2017 ([youtu.be/Nn75sRAOIIY](https://youtu.be/Nn75sRAOIIY))

- [Логинится](#)
- [Скачивает файлы](#)
- [Скачивает видео](#)
- [Скачивает потоковые видео](#)

Другие люди в интернетах тоже

- [Скачивание с ютуба](#)
- [Лайкаем фоточки в инсте \(лайки, подписки, отписки\)](#)

# Автоматизация браузера

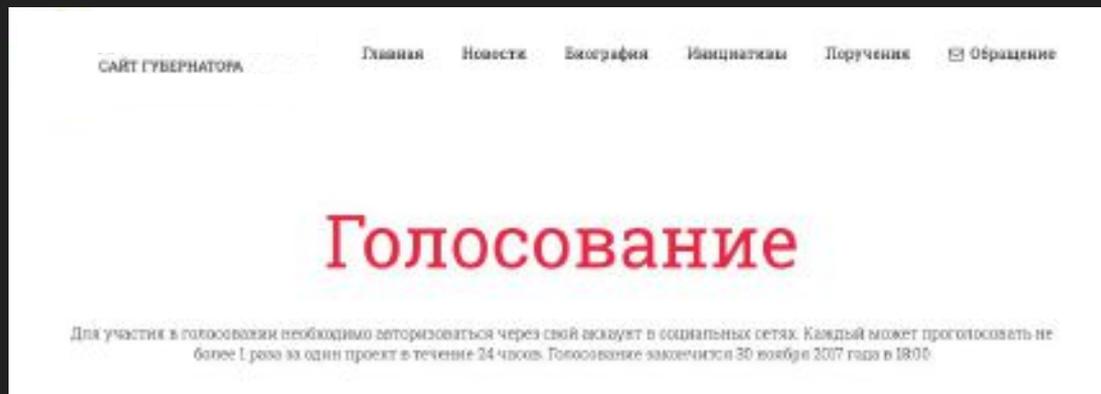
- Предыстория
- Простота вдохновляет
- **Моя история**
- Готовые утилиты
- Тесты и мониторинг

# Plague doctor



# Я хотел помочь другу

И заодно изучить puppeteer



# Призван докладывать о изменениях DOM

**Manager**



**Photographer**



**Reporter**



**Postman**

Manager



Photographer



Reporter



Postman

# new Manager

- `async delegateWork ()`
- `scheduleWork ()`

Manager



Photographer



Reporter



Postman

# new Photographer

- `async makePhoto ()`
- `async savePhoto (page)`

Manager  
↓  
Photographer  
↓  
Reporter  
↓  
Postman

```
const browser = await puppeteer.launch({
  headless: this.options.headless,
  args: ['--no-sandbox']
})
const page = await browser.newPage()
await page.setViewport(this.options.viewport)
await page.goto(this.options.url, {waitUntil: 'load'})

const votes = await page.$eval(`.sv_votes_${this.options.id}`, e => e.innerText)

const time = await this.savePhoto(page)
await browser.close()

return { time, votes }
```

Manager



Photographer



Reporter



Postman

**new Photographer  
extends Designer**

Manager



Photographer



Reporter



Postman

# class Designer

- `async optimizePhoto (path)`
- `async saveThumbnail (now, path)`



# new Reporter

- `async getReport ()`
- `async add (report)`

Manager



Photographer



Reporter



Postman

# new Postman

- async sent (stat)

# Автоматизация браузера

- Предыстория
- Простота вдохновляет
- Моя история
- Готовые утилиты
- Тесты и мониторинг

## Puppeteer extra

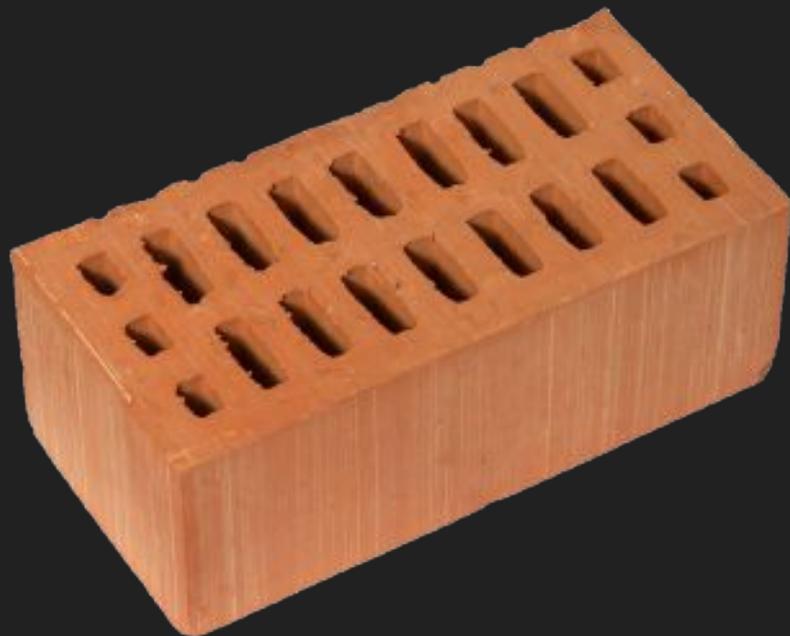
- Стелс-режим
- Анонимизация user-agent
- Обход Google Recaptcha
- Блокировка ресурсов (а-ля Adblock)



# Prerendering SPA

Любой SPA превратит в кирпич HTML

- [puppeteer](#)
- [prerenderer](#)



# Видеозапись сайта

[timecut](#)

[webgif](#)

[puppeteer-lottie](#)



# puppeteer с чужой машины

[puppetron](#) для скриншотов

[puppeteer sandbox](#) для запуска

[browserless](#) для автоматизации

Puppeteer Sandbox 1.19.0 1. basics - Emulate devices

▶ Run

◀ Save

```
1 const puppeteer = require('puppeteer');
2 const devices = require('puppeteer/DeviceDescriptors');
3
4
5 const browser = await puppeteer.launch()
6 const page = await browser.newPage()
7 await page.emulate(devices['iPhone 6'])
8 await page.goto('https://google.com/')
9 await page.screenshot({
10   path: 'full.png',
11   fullPage: true
12 })
13 console.log(await page.title())
14 await browser.close()
```

## Console output

```
DEBUG - Starting job
DEBUG - Compiling en
DEBUG - Running scri
INFO - | 'I just lo
'Interested in imp
'The sudden death
'OnionShare bakes
'California's new
'Study in mice exp
'Firefox Privacy H
'Firefox's New Web
'Banning cars on S
'Twitter says it w
DEBUG - Run finished
DEBUG - Job finished
DEBUG - Uploading to
```

## Screenshots

There's a maximum of three screenshots after 30 hours.



# FACEBOOK BIRTHDAY CLI

[facebook-birthday-cli](#)



```
Rams-MBP:facebook-birthday-cli rampasala$ █
```

# Автоматизация браузера

- Предыстория
- Простота вдохновляет
- Моя история
- Готовые утилиты
- Тесты и мониторинг

# Тестирование скриншотами

- [jest](#)
- [mocha](#)
- [karma](#)
- [istanbul](#)
- [ava](#)

Для storybook: [storybook](#)

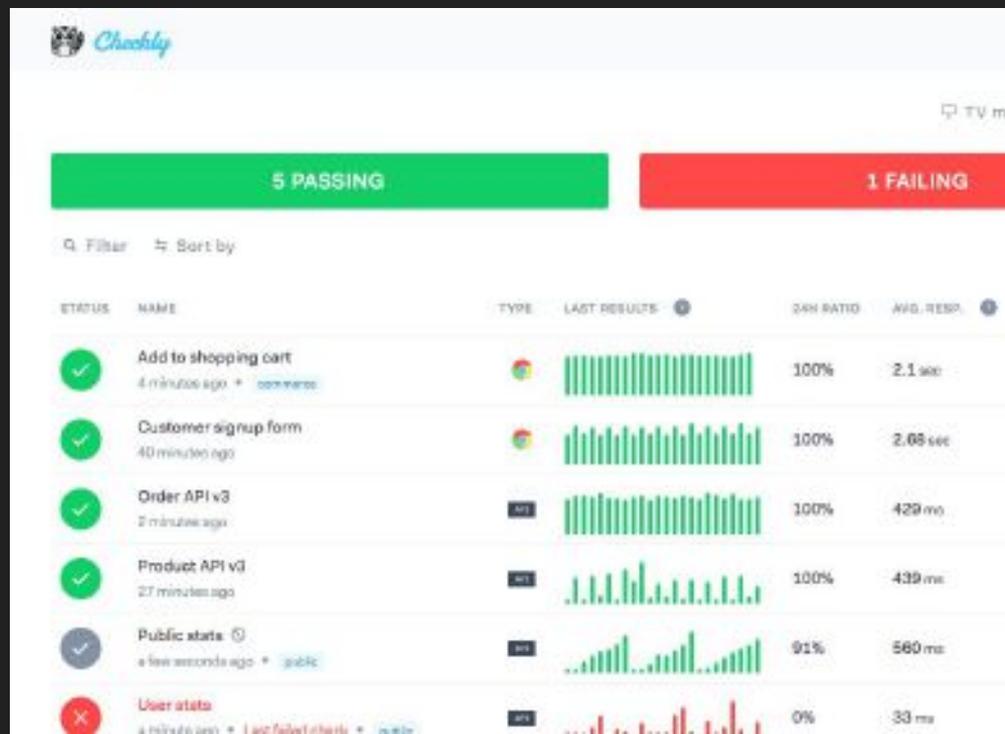


# Checkly

## Мониторинг стабильности



# Checkly



# Garie

## Аналитика показателей производительности

<https://www.bbc.co.uk/>

Sep 11, 2018, 7:50 AM GMT+1

Emulated Nexus 5X, Simulated Fast 3G network



Performance



Progressive Web App



Accessibility



Best Practices

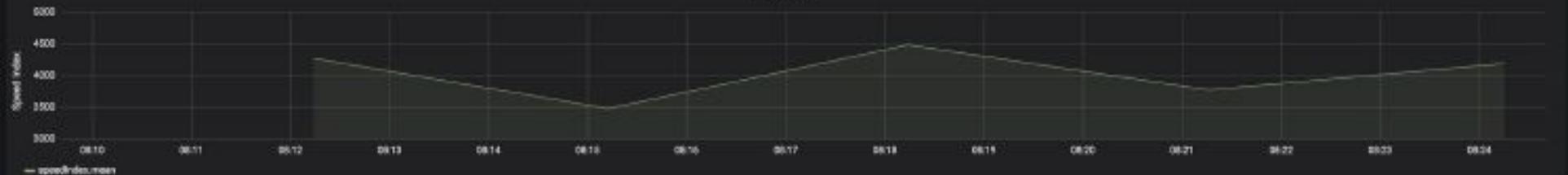


SEO

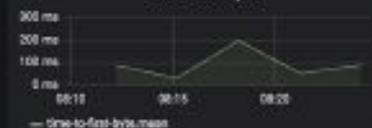
Score scale: 0-44 45-74 75-100

<https://www.bbc.co.uk>

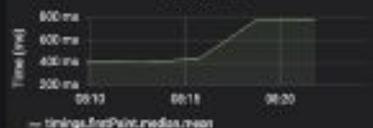
Speed Index



Time to first byte



First Paint



First Contentful Paint



First Meaningful Paint



Time to interactive



PWA Score

58

Accessibility Score

87

Best Practices Score

87

SEO Score

100

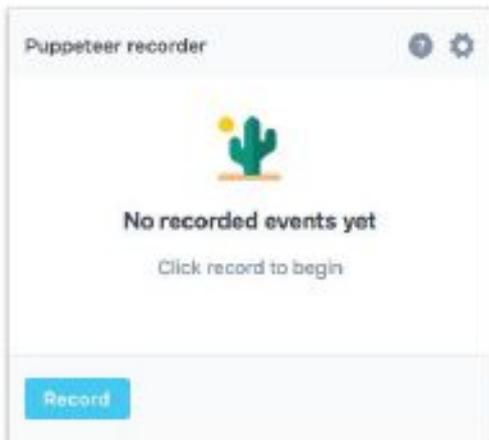
Performance Score

83

# Puppeteer-recorder

[Chrome extension](#)

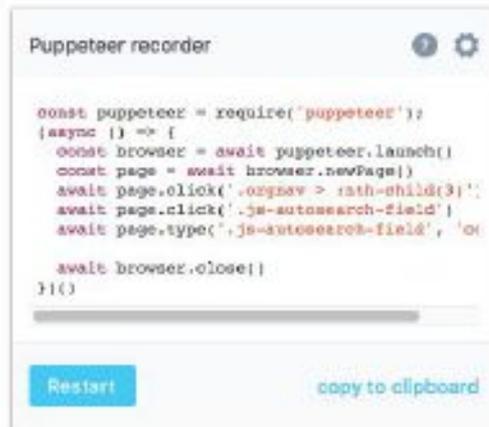
## 1. click record



## 2. track events



## 3. generate Puppeteer code



# Puppetry

[puppetry](#)

The screenshot displays the Puppetry web application interface. On the left is a dark blue sidebar with navigation options: File, Snippets, Settings..., Run... FT, Projects, ACME form, and **rwil.json**. The main content area is titled 'Project: ACME form' and shows a 'Targets' tab. Below the tab, there is a brief explanation: 'Targets are identifiers associated with locators (CSS selector or XPath) that we can refer in the test cases.' followed by a 'LEARN MORE +' link. A table lists three targets:

Target	Selector	Action
MENU_MOBILE_CONTAINER	Amobile-menu	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">↕</a>
MENU_DESKTOP_CONTAINER	Amain-row	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">↕</a>
TEASER	header	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">↕</a>

Below the table, there are two input fields: 'Enter target name' and 'Enter selector', with an 'Add' button to the right. At the bottom of the main area is a button labeled 'Edit as CSV'. The footer of the application includes the text '© 2019 MIT License. Created by Dmitry Shalko' and social media icons for GitHub, Twitter, and YouTube.

# Итого

- puppeteer - это просто
- им можно заменить любую рутину
- в том числе повторяющиеся ручные тесты
- в том числе мониторинг и анализ проблем
- с ним можно заработать

**У вас возникла мысль в духе  
«было бы классно автоматизировать...»?**



# Спасибо за внимание

Слайды: [vk.cc/9UDhSe](https://vk.cc/9UDhSe)

Андрей Гурылев

[t.me/fogrew](https://t.me/fogrew)

Вопросы?

