



SOFTWARE
упражнения

Осигуряване на качество на софтуера (Q.A.)

гл. ас. д-р Георги Шарков

гл. ас. д-р Мая Стоева





Здравейте!

гл. ас. д-р Георги Шарков

gesha@esicenter.bg

гл. ас. д-р Мая Стоева

maya@fmi-plovdiv.org

Съдържание

1. Инструменти за тестване на софтуера
2. “Card Sorting” пример
3. Тестване с <https://www.browserstack.com>





1

Инструменти за тестване на софтуера

Инструменти тестване

1. Helium (<https://github.com/mherrmann/selenium-python-helium>)
<https://testingbot.com/support/getting-started/helium.html>

Helium е инструмент за автоматизирано тестване на уебсайтове в браузъри. Той предлага прости команди като `start_chrome`, `кликване` и `запис`. Helium ни предоставя възможност да използваме команди точно, както бихме го направили, когато даваме инструкции на някой, гледащ през рамо към екрана - единствената разлика е, че ги въвеждаме. След като зададем няколко команди, можем да ги запишем във файл, който да бъде възпроизведен с едно натискане на бутон, в определен интервал от време или да се задейства от инструмент за управление на изграждане/тестване.



Инструменти тестване

1. Helium (<http://heliumhq.com/>)

README.md

Selenium-python but lighter: Helium

Selenium-python is great for web automation. Helium makes it easier to use. For example:

```
Python 3.7.3 (default, Dec 20 2019, 18:57:59)
>>> from helium import *
```

Under the hood, Helium forwards each call to Selenium. The difference is that Helium's API is much more high-level. In Selenium, you need to use HTML IDs, XPaths and CSS selectors to identify web page elements. Helium on the other hand lets you refer to elements by user-visible labels. As a result, Helium scripts are typically 30-50% shorter than similar Selenium scripts. What's more, they are easier to read and more stable with respect to changes in the underlying web page.

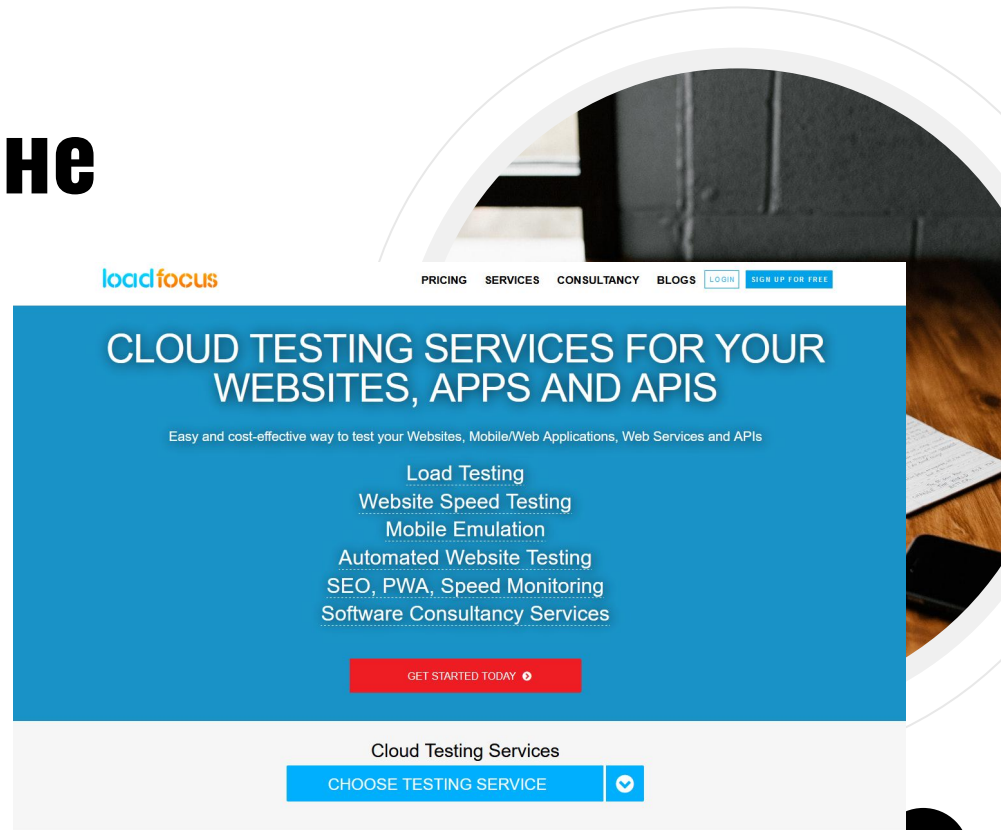
Because Helium is simply a wrapper around Selenium, you can freely mix the two libraries. For example:

```
# A Helium function:
driver = start_chrome()
# A Selenium API:
driver.execute_script("alert('HI!');")
```

Инструменти тестване

2. Loadfocus (<https://loadfocus.com/>)

Това е cloud-базиран инструмент за stress тестване на за сайтове, уеб app-ове и API-та.



Инструменти тестване

3. Ghost Inspector (<https://ghostinspector.com/>)

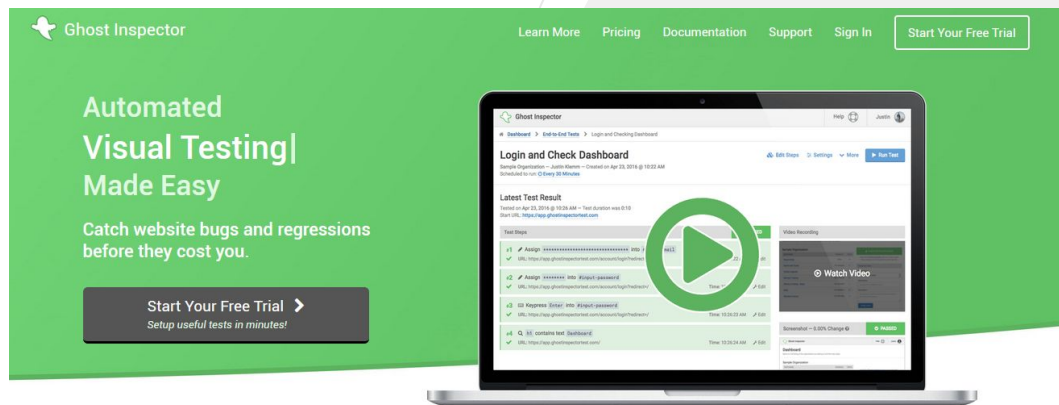
Това е инструмент, специализиран в automated UI testing. Тестерите му дават много добри оценки и са доволни от него. Съществува като extension или addon към Chrome-а и е много удобен и лесен за употреба. Налични са доста примери за работа с него.

Предоставя безплатна версия.



Инструменти тестване

3. Ghost Inspector (<https://ghostinspector.com/>)



✓ Test Recorder & Codeless Editor

✓ Video & Screenshot Comparison

✓ Multiple Browsers & Screen Sizes

✓ API & Third Party Integrations

✓ Advanced Test Scheduling

✓ Parallel Testing by Default

Empower your entire team to start testing.

Improve Efficiency. Deliver Faster. Ensure Quality.

Инструменти тестване

4. iMacros (<http://imacros.net/overview/web-testing/>)

Предоставя regression тестване (напр. проверка на стари функционалности след update на software...), може да тества AJAX уебсайтове, позволява и генериране на някои таблични и графични статистики.



The image shows a screenshot of the iMacros website. The header includes the iMacros logo with a 'Progress' indicator and navigation links: FEATURES, PURCHASE, ABOUT, SUPPORT, DOWNLOAD, and CONTACT. The main banner reads 'Web Testing' and 'Easily automate functional, performance, and regression testing'. Below this, the 'Functional, Performance, and Regression Testing' section explains that iMacros can automate testing across various browsers and technologies. It includes sub-sections for 'Test with IE, Chrome and Firefox', 'Regression Testing', and 'Flash, Javascript & Ajax'. A sidebar on the right contains buttons for 'TRY IMACROS' and 'BUY IMACROS', a 'Features' list, and a 'Questions?' section.

iMacros
Progress

FEATURES PURCHASE ABOUT SUPPORT DOWNLOAD CONTACT

Home » Overview » Web Testing

Web Testing

Easily automate functional, performance, and regression testing

Functional, Performance, and Regression Testing

With iMacros you can quickly and easily automate functional, performance, and regression testing across any website technology including Java, Flash, Flex or Silverlight applets and all AJAX elements. And with iMacros built-in STOPWATCH command you can capture exact web page response times for every step of a process.

Test with IE, Chrome and Firefox

With iMacros you can perform in-browser testing across all major browsers including Internet Explorer, Firefox and Chrome. Test cases created in Internet Explorer work in Firefox and Chrome – and vice versa. And with iMacros you have “multi-threading” – meaning you can run up to 50 separate instances of iMacros at the same time.

Regression Testing

Regression testing with iMacros is easy and very flexible. When parsing a web site you want to find out if a certain word exists on a web page to trigger further action (like saving the web page, printing the page or running the next macro). This is also called keyword assertion or regression testing. If the keyword does not exist, iMacros retries the

Flash, Javascript & Ajax

Recording Click Mode

As we recorded

1. Use one or more HTML attributes to capture HTML tags

2. Use one or more HTML attributes to capture HTML tags

3. Use one or more HTML attributes to capture HTML tags

4. Use one or more HTML attributes to capture HTML tags

5. Use one or more HTML attributes to capture HTML tags

6. Use one or more HTML attributes to capture HTML tags

7. Use one or more HTML attributes to capture HTML tags

8. Use one or more HTML attributes to capture HTML tags

9. Use one or more HTML attributes to capture HTML tags

10. Use one or more HTML attributes to capture HTML tags

Testing & verifying Flash

Load test performance

Load test performance

TRY IMACROS

BUY IMACROS

Features

- Overview
- Browser Automation
- Data Extraction
- Web Testing
- Web Browser API
- iMacros Component
- FAQs

Questions?

Please [contact our sales team](#), we'd be happy to talk with you about iMacros products.

Инструменти тестване

5. Селениум (<https://selenium.dev/>)

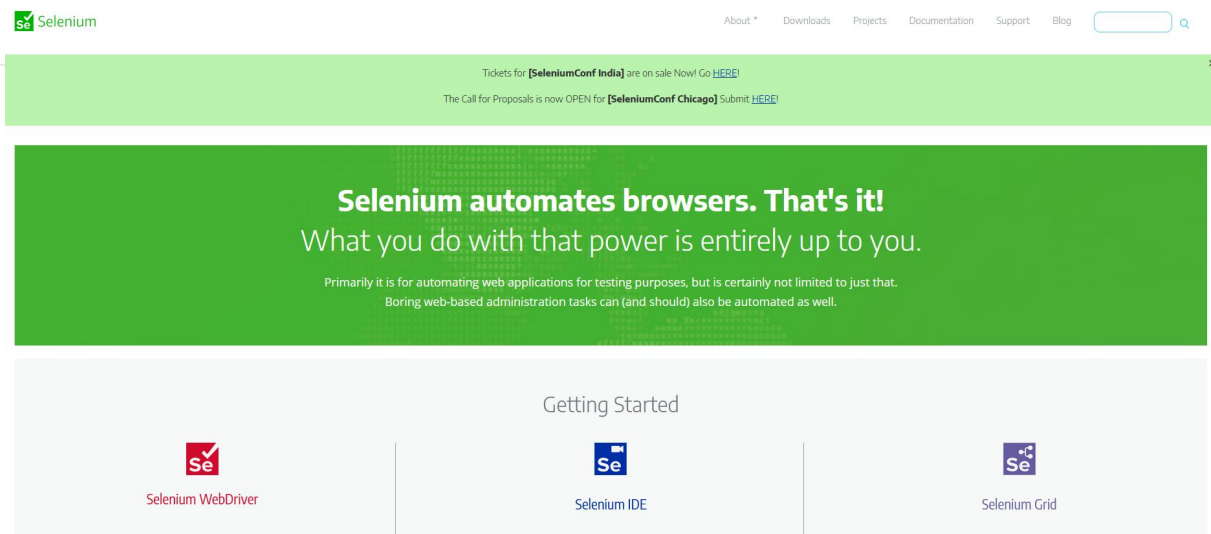
Предоставя група инструменти с възможност за поддръжка на автоматизирани тестове. Последните включват множество функции за тестване на различни видове уеб приложения.

Позволява сравняването на резултата от теста с настоящото поведение на приложението. Selenium има възможност за изпълнение на тестовите върху множество браузър платформи. Притежава собствен специфичен език (Selenese), на който могат да се пишат тестове на програмни езици като Java, C#, Groovy, Perl, PHP, Python и Ruby. Той работи върху Windows, Linux и Macintosh OS. Освен това е open-source софтуер и може да се използва безплатно.



Инструменти тестване

5. Селениум (<https://selenium.dev/>)



The screenshot shows the Selenium website homepage. At the top, there is a navigation bar with links for 'About', 'Downloads', 'Projects', 'Documentation', 'Support', and 'Blog', along with a search bar. Below the navigation bar, a green banner contains text about SeleniumConf India and Chicago. The main content area features a large green box with the text 'Selenium automates browsers. That's it!' and 'What you do with that power is entirely up to you.' Below this, a smaller text block explains that Selenium is primarily for automating web applications for testing purposes. At the bottom, there is a 'Getting Started' section with three columns: 'Selenium WebDriver', 'Selenium IDE', and 'Selenium Grid', each with its respective logo.

Selenium

About * Downloads Projects Documentation Support Blog

Tickets for **SeleniumConf India** are on sale Now! Go [HERE](#)

The Call for Proposals is now OPEN for **SeleniumConf Chicago** Submit [HERE](#)

Selenium automates browsers. That's it!

What you do with that power is entirely up to you.

Primarily it is for automating web applications for testing purposes, but is certainly not limited to just that. Boring web-based administration tasks can (and should) also be automated as well.

Getting Started

Se
Selenium WebDriver

Se
Selenium IDE

Se
Selenium Grid

Инструменти тестване

6. Browserstack (<https://www.browserstack.com/responsive>)

една отлична облачна платформа за паралелно тестване на уеб приложения на десктоп и мобилните устройства, която дава възможност на разработчиците да проверят своите уеб сайтове и мобилни приложения в браузъри, операционни системи и реални мобилни устройства, без да се изисква от потребителите да инсталират или поддържат физически. Той включва четири основни продукта - Live, App Live, Automate и App Automate.



Live

Features

Mobile Features

Browsers & Devices

World's leading cross browser testing tool

Interactive web-based testing on 2000+ browsers and real devices instantly. Say goodbye to your lab of devices and virtual machines.

[Get started free](#)

Test your websites

**Live**

Interactive cross browser testing

**Automate**

Selenium testing at scale

Test your mobile apps

**App Live**

Interactive native & hybrid app testing

**App Automate**

Test automation for native & hybrid mobile apps

Use BrowserStack with your favourite products. See our [Integrations](#) →

✦ For Teams

Enterprise

✦ Tools

Screenshots

[Responsive](#)

Responsive

Responsive Design Testing on Real Devices

Check website responsiveness on multiple iOS & Android devices

 [Invite your team](#) to join BrowserStack and test collaboratively. ✕

<https://mw-medianetworks.eu/>

Check

[Enable Local Testing](#) ▾

[Learn more about local testing](#)



iPhone X
375x812
🍏 11.0



Galaxy Note 10
412x869
🌐 9.0



iPhone 8 Plus
414x736
🍏 12.0



Galaxy S9 Plus
412x740
🌐 8.0



iPhone SE
320x568
🍏 11.0



Pixel 3 XL
412x846
🌐 9.0



iPad Mini 2019
768x1024
🍏 12.0



Galaxy S7
360x640
🌐 6.0



iPad Pro
1024x1366
🍏 12.0



Kindle Fire HDX 7
600x960
🌐 4.3



Galaxy Tab S4
1134x712
🌐 8.1



Windows
1280x1024
7



Mac OSX
1920x1080
🍏 Yosemite



Try Live Interactive Testing on 2000+ real browsers and devices. [Get Started Free](#)

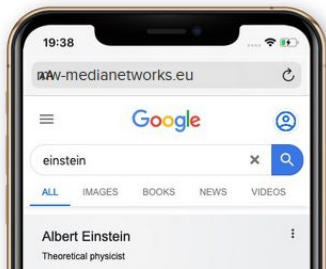


iPhone X

Galaxy Note 10

Viewport: **375x812** | Screen size: **5.8"** | Monitor size: **15.4"**

 **Portrait**  Landscape 





Още няколко инструмента за софтуерното тестване

<http://www.softwareqatest.com/qatweb1.html>



2

Въпросчета...

Имате 100 бѹга за 10 дни, но в даден момент ви казват, че имате 5 дни, какво ще направите?

Верен отговор би бил: ще приоритизират най-важните бѹгове.

Кои test cases бихте автоматизирали – критичните или тези, които се изпълняват най-често?

Верен отговор би бил: Тези, които се изпълняват най-често, за да ни спестят време, а за критичните отделяме време и внимание!

Каква е разликата между Re-testing и Regression testing?

Верен отговор би бил: re-testing – след като се открие бѹг, го поправяме и проверяваме отново, за премахнат. Тук трябва да се повтори същият test case, при който е намерен бѹга.

Regression testing – проверяваме дали след като сме открили и отстранили бѹга, не сме променили друга функционалност. Това най-често са автоматизирани тестове. Те трябва да са документирани и добре описани. Имаме четири нива на тестване: тестване за дефекти, тестване на променени пътища, тест на нова функционалност и накрая – пълно тестване.

Кога е удачна автоматизацията?

Верен отговор би бил: При regression testing.

Кой е най-интересния бѹг, който сте откривали?

Разкажете какво сте открили, какво бихте променили и как бихте установили дали бѹга е отстранен.

Какво точно представя severity (тежест) и priority (приоритет) на дефекта/бъга?

Верен отговор би бил: Severity показва какво е въздействието на нашия бърк върху нашата система.

Какво тестваме при Black Box Testing и White Box Testing?

Верен отговор би бил: При Black Box Testing-а тестваме само интерфейса на софтуера, без познаваме неговата Функционалност, а при White Box Testing проверяваме качеството на кода, без без да ни интересува интерфейса.

Какво съдържа един тест план?

Верен отговор би бил: **test scope** – дефинира какво ще се тества в дадения период и кои точно модули.

test Objectives – дефинициите, използвани тест плана като бързина на сайт, тестовите резултати, които се очаква да бъдат се покрият и други.

assumptions – какво се очаква от тестерите, бюджет за тестване, начална и крайна дата и инструменти;

risk Analysis – дефинират се по-важни неща, които може да провалят тестването;

test design – какви са етапите на тестване, каква е последователността, каква е времевата рамка;

roles & Responsibilities – кой какво прави и за колко време го прави, приоритети върху задачите;

test data management – събират се проекти, които да се обновяват;

test environment – с каква среда разполагаме, с каква машина/сървър разполагаме, къде ще тестваме;


communication approach – кога да има срещи с екипа и т. Н.;

test tools – какви инструменти ще се използват за различните дейности.

Литература

<http://forum.dev.bg/index.php/220/%D0%BA%D0%BE%D1%8F-bug-tracking-software-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0-%D0%B8%D0%B7%D0%BF%D0%BE%D0%BB%D0%B7%D0%B2%D0%B0%D1%82%D0%B5-%D0%B8-%D0%B7%D0%B0%D1%89%D0%BE>

<https://kvasilev.com/sqa/test-plan-za-software/>

A circular frame containing a photograph of a desk. On the desk, there is a potted plant in a white pot with a lace-like top, a large black letter 'A', and a wooden block with the letter 'S'. The background shows a window with light coming through. The text "Благодаря за Вашето внимание!" is overlaid on the bottom right of the circle.

**Благодаря за
Вашето внимание!**