DetectGaYm

**Table of contents**

[Описание](#_topic_Newtopic) 4

[Установка](#_topic_Newtopic1) 4

[Структура программы](#_topic_Newtopic2) 5

[Редактор тестов](#_topic_Newtopic5) 6

[Запуск редактора](#_topic_Newtopic11) 6

[Основное меню](#_topic_Newtopic3) 9

[Панель инструментов](#_topic_Newtopic4) 11

[Общие параметры](#_topic_Newtopic6) 12

[Данные](#_topic_Newtopic7) 13

[Шаги](#_topic_Newtopic8) 14

[Типы действий в шагах](#_topic_Newtopic9) 15

[Выполнение тестов](#_topic_Newtopic10) 18

[Сформировать команду для запуска](#_topic_Newtopic13) 20

[Закрыть заблокированный порт](#_topic_Newtopic14) 22

[Структура теста](#_topic_Newtopic17) 24

[Запуск тестов из командной строки](#_topic_Newtopic12) 26

[Запуск тестов в Jenkins](#_topic_Jenkins) 29

[Обновление веб-драйвера](#_topic__) 33

[Примеры разработки тестов](#_topic_Newtopic15) 34

[Демонстрационный тест](#_topic_Newtopic16) 34

[Базовая проверка событий](#_topic_Newtopic19) 38

[Опциональная проверка событий](#_topic_Newtopic18) 40

[Получение данных из элементов](#_topic_Newtopic21) 42

[Использование условий](#_topic_Newtopic20) 45

[Применение JavaScript](#_topic_JavaScript) 48

**Описание**

Бесплатная программа автоматизированного тестирования событий Google analytics и Yandex metrika.

Программа позволяет проверить события Google analytics и Yandex metrika отправляемые в момент выполнения каких либо действий на странице сайта.

Сценарии тестирования описываются в отдельных файлах в формате json. Для создания и редактирования сценариев предусмотрен удобный редактор.

Программа портативная и кроссплатформенная с поддержкой операционных систем Windows и Linux.

Системные требования:

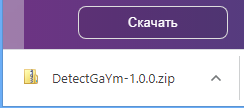
* Windows 7, 8, 10 | Linux (Ubuntu, Mint, и др.)
* [.NET Framework 3.5 SP1](https://www.microsoft.com/ru-ru/download/details.aspx?id=22)
* [Visual C++ Redistributable](https://support.microsoft.com/ru-ru/help/2977003/the-latest-supported-visual-c-downloads)
* [JRE 8 (Java SE Runtime Environment)](https://www.oracle.com/java/technologies/javase-jre8-downloads.html)
* [JDK 8 (Java SE Development Kit)](https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html)

*Created with the Personal Edition of HelpNDoc:* [*Free EBook and documentation generator*](https://www.helpndoc.com)

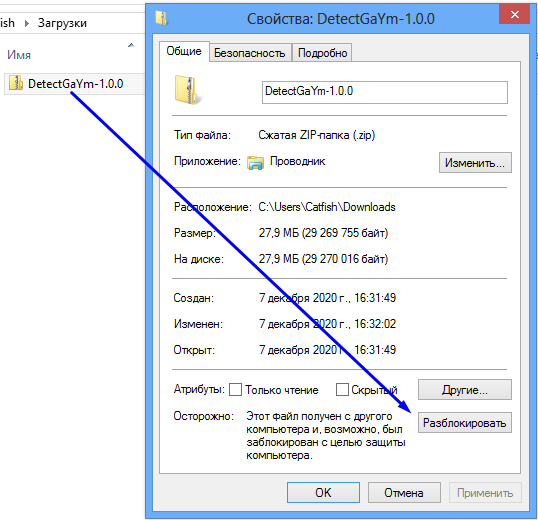
**Установка**

**Установка программы**

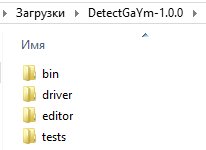
Скачайте архив программы с официального сайта <https://detect-gaym.ucoz.net>



Разблокируйте архив через свойства



Программа портативная и не требует инсталяции, достаточно просто распаковать архив в любое место на жестком диске.



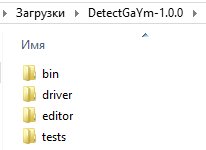
Установка программы завершена.

*Created with the Personal Edition of HelpNDoc:* [*Full-featured Documentation generator*](https://www.helpndoc.com)

**Структура программы**

**Структура программы**

После распаковки архива вы увидите в папке DetectGaYm-1.0.0 следующие папки



**bin** - содержит программу detect-gaym.jar и вспомогательные библиотеки. (файлы run-test нужны для запуска автотестов с помощью редактора тестов)

**driver** - содержит веб-драйвер chromedriver необходимый для взаимодействия программы с браузером. (в папке драйвера для операционных систем Windows и Linux) [ChromeDriver - WebDriver for Chrome](https://chromedriver.chromium.org/)

**editor** - содержить программы визуального редактирования тестов.

tests-editor-windows.exe - редактор тестов для Windows

tests-editor-linux.jar - редактор тестов для Linux

**tests** - папка для хранения тестов, в папке присутствует демонстрационный тест test-search-ga.json

**errors** - папка появится в случае падения теста, в папке будет храниться скриншот экрана сделанный во время ошибки.

*Created with the Personal Edition of HelpNDoc:* [*Create cross-platform Qt Help files*](https://www.helpndoc.com/feature-tour/create-help-files-for-the-qt-help-framework)

**Редактор тестов**

Редактор тестов - это встроенная вспомогательная программа которая призвана упростить процесс разработки сценариев автоматизированного тестирования.

Визуальный редактор тестов версии для Windows позволяет не только легко и просто создавать сценарии но так же быстро запускать их выполнение. Работая в редакторе тестов вам не потребуется запускать сценарий из консоли, достаточно будет нажать одну кнопку и редактор сам сформирует команду в файле run-test.bat в папке bin и запустит выполнение сценария автоматизированного тестирования.

*Created with the Personal Edition of HelpNDoc:* [*Qt Help documentation made easy*](https://www.helpndoc.com/feature-tour/create-help-files-for-the-qt-help-framework)

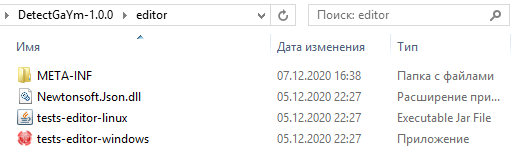
**Запуск редактора**

**Запуск редактора**

В папке editor содержится две версии редактора для систем Windows и Linux

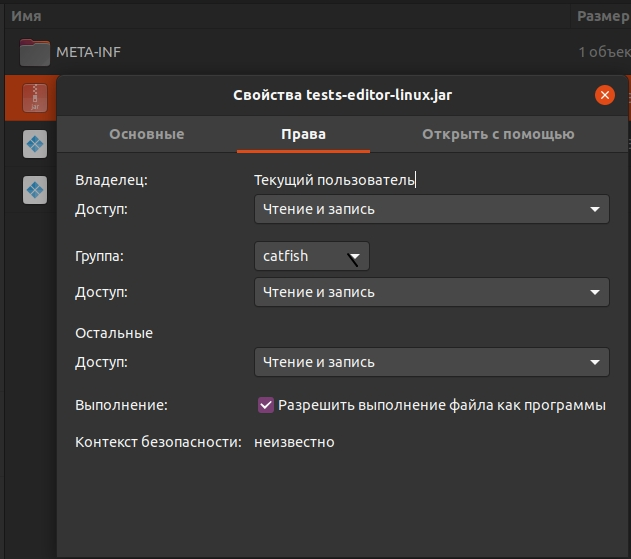
tests-editor-windows.exe - редактор тестов для Windows

tests-editor-linux.jar - редактор тестов для Linux

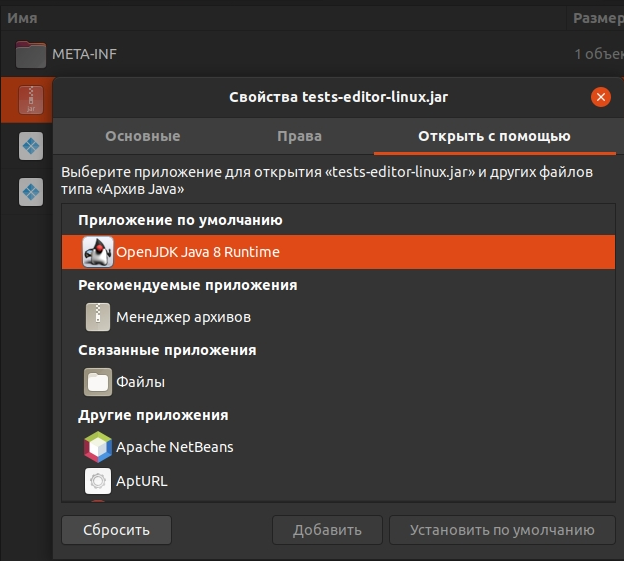


**Особенности запуска редактора в системе Linux**

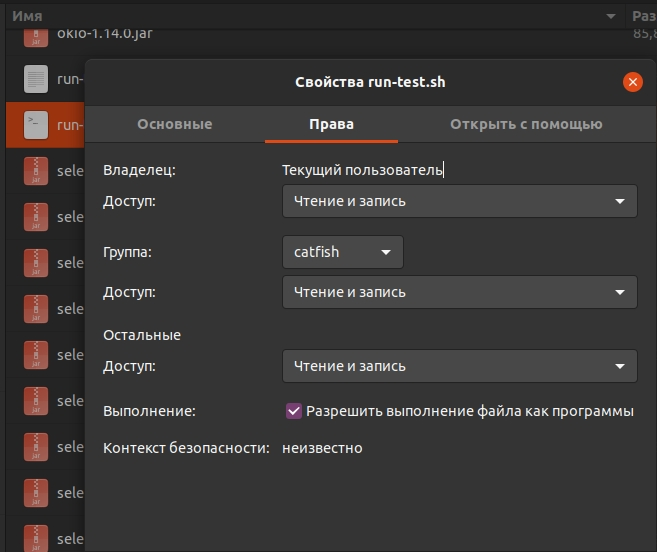
Для запуска редактора, файл tests-editor-linux.jar в папке editor, обязательно нужно включить права на чтение и запись, а так же включить разрешение на выполнение файла как программы, включить эти права можно в свойствах файла на вкладке "Права".



Редактор запускается с помощью OpenJDK Java 8 Runtime который устанавливается с пакетом JDK8



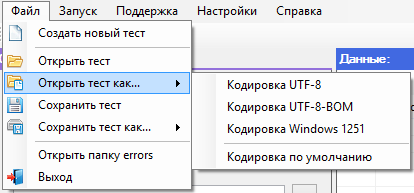
Для корректной работы файла run-test.sh так же требуется предоставить права на чтение и запись и включить разрешение на выполнение файла как программу.



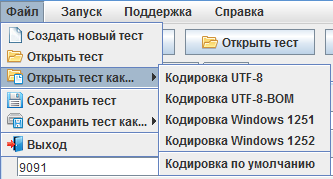
*Created with the Personal Edition of HelpNDoc:* [*Full-featured Help generator*](https://www.helpndoc.com/feature-tour)

**Основное меню**

**Основное меню программы**



(версия для Windows)



(версия для Linux)

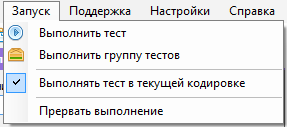
Чтобы создать тест нужно в меню "Файл" нажать пункт "Создать новый тест"

Чтобы открыть тест нужно в меню "Файл" нажать пункт "Открыть тест"

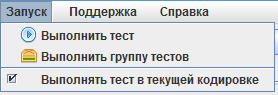
Чтобы открыть тест используя специальную кодировку нужно в меню "Файл" нажать пункт "Открыть тест как..." и выбрать необходимую кодировку

Чтобы сохранить тест нужно в меню "Файл" нажать пункт "Сохранить тест"

Чтобы сохранить тест используя специальную кодировку нужно в меню "Файл" нажать пункт "Сохранить тест как" и выбрать необходимую кодировку



(версия для Windows)



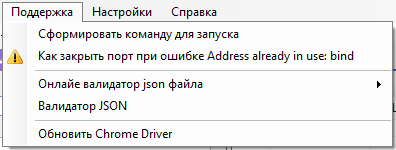
(версия для Linux)

Чтобы запустить выполнение теста нужно в меню "Запуск" нажать пункт "Выполнить тест"

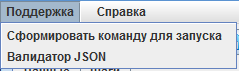
Чтобы запустить выполнение группы тестов нужно в меню "Запуск" нажать пункт "Выполнить группу тестов"

Чтобы тесты учитывали кодировку необходимо включить флаг "Выполнять тест в текущей кодировку"

Чтобы прервать выполнение запущенного теста нужно в меню "Запуск" нажать пункт "Прервать выполнение"



(версия для Windows)



(версия для Linux)

Если нужно сформировать команду для запуска теста например из bat файла, для этого в меню "Поддержка" нажать пункт "Сформировать команду для запуска"

Если нужно выполнить проверку json файла, для этого в меню "Поддержка" выберите пункт "Валидатор JSON"

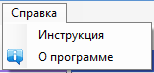
Если заблокирован порт вы можете воспользоваться подсказкой "Как закрыть порт при ошибке Address already in use: bind" выполнив описанные действия вы сможете разблокировать порт.

Так же присутствуют ссылки на валидаторы json файлов и на источник веб-драйвера.

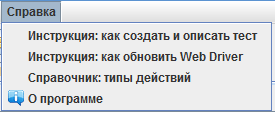


(версия для Windows)

В настройках вы можете включить параметр детального вывода информации при выполнении теста



(версия для Windows)



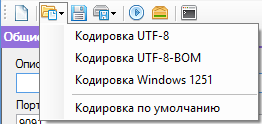
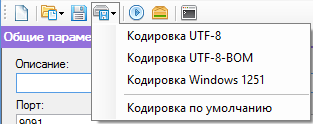
(версия для Linux)

Встроенная справочная информация поможет разобраться в работе с программой.

*Created with the Personal Edition of HelpNDoc:* [*Free help authoring environment*](https://www.helpndoc.com/help-authoring-tool)

**Панель инструментов**

**Панель инструментов**

(версия для Windows)

Панель инструментов дублирует основные функции из основного меню.

* Создать новый тест
* Открыть тест как...
* Сохранить тест
* Сохранить тест как...
* Выполнить тест
* Выполнить группу тестов
* Открыть системную консоль



(версия для Linux)

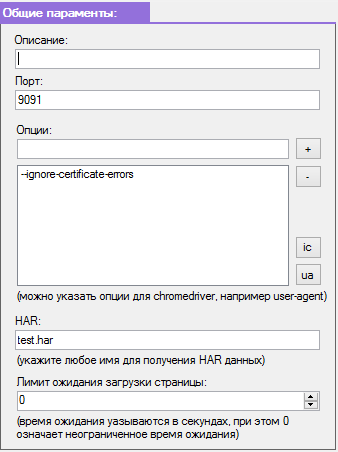
Панель инструментов дублирует основные функции из основного меню.

* Создать новый тест
* Открыть тест
* Сохранить тест
* Выполнить тест

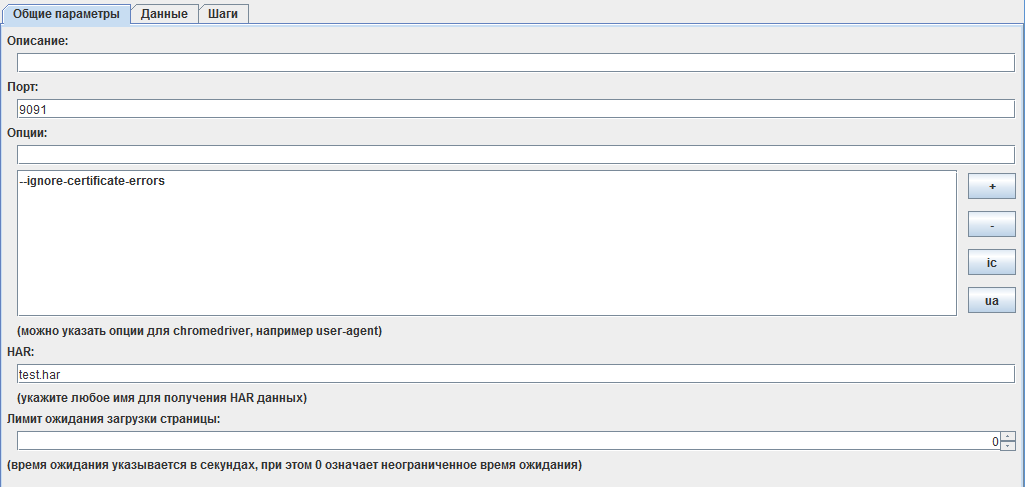
*Created with the Personal Edition of HelpNDoc:* [*Easily create PDF Help documents*](https://www.helpndoc.com/feature-tour)

**Общие параметры**

**Общие параметры**



(версия для Windows)



(версия для Linux)

На вкладке "Общие параметры" задаются следующие параметры теста:

Описание - текстовое поле для краткого описания теста (наименование)

Порт - текстовое поле в котором указывается порт для работы прокси (по умолчанию 9091)

Опции - список настроек необходимых при работе Chrome WebDriver (по умолчанию --ignore-certificate-errors)

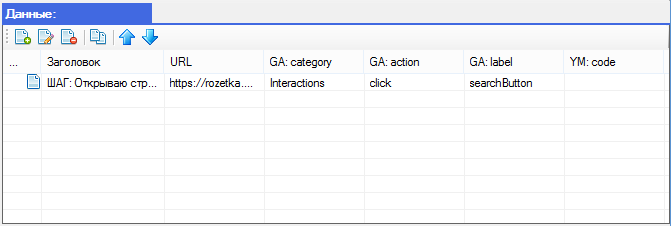
HAR - текстовое поле в котором указывается имя для текущего пакета данных

Лимит - текстовое поле в котором указывается количество секунд ожидания для загрузки старницы (0 означает неограниченное время ожидания)

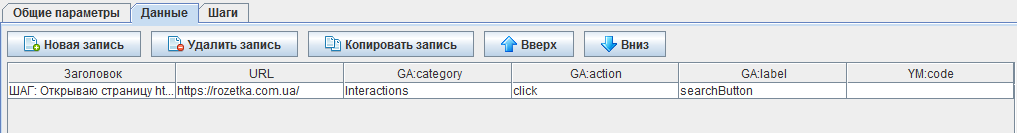
*Created with the Personal Edition of HelpNDoc:* [*Full-featured Kindle eBooks generator*](https://www.helpndoc.com/feature-tour/create-ebooks-for-amazon-kindle)

**Данные**

**Данные**



(версия для Windows)



(версия для Linux)

На вкладке "Данные" задаются параметры для проверки событий GA и YM

Заголовок - текстовое поле для ввода любого текста выводимого в консоль при выполнении теста

URL - текстовое поле в котором указывается ссылка на страницу на которой будет выполнятся тест

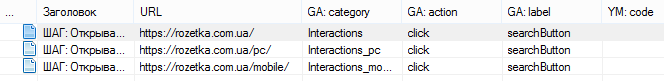
GA:category - текстовое поле для ввода ключа category события GA

GA:action - текстовое поле для ввода ключа action события GA

GA:label - текстовое поле для ввода ключа label события GA

YM:code - текстовое поле для ввода ключа события YM

Данная таблица выполняется цеклично, то есть все строки в таблице будут выполнены по очереди при запуске теста. При выполнении каждой строки будут использоваться те действия которые описаны в таблице "Шаги"

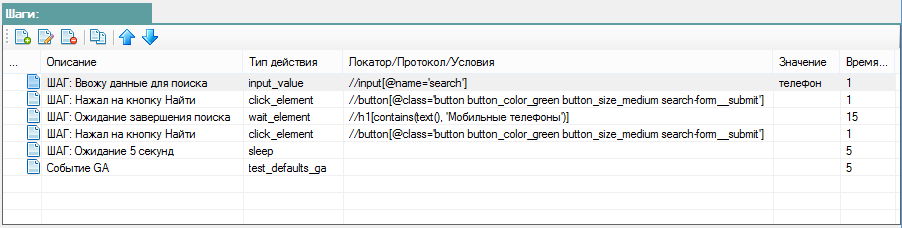


Таким образом вы можете одним тестом выполнить одинаковую проверку для множества страниц.

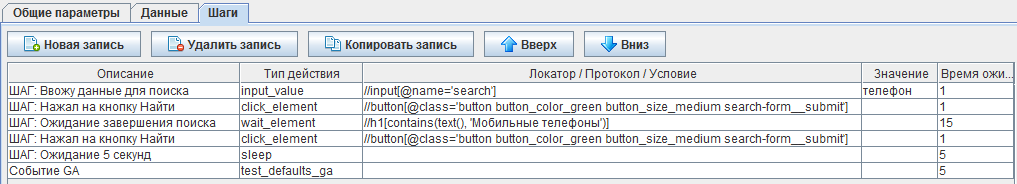
*Created with the Personal Edition of HelpNDoc:* [*Free CHM Help documentation generator*](https://www.helpndoc.com)

**Шаги**

**Шаги**



(версия для Windows)



(версия для Linux)

На вкладке "Шаги" описываются шаги которые необходимо выполнить на странице сайта включая проверку событий GA/YM

Описание - текстовое поле в котором описывается данный шаг, данный текст выводится в консоль при выполнении теста

Тип действия - текстовое поле для выбора или ввода ключа действия

Локатор/Протокол/Условия - текстовое поле для ввода локатора в формате xpath или протокол GA/YM или ключ условия

Значение - текстовое поле для ввода значения которое необходимо вписать в какой либо элемент (например input) или использовать в условии

Время ожидания - указывается количество секунд ожидания для выполнения действия (или количество попыток при проверке событий GA/YM)

*Created with the Personal Edition of HelpNDoc:* [*Free Qt Help documentation generator*](https://www.helpndoc.com)

**Типы действий в шагах**

**Справочник: Типы действий в шагах**

open\_page - открывает указанную страницу сайта в браузере.

- Описание - текстовое представление действия

- Локатор - не нужно заполнять

- Значение - ссылка на страницу формата https://site.com

- Время ожидания - не нужно заполнять

open\_default\_page - открывает страницу сайта (прописанную на вкладке Данные) в браузере.

- Описание - текстовое представление действия

- Локатор - не нужно заполнять

- Значение - не нужно заполнять

- Время ожидания - не нужно заполнять

refresh\_page - перезагружает текущую страницу.

- Описание - текстовое представление действия

- Локатор - не нужно заполнять

- Значение - не нужно заполнять

- Время ожидания - не нужно заполнять

input\_value - ввод данных в поле (элемент input).

- Описание - текстовое представление действия

- Локатор - в формате XPath запрос к элементу html страницы

- Значение - данные которые должны быть введены в поле (или данные из find\_element)

- Время ожидания - количество секунд до появления элемента

click\_element - нажать на элемент страницы.

- Описание - текстовое представление действия

- Локатор - в формате XPath запрос к элементу html страницы

- Значение - не нужно заполнять

- Время ожидания - количество секунд до появления элемента

find\_element - выполняется поиск элемента который затем будет хранится в глобальной переменной.

- Описание - текстовое представление действия

- Локатор - в формате XPath запрос к элементу html страницы

- Значение - не нужно заполнять

- Время ожидания - количество секунд до появления текста

if\_get\_text - начало условия при котором сравниваются данные (text) элемента глобальной переменной (find\_element)

- Описание - текстовое представление действия

- Условие - значение в формате: ==, !=, <, <=, >, >= (равно, не равно, меньше, меньше или равно, больше, больше или равно)

- Значение - данные которые должны быть проверены на соответствие

- Время ожидания - не нужно заполнять

if\_get\_attribute\_ - начало условия при котором сравниваются данные (attribute) элемента глобальной переменной (find\_element)

(обязательно нужно дописывать объект, например if\_get\_attribute\_id)

- Описание - текстовое представление действия

- Условие - значение в формате: ==, !=, <, <=, >, >= (равно, не равно, меньше, меньше или равно, больше, больше или равно)

- Значение - данные которые должны быть проверены на соответствие

- Время ожидания - не нужно заполнять

else\_if\_get\_text - иначе условие при котором сравниваются данные (text) элемента глобальной переменной (find\_element)

- Описание - текстовое представление действия

- Условие - значение в формате: ==, !=, <, <=, >, >= (равно, не равно, меньше, меньше или равно, больше, больше или равно)

- Значение - данные которые должны быть проверены на соответствие

- Время ожидания - не нужно заполнять

else\_if\_get\_attribute\_ - иначе условие при котором сравниваются данные (attribute) элемента глобальной переменной (find\_element)

(обязательно нужно дописывать объект, например else\_if\_get\_attribute\_id)

- Описание - текстовое представление действия

- Условие - значение в формате: ==, !=, <, <=, >, >= (равно, не равно, меньше, меньше или равно, больше, больше или равно)

- Значение - данные которые должны быть проверены на соответствие

- Время ожидания - не нужно заполнять

else - иначе условие выполняется в том случае если критерии условия не выполнены.

- Описание - не нужно заполнять

- Локатор - не нужно заполнять

- Значение - не нужно заполнять

- Время ожидания - не нужно заполнять

end\_if - завершение условия.

- Описание - не нужно заполнять

- Локатор - не нужно заполнять

- Значение - не нужно заполнять

- Время ожидания - не нужно заполнять

wait\_text - ожидание появления текста на странице.

- Описание - текстовое представление действия

- Локатор - в формате XPath запрос к элементу html страницы

- Значение - текст который должен быть в элементе

- Время ожидания - количество секунд до появления текста

wait\_element - ожидание появления элемента на странице.

- Описание - текстовое представление действия

- Локатор - в формате XPath запрос к элементу html страницы

- Значение - не нужно заполнять

- Время ожидания - количество секунд до появления элемента

wait\_element\_not\_visible - ожидание удаления элемента со страницы.

- Описание - текстовое представление действия

- Локатор - в формате XPath запрос к элементу html страницы

- Значение - не нужно заполнять

- Время ожидания - количество секунд до удаления элемента

sleep - задержка действия на указанное количество секунд.

- Описание - текстовое представление действия

- Локатор - не нужно заполнять

- Значение - не нужно заполнять

- Время ожидания - количество секунд паузы

execute\_js - позволяет выполнять JavaScript код.

- Описание - текстовое представление действия

- Локатор - не нужно заполнять

- Значение - строка JavaScript кода (двойные кавычки обязательно нужно экранировать)

- Время ожидания - не нужно заполнять

test\_defaults\_ga - проверка GA в режиме по умолчанию (данными берутся из вкладки Данные).

- Описание - текстовое представление действия

- Локатор - не нужно заполнять

- Значение - не нужно заполнять

- Время ожидания - количество попыток чтобы найти GA значение

test\_optionally\_ga - опциональная проверка GA (данные вводятся вручную).

- Описание - текстовое представление действия

- Протокол - по умолчанию google-analytics.com/collect

- Значение - ключевое слово GA

- Время ожидания - количество попыток чтобы найти GA значение

test\_defaults\_ym - проверка YM в режиме по умолчанию (данными берутся из вкладки Данные).

- Описание - текстовое представление действия

- Локатор - не нужно заполнять

- Значение - не нужно заполнять

- Время ожидания - количество попыток чтобы найти YM значение

test\_optionally\_ym - опциональная проверка YM (данные вводятся вручную).

- Описание - текстовое представление действия

- Протокол - по умолчанию mc.yandex.ru/watch

- Значение - ключевое слово YM

- Время ожидания - количество попыток чтобы найти YM значение

clear\_har - очистить список отправленных событий.

- Описание - не нужно заполнять

- Локатор - не нужно заполнять

- Значение - не нужно заполнять

- Время ожидания - не нужно заполнять

get\_har - получить список всех отправленных событий.

- Описание - не нужно заполнять

- Локатор - не нужно заполнять

- Значение - не нужно заполнять

- Время ожидания - не нужно заполнять

get\_har\_ga - получить список всех отправленных GA событий.

- Описание - не нужно заполнять

- Локатор - не нужно заполнять

- Значение - не нужно заполнять

- Время ожидания - не нужно заполнять

get\_har\_ym - получить список всех отправленных YM событий.

- Описание - не нужно заполнять

- Локатор - не нужно заполнять

- Значение - не нужно заполнять

- Время ожидания - не нужно заполнять

*Created with the Personal Edition of HelpNDoc:* [*Easy CHM and documentation editor*](https://www.helpndoc.com)

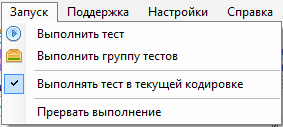
**Выполнение тестов**

**Выполнение тестов**

Есть два способа запускать тесты:

* запуск одного отдельного теста
* запуск группы тестов

**Запуск одного отдельного тест**



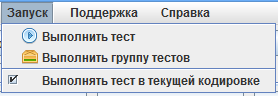
(версия для Windows)

Чтобы запустить выполнение теста его нужно открыть в редакторе и в меню "Запуск" нажать пункт "Выполнить тест". При запуске теста редактор формирует команды и записывает их в файл run-test.bat в папке bin.

cd C:\DetectGaYm\bin

java -jar detect-gaym.jar UTF-8 C:\...\DetectGaYm\tests\test-search-ga.json

Чтобы прервать выполнение запущенного теста нужно в меню "Запуск" нажать пункт "Прервать выполнение"



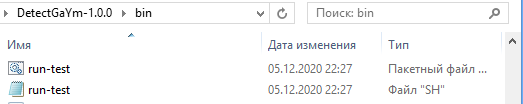
(версия для Linux)

Чтобы запустить выполнение теста его нужно открыть в редакторе и в меню "Запуск" нажать пункт "Выполнить тест". При запуске теста редактор формирует команды и записывает их в файл run-test.sh в папке bin.

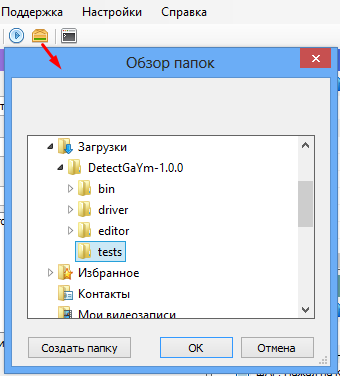
#!/bin/bash

cd /DetectGaYm/bin

java -jar detect-gaym.jar UTF-8 /home/.../DetectGaYm/tests/test-search-ga.json



**Запуск группы тестов**

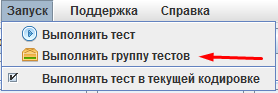


(версия для Windows)

Чтобы запустить выполнение группы тестов нужно в меню "Запуск" нажать пункт "Выполнить группу тестов". Будет открыто диалоговое окно для выбора папки с тестами. При запуске редактор формирует команды и записывает их в файл run-test.bat в папке bin.

cd C:\DetectGaYm\bin

java -jar detect-gaym.jar UTF-8 C:\...\DetectGaYm\tests\



(версия для Linux)

Чтобы запустить выполнение группы тестов нужно открыть в редакторе любой тест из группы тестов и в меню "Запуск" нажать пункт "Выполнить группу тестов". При запуске редактор формирует команды и записывает их в файл run-test.sh в папке bin.

#!/bin/bash

cd /DetectGaYm/bin

java -jar detect-gaym.jar UTF-8 /home/.../DetectGaYm/tests/

Чтобы тесты учитывали кодировку необходимо включить флаг "Выполнять тест в текущей кодировку"





Если флаг отключен редактор не будет учитывать кодировку выбранную при открытии/сохранении файла и будет использывать кодировку по умолчанию (в системе Windows это Windows-1251 в системе Linux это UTF-8)

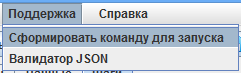
*Created with the Personal Edition of HelpNDoc:* [*Full-featured multi-format Help generator*](https://www.helpndoc.com/help-authoring-tool)

**Сформировать команду для запуска**

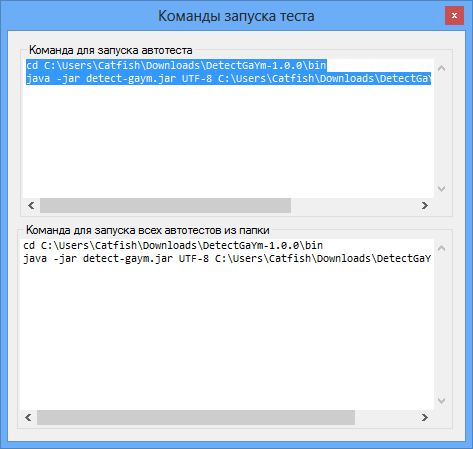
**Сформировать команду для запуска**

В редакторе тестов в меню "Поддержка" добавлена специальная функция которая позволяет формировать команды запуска для редактируемого сценария.

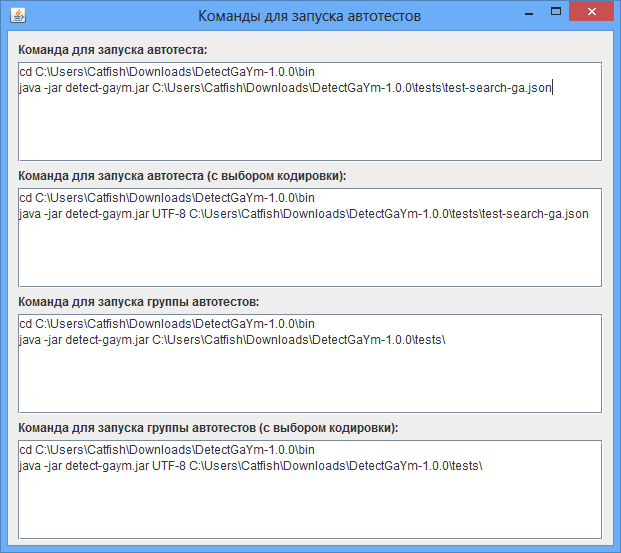




Такми образов вы легко и просто получити сформированные команды для запуска



(версия для Linux)



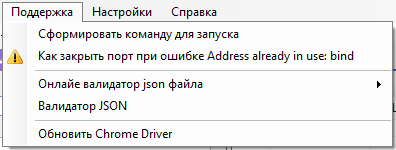
(версия для Linux)

Данный способ запуска тестов удобно использовать в программах предназначенных для обеспечения процесса непрерывной интеграции, таких как [Jenkins](https://www.jenkins.io/) или [TeamCity](https://www.jetbrains.com/ru-ru/teamcity/). Вам нужно будет всего лишь вписать команды запуска.

*Created with the Personal Edition of HelpNDoc:* [*Easily create PDF Help documents*](https://www.helpndoc.com/feature-tour)

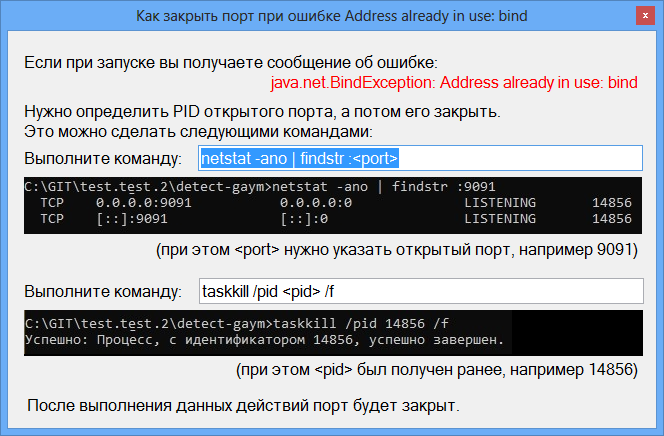
**Закрыть заблокированный порт**

**Закрыть заблокированный порт**



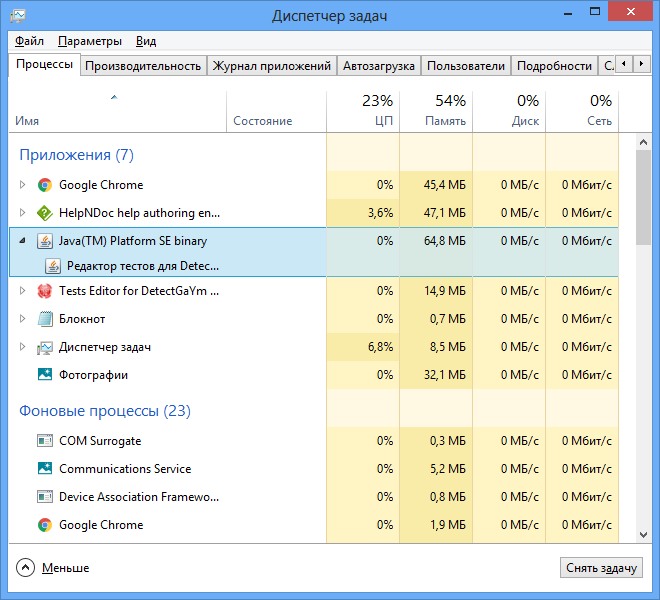
(версия для Windows)

Если заблокирован порт вы можете воспользоваться подсказкой "Как закрыть порт при ошибке Address already in use: bind" выполнив описанные действия вы сможете разблокировать порт.



(версия для Windows)

Или в диспетчере задач найти данную задачу и закрыть её нажав кнопку "Снять задачу"

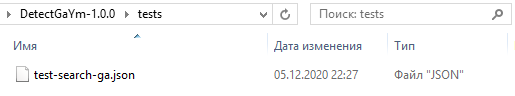


*Created with the Personal Edition of HelpNDoc:* [*Free help authoring environment*](https://www.helpndoc.com/help-authoring-tool)

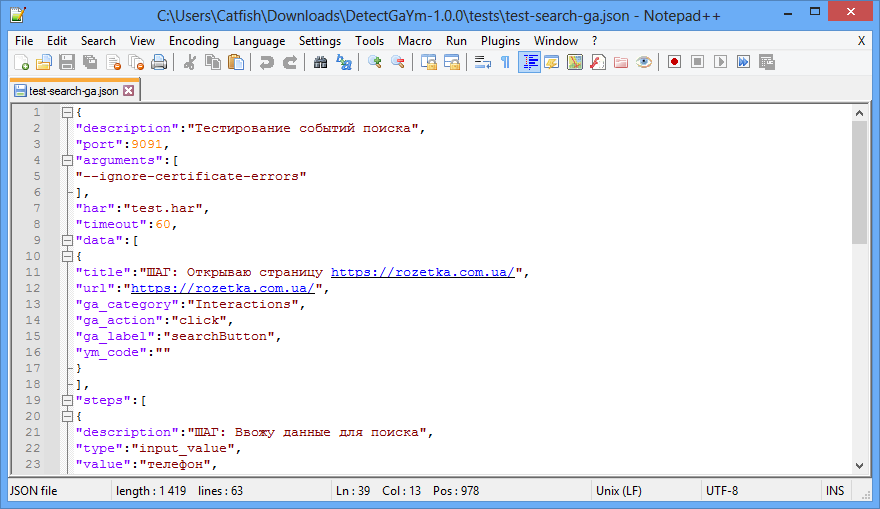
**Структура теста**

**Структура теста**

Сценарии тестов пишутся в формате [JSON](https://www.json.org/) и могут храниться в любом месте на жестком диске.



Под сценарием понимается файл в формате [JSON](https://www.json.org/) внутри которого описаны настройки необходимые для работы автоматизированного теста, а так же входные данные и шаги их выполнения.   
Сценарий можно создать в любом текстовом редакторе но удобнее это делать в специальном редакторе тестов.



Ниже приведен пример демонстрационного теста

Блок описания общих параметров:

{

"description":"Тестирование событий поиска",

"port":9091,

"arguments":[

"--ignore-certificate-errors"

],

"har":"test.har",

"timeout":60,

Блок описания данных:

"data":[

{

"title":"ШАГ: Открываю страницу https://rozetka.com.ua/",

"url":"https://rozetka.com.ua/",

"ga\_category":"Interactions",

"ga\_action":"click",

"ga\_label":"searchButton",

"ym\_code":""

}

],

Блок описания шагов:

"steps":[

{

"description":"ШАГ: Ввожу данные для поиска",

"type":"input\_value",

"value":"телефон",

"locator":"//input[@name='search']",

"timeout":1

},

{

"description":"ШАГ: Нажал на кнопку Найти",

"type":"click\_element",

"value":"",

"locator":"//button[@class='button button\_color\_green button\_size\_medium search-form\_\_submit']",

"timeout":1

},

{

"description":"ШАГ: Ожидание завершения поиска",

"type":"wait\_element",

"value":"",

"locator":"//h1[contains(text(), 'Мобильные телефоны')]",

"timeout":15

},

{

"description":"ШАГ: Нажал на кнопку Найти",

"type":"click\_element",

"value":"",

"locator":"//button[@class='button button\_color\_green button\_size\_medium search-form\_\_submit']",

"timeout":1

},

{

"description":"ШАГ: Ожидание 5 секунд",

"type":"sleep",

"value":"",

"locator":"",

"timeout":5

},

{

"description":"Событие GA",

"type":"test\_defaults\_ga",

"value":"",

"locator":"",

"timeout":5

}

]

}

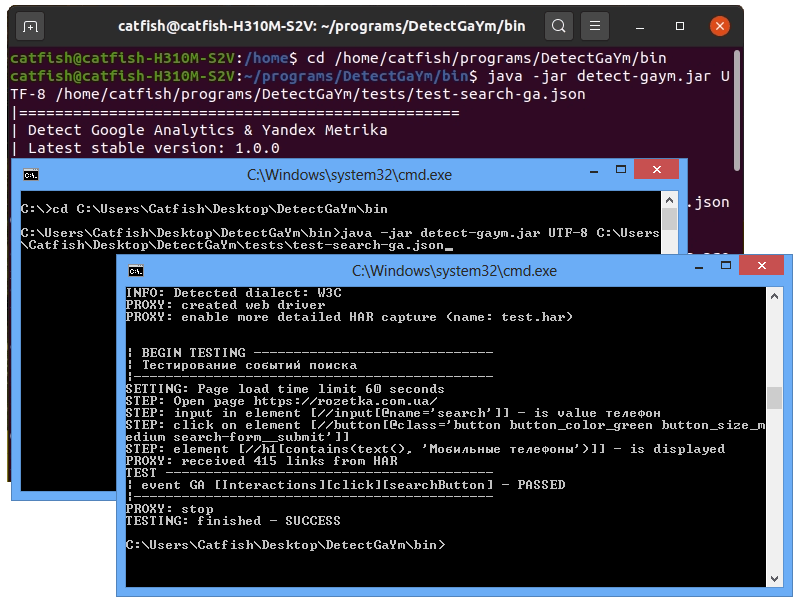
Для создания и редактирования тестов рекомендуется использовать специальный редактор.

*Created with the Personal Edition of HelpNDoc:* [*Free Kindle producer*](https://www.helpndoc.com/feature-tour/create-ebooks-for-amazon-kindle)

**Запуск тестов из командной строки**

**Запуск тестов из командной строки**

Программа DetectGaYm позволяет запускать сценарии автоматизированного тестирования из консоли. В системе Windows консоль это CMD (command line interpreter), а в системе Linux консоль это Terminal (gnome terminal)



В системе Windows чтобы запустить заранее созданный тест нужно открыть консоль, перейти в папку bin и выполнить следующие команды:

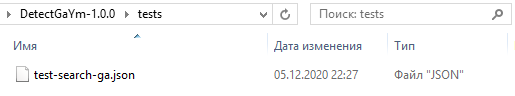
cd C:\...\DetectGaYm\bin

java -jar detect-gaym.jar UTF-8 C:\...\DetectGaYm\tests\test-search-ga.json

В системе Linux чтобы запустить заранее созданный тест нужно открыть консоль, перейти в папку bin и выполнить следующие команды:

cd /home/.../DetectGaYm/bin

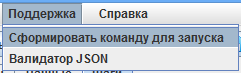
java -jar detect-gaym.jar UTF-8 /home/.../DetectGaYm/tests/test-search-ga.json



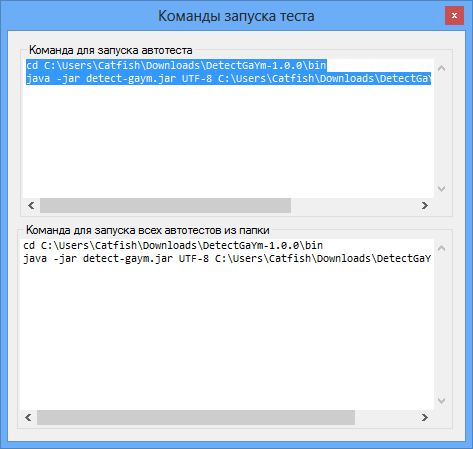
Данный способ запуска тестов удобно использовать в программах предназначенных для обеспечения процесса непрерывной интеграции, таких как [Jenkins](https://www.jenkins.io/) или [TeamCity](https://www.jetbrains.com/ru-ru/teamcity/). Вам нужно будет всего лишь вписать команды запуска.

В редакторе тестов в меню "Поддержка" добавлена специальная функция которая позволяет формировать команды запуска для редактируемого сценария.

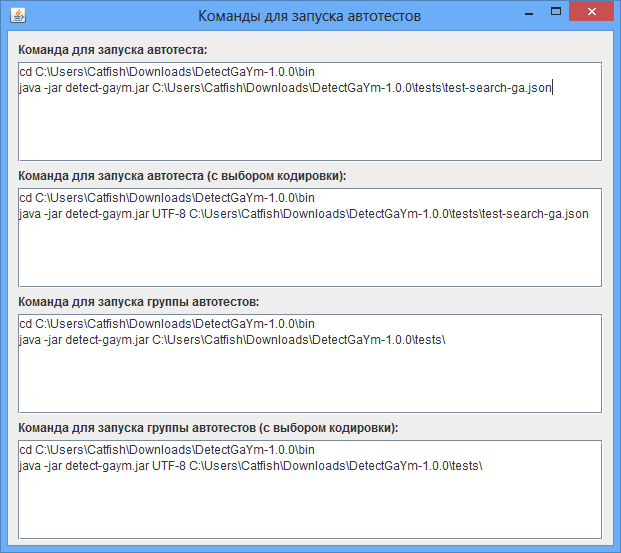




Такми образов вы легко и просто получити сформированные команды для запуска



(версия для Linux)



(версия для Linux)

*Created with the Personal Edition of HelpNDoc:* [*Create help files for the Qt Help Framework*](https://www.helpndoc.com/feature-tour/create-help-files-for-the-qt-help-framework)

**Запуск тестов в Jenkins**

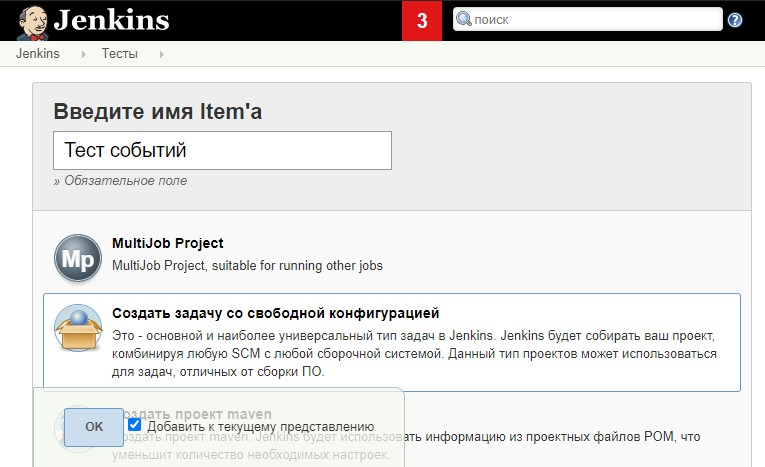
**Запуск тестов в Jenkins**

Данный пример вы можете увидеть на видео: <https://youtu.be/M0WG0gu_6b8?t=483>

Перейдите в систему Jenkins и создайте новый Item

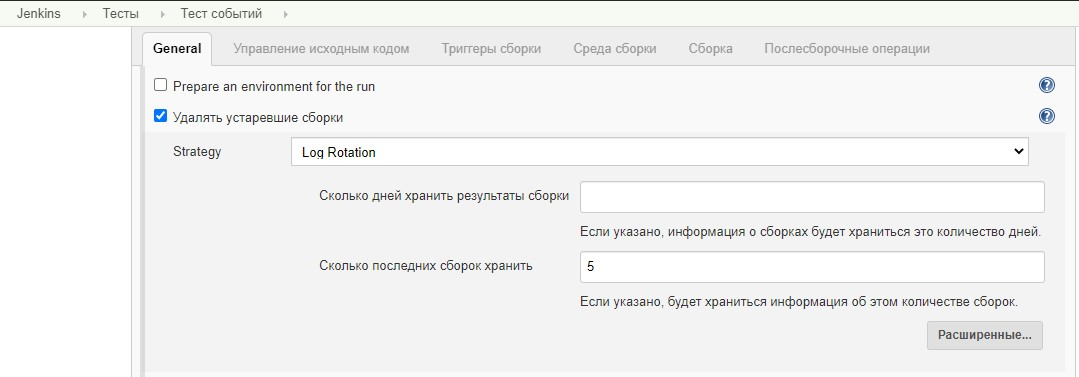


Укажите наименование сборки и выберите тип (например задача со свободной конфигурацией)

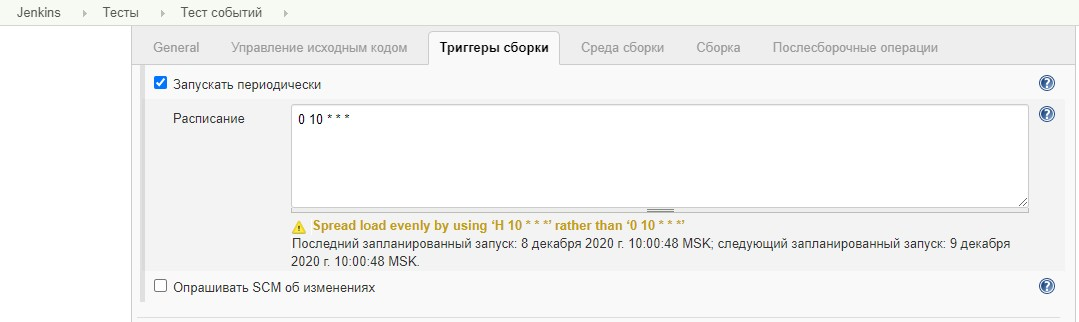


Настройки сборки:

Включите опцию удаление устаревших сборок, оставьте в хранилище 5-ть последних сборок.

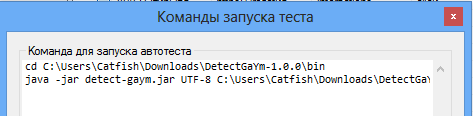


Включите опцию периодического запуска, это позволит запускать автотест автоматически в заданное время (например в 10:00 утра)

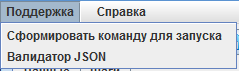


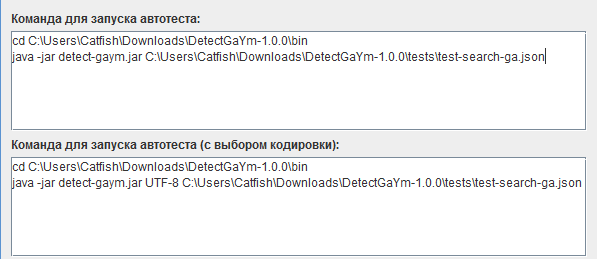
Дальше необходимо перейти в редактор тестов и сформировать команду запуска.





(версия для Windows)



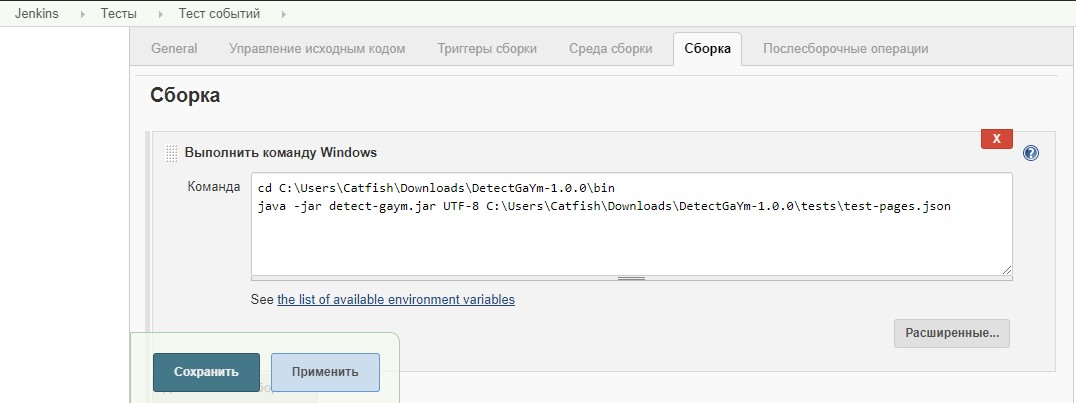


(версия для Linux)

Копируем сформированную команду в буфер обмена.

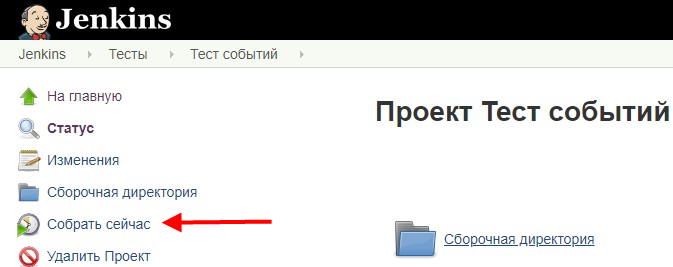
Переходим в Jenkins и включаем опцию сборки "Выполнение команды Windows"

Вставляем скопированную команду в поле "Команда"

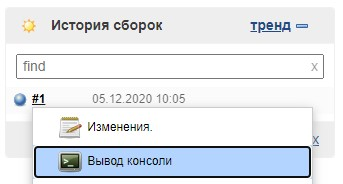


Сохраняем Item

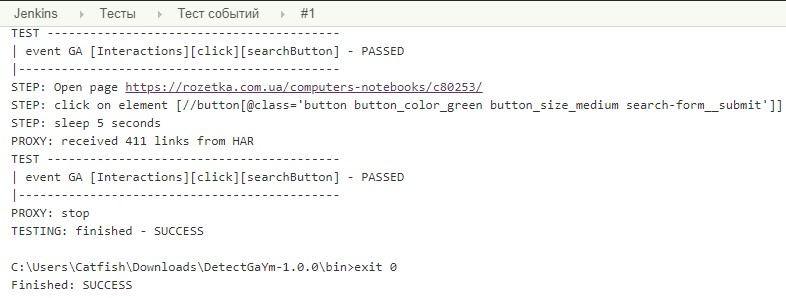
Теперь необходимо перейти в наш созданный Item и запустить сборку.



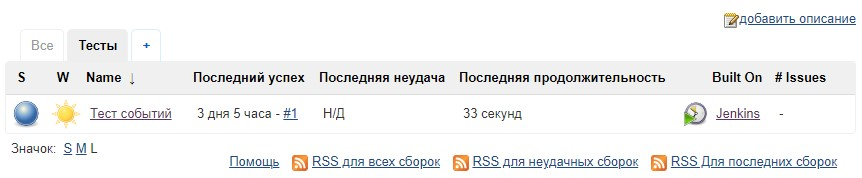
Во время сборки можно открыть консоль и посмотреть ход выполнения теста.



В консоли отобразится вся выводимая тестом информация



В результате в Jenkins будет создана сборка которая покажет результат тестирования



Данный пример вы можете увидеть на видео: <https://youtu.be/M0WG0gu_6b8?t=483>

*Created with the Personal Edition of HelpNDoc:* [*Write eBooks for the Kindle*](https://www.helpndoc.com/feature-tour/create-ebooks-for-amazon-kindle)

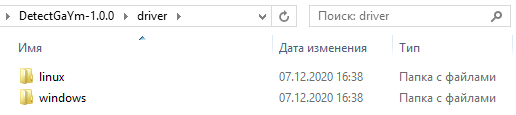
**Обновление веб-драйвера**

**Обновление веб-драйвера**

Программа DetectGaYm способна выполнять действия над элементами страницы в окне браузера используя для этого [Selenium](https://www.selenium.dev/) - инструмент автоматизации действий веб-браузера. Для его работы используется специальный [WebDriver](https://chromedriver.chromium.org/) под браузер Google Chrome.

Периодически браузер Google Chrome обновляетя и в результате этого [WebDriver](https://chromedriver.chromium.org/) перестает корректно работать с браузером. В этом случае необходимо зайти на официальную страницу [ChromeDriver - WebDriver for Chrome](https://chromedriver.chromium.org/) и скачать архив с последней стабильной версией Latest stable release:

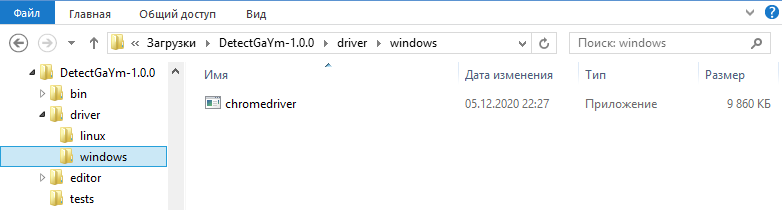
* chromedriver\_win32.zip - архив с драйвером для системы Windows
* chromedriver\_linux64.zip - архив с драйвером для системы Linux



Далее драйвер необходимо распаковать из скаченного архива в папки:

DetectGaYm\driver\windows - для системы Windows

DetectGaYm/driver/linux - для системы Linux



В системе Linux кроме обновления драйвера в программе DetectGaYm так же необходимо обновить драйвер в системе. Для этого нужно выполнить команды:

unzip chromedriver\_linux64.zip

sudo mv chromedriver /usr/local/bin/chromedriver

sudo chown root:root /usr/local/bin/chromedriver

sudo chmod +x /usr/local/bin/chromedriver

chromedriver --version

*Created with the Personal Edition of HelpNDoc:* [*Free EPub producer*](https://www.helpndoc.com/create-epub-ebooks)

**Примеры разработки тестов**

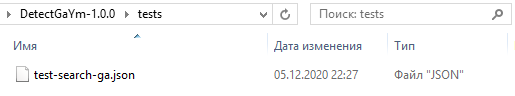
*Created with the Personal Edition of HelpNDoc:* [*Create cross-platform Qt Help files*](https://www.helpndoc.com/feature-tour/create-help-files-for-the-qt-help-framework)

**Демонстрационный тест**

**Демонстрационный тест**

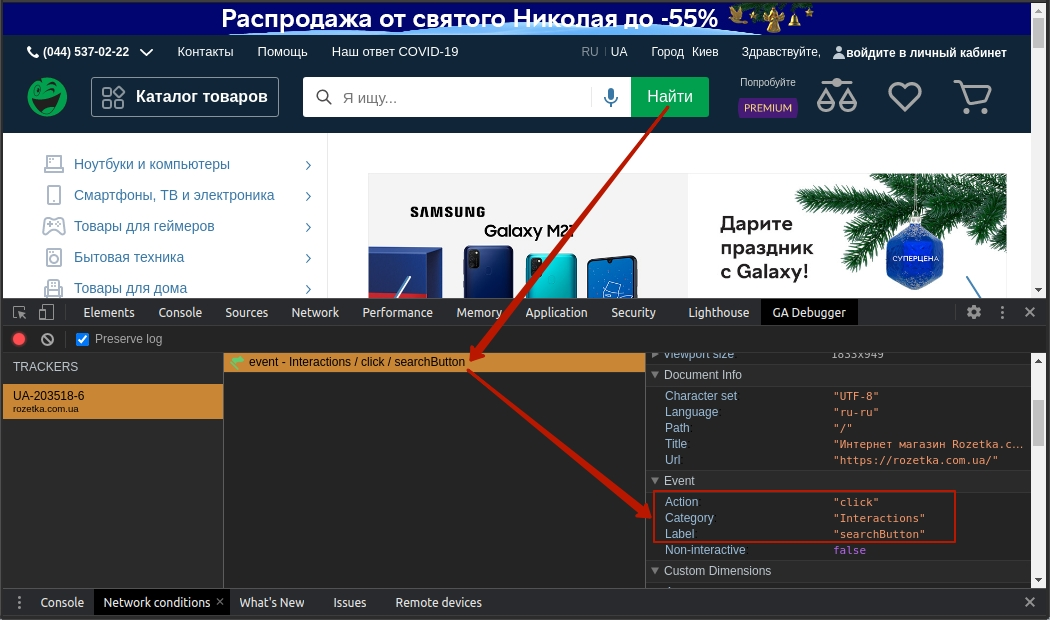
В папке tests вы найдете демонстрационный тест, данный сценарий выпонялет проверку событий GA.

Структура тестов описана в разделе [структура теста](#_topic_Newtopic17)



В данном примере используется сайт [https://rozetka.com.ua](https://rozetka.com.ua/) на котором присутствуют события GA.

При нажатии на кнопку Найти мы можем видить ключи события GA, эти ключи используются в тесте.



Откроем файл с тестом test-search-ga.json в редакторе.

Рассмотрим общие параметры необходимые для работы теста.

|  |  |
| --- | --- |
|  | **Общие параметры:**  На вкладке "Общие параметры" задаются следующие параметры теста:  Описание - текстовое поле для краткого описания теста (наименование)  Порт - текстовое поле в котором указывается порт для работы прокси (по умолчанию 9091)  Опции - список настроек необходимых при работе Chrome WebDriver (по умолчанию --ignore-certificate-errors)  HAR - текстовое поле в котором указывается имя для текущего пакета данных.  Лимит - текстовое поле в котором указывается количество секунд ожидания для загрузки старницы (0 означает неограниченное время ожидания) |

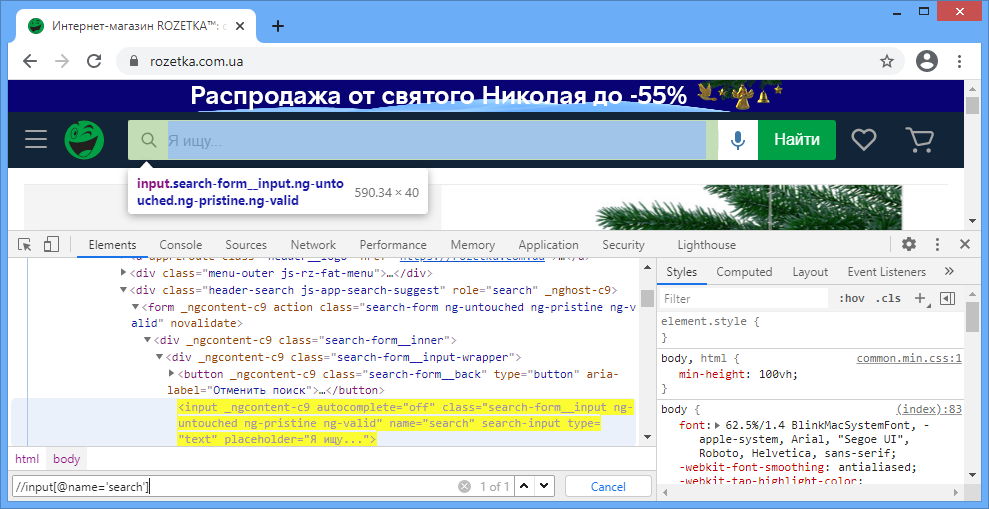
Рассмотрим таблицу данных.

Данная таблица выполняется цеклично, то есть все строки в таблице будут выполнены по очереди при запуске теста. При выполнении каждой строки будут использоваться те действия которые описаны в таблице "Шаги"

|  |
| --- |
|  |
| На вкладке "Данные" задаются параметры для проверки событий GA и YM  Заголовок - текстовое поле для ввода любого текста выводимого в консоль при выполнении теста  URL - текстовое поле в котором указывается ссылка на страницу на которой будет выполнятся тест  GA:category - текстовое поле для ввода ключа category события GA  GA:action - текстовое поле для ввода ключа action события GA  GA:label - текстовое поле для ввода ключа label события GA  YM:code - текстовое поле для ввода ключа события YM |

Рассмотрим таблицу шагов.

Для выполнения действий над элементами необходимо сформировать локаторы этих элементов



Конструировать локаторы xpath удобно в интрументе разработчика браузера Chrome.

|  |
| --- |
|  |
| На вкладке "Шаги" описываются шаги которые необходимо выполнить на странице сайта включая проверку событий GA/YM  Описание - текстовое поле в котором описывается данный шаг, данный текст выводится в консоль при выполнении теста  Тип действия - текстовое поле для выбора или ввода ключа [действия](#_topic_Newtopic9)  Локатор/Протокол/Условия - текстовое поле для ввода локатора в формате xpath или протокол GA/YM или ключ условия  Значение - текстовое поле для ввода значения которое необходимо вписать в какой либо элемент (например input) или использовать в условии  Время ожидания - указывается количество секунд ожидания для выполнения действия (или количество попыток при проверке событий GA/YM) |

Справочник с описанием всех встроенных [типов действий](#_topic_Newtopic9) вы найдете в этой документации.

Локаторы описываются в формате [xpath](https://ru.wikipedia.org/wiki/XPath) и позволяют взаимодействовать с элементами страницы.

Запустив данный тест будет выполнена проверка событий GA на сайте Розетка.

Исходный код теста:

{

"description":"Тестирование событий поиска",

"port":9091,

"arguments":[

"--ignore-certificate-errors"

],

"har":"test.har",

"timeout":60,

"data":[

{

"title":"ШАГ: Открываю страницу https://rozetka.com.ua/",

"url":"https://rozetka.com.ua/",

"ga\_category":"Interactions",

"ga\_action":"click",

"ga\_label":"searchButton",

"ym\_code":""

}

],

"steps":[

{

"description":"ШАГ: Ввожу данные для поиска",

"type":"input\_value",

"value":"телефон",

"locator":"//input[@name='search']",

"timeout":1

},

{

"description":"ШАГ: Нажал на кнопку Найти",

"type":"click\_element",

"value":"",

"locator":"//button[@class='button button\_color\_green button\_size\_medium search-form\_\_submit']",

"timeout":1

},

{

"description":"ШАГ: Ожидание завершения поиска",

"type":"wait\_element",

"value":"",

"locator":"//h1[contains(text(), 'Мобильные телефоны')]",

"timeout":15

},

{

"description":"ШАГ: Нажал на кнопку Найти",

"type":"click\_element",

"value":"",

"locator":"//button[@class='button button\_color\_green button\_size\_medium search-form\_\_submit']",

"timeout":1

},

{

"description":"ШАГ: Ожидание 5 секунд",

"type":"sleep",

"value":"",

"locator":"",

"timeout":5

},

{

"description":"Событие GA",

"type":"test\_defaults\_ga",

"value":"",

"locator":"",

"timeout":5

}

]

}

*Created with the Personal Edition of HelpNDoc:* [*Create cross-platform Qt Help files*](https://www.helpndoc.com/feature-tour/create-help-files-for-the-qt-help-framework)

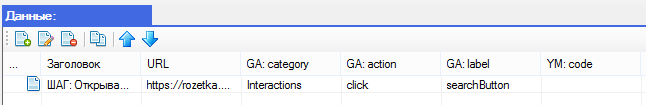
**Базовая проверка событий**

**Базовая проверка событий**

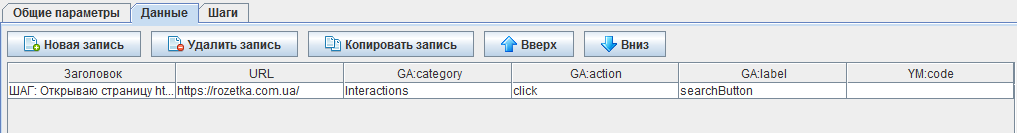
При базовой проверке используются [действия](#_topic_Newtopic9):

* test\_defaults\_ga
* test\_defaults\_ym

Что значит базовая проверка? Это значит что ключи GA и YM описываются в таблице "Данные" (смотрите раздел [Данные](#_topic_Newtopic7))



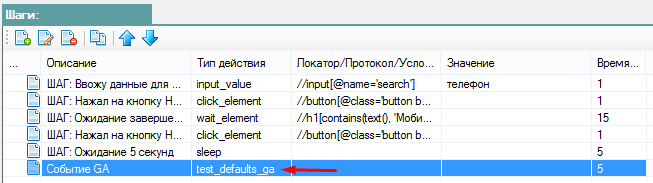
(версия для Windows)



(версия для Linux)

Для того чтобы выполнить проверку ключей указанных в таблице "Данные" нужно использовать специальные [действия](#_topic_Newtopic9) перечисленные выше в таблице "Шаги".

В начале идут [действия](#_topic_Newtopic9) приводящие к срабатыванию событий GA и YM, а затем вызывается [действие](#_topic_Newtopic9) базовой проверки test\_defaults\_ga и/или test\_defaults\_ym.



[Действие](#_topic_Newtopic9) test\_defaults\_ga и/или test\_defaults\_ym смотрит ключи указанные в таблице "Данные" и сверяет их с отправленными ключами в момент выполнения действий на странице.

Исходный код теста:

{

"description":"Тестирование событий поиска",

"port":9091,

"arguments":[

"--ignore-certificate-errors"

],

"har":"test.har",

"timeout":60,

"data":[

{

"title":"ШАГ: Открываю страницу https://rozetka.com.ua/",

"url":"https://rozetka.com.ua/",

"ga\_category":"Interactions",

"ga\_action":"click",

"ga\_label":"searchButton",

"ym\_code":""

}

],

"steps":[

{

"description":"ШАГ: Ввожу данные для поиска",

"type":"input\_value",

"value":"телефон",

"locator":"//input[@name='search']",

"timeout":1

},

{

"description":"ШАГ: Нажал на кнопку Найти",

"type":"click\_element",

"value":"",

"locator":"//button[@class='button button\_color\_green button\_size\_medium search-form\_\_submit']",

"timeout":1

},

{

"description":"ШАГ: Ожидание завершения поиска",

"type":"wait\_element",

"value":"",

"locator":"//h1[contains(text(), 'Мобильные телефоны')]",

"timeout":15

},

{

"description":"ШАГ: Нажал на кнопку Найти",

"type":"click\_element",

"value":"",

"locator":"//button[@class='button button\_color\_green button\_size\_medium search-form\_\_submit']",

"timeout":1

},

{

"description":"ШАГ: Ожидание 5 секунд",

"type":"sleep",

"value":"",

"locator":"",

"timeout":5

},

{

"description":"Событие GA",

"type":"test\_defaults\_ga",

"value":"",

"locator":"",

"timeout":5

}

]

}

*Created with the Personal Edition of HelpNDoc:* [*Free PDF documentation generator*](https://www.helpndoc.com)

**Опциональная проверка событий**

**Опциональная проверка событий**

В отличии от [базовой проверки событий](#_topic_Newtopic19) опциональная проверка использует данные прописанные в таблице "Шаги"

Бывают ситуации когда невозможно использовать данные из таблицы "Данные" в таком случае чтобы выполнить опциональную проверку GA и YM событий описываются шаги специального [типа действия](#_topic_Newtopic9) и с указанием протокола.

Действие: test\_optionally\_ga | GA протокол: google-analytics.com/collect

pageview&\_s=1&dl=https://test.ru

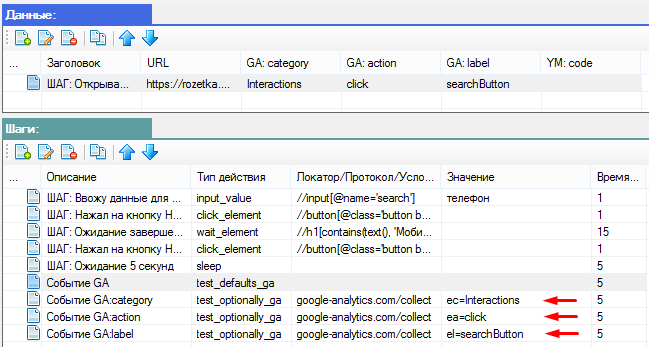
ec=test\_ga\_category

ea=test\_ga\_action

el=test\_ga\_label

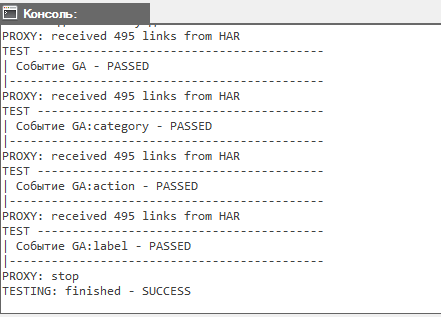
Действие: test\_optionally\_ym | YM протокол: mc.yandex.ru/watch

test\_ym\_code



Как мы можем видет из примера выполняется базовая проверка событий по таблице "Данные", а ниже та же самая проверка выполняется в шагах с использованием действия test\_optionally\_ga и/или test\_optionally\_ym.

В результате мы видим что все проверки успешно были выполнены.



Исходный код теста:

{

"description":"Тестирование событий поиска",

"port":9091,

"arguments":[

"--ignore-certificate-errors"

],

"har":"test.har",

"timeout":60,

"data":[

{

"title":"ШАГ: Открываю страницу https://rozetka.com.ua/",

"url":"https://rozetka.com.ua/",

"ga\_category":"Interactions",

"ga\_action":"click",

"ga\_label":"searchButton",

"ym\_code":""

}

],

"steps":[

{

"description":"ШАГ: Ввожу данные для поиска",

"type":"input\_value",

"value":"телефон",

"locator":"//input[@name='search']",

"timeout":1

},

{

"description":"ШАГ: Нажал на кнопку Найти",

"type":"click\_element",

"value":"",

"locator":"//button[@class='button button\_color\_green button\_size\_medium search-form\_\_submit']",

"timeout":1

},

{

"description":"ШАГ: Ожидание завершения поиска",

"type":"wait\_element",

"value":"",

"locator":"//h1[contains(text(), 'Мобильные телефоны')]",

"timeout":15

},

{

"description":"ШАГ: Нажал на кнопку Найти",

"type":"click\_element",

"value":"",

"locator":"//button[@class='button button\_color\_green button\_size\_medium search-form\_\_submit']",

"timeout":1

},

{

"description":"ШАГ: Ожидание 5 секунд",

"type":"sleep",

"value":"",

"locator":"",

"timeout":5

},

{

"description":"Событие GA",

"type":"test\_defaults\_ga",

"value":"",

"locator":"",

"timeout":5

},

{

"description":"Событие GA:category",

"type":"test\_optionally\_ga",

"value":"ec=Interactions",

"locator":"google-analytics.com/collect",

"timeout":5

},

{

"description":"Событие GA:action",

"type":"test\_optionally\_ga",

"value":"ea=click",

"locator":"google-analytics.com/collect",

"timeout":5

},

{

"description":"Событие GA:label",

"type":"test\_optionally\_ga",

"value":"el=searchButton",

"locator":"google-analytics.com/collect",

"timeout":5

}

]

}

*Created with the Personal Edition of HelpNDoc:* [*Free CHM Help documentation generator*](https://www.helpndoc.com)

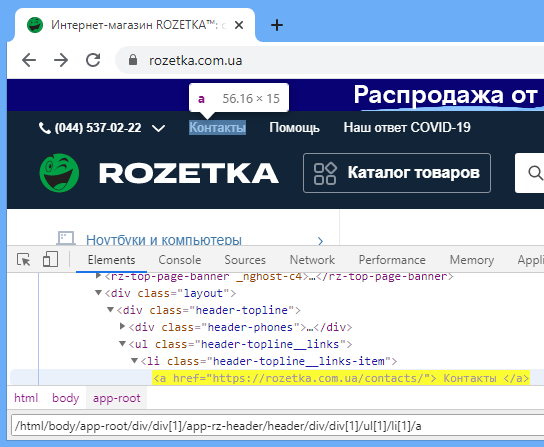
**Получение данных из элементов**

**Получение данных из элементов**

Иногда бывает необходимо получить какое-то значение из элемента на странице чтобы потом это значение использовать в другом элементе страницы.

Допустим у нас на странице есть элемент текст которого "Контакты". Наша задача получить этот текст из этемента и вставить его в поле поиска.

Сначала сформируем локатор (xpath) этого элемента: /html/body/app-root/div/div[1]/app-rz-header/header/div/div[1]/ul[1]/li[1]/a



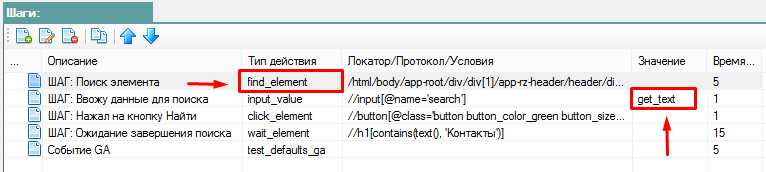
Дальше опишим шаги.

Сначала используем [действие](#_topic_Newtopic9) find\_element для того чтобы найти элемент.

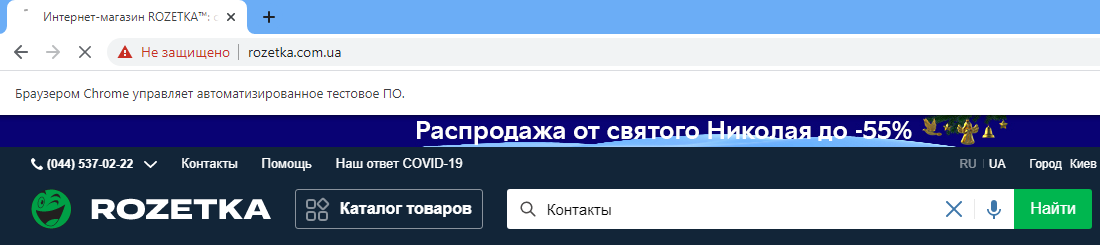
После того как элемент найден он будет помещен в глобальную переменную доступ к которой есть у следующих методов:

* + get\_text
  + get\_attribute\_id
  + get\_attribute\_name
  + get\_attribute\_class
  + get\_attribute\_value
  + get\_attribute\_href
  + get\_attribute\_ в данном случае вы можете дописать имя атрибута самостоятельно после подчеркивания

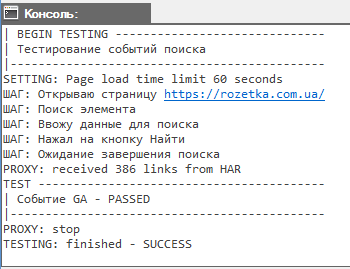
В нашем случае мы вослользуемся методом get\_text для того чтобы встатить текст в поле поиска input.



Запустим тест на выполнение.



Как мы видим текст "Контакты" был вставлен в поиск.



В консоли мы видим шаги выполнения действий и успешное завершение теста.

Исходный код теста:

{

"description":"Тестирование событий поиска",

"port":9091,

"arguments":[

"--ignore-certificate-errors"

],

"har":"test.har",

"timeout":60,

"data":[

{

"title":"ШАГ: Открываю страницу https://rozetka.com.ua/",

"url":"https://rozetka.com.ua/",

"ga\_category":"Interactions",

"ga\_action":"click",

"ga\_label":"searchButton",

"ym\_code":""

}

],

"steps":[

{

"description":"ШАГ: Поиск элемента",

"type":"find\_element",

"value":"",

"locator":"/html/body/app-root/div/div[1]/app-rz-header/header/div/div[1]/ul[1]/li[1]/a",

"timeout":5

},

{

"description":"ШАГ: Ввожу данные для поиска",

"type":"input\_value",

"value":"get\_text",

"locator":"//input[@name='search']",

"timeout":1

},

{

"description":"ШАГ: Нажал на кнопку Найти",

"type":"click\_element",

"value":"",

"locator":"//button[@class='button button\_color\_green button\_size\_medium search-form\_\_submit']",

"timeout":1

},

{

"description":"ШАГ: Ожидание завершения поиска",

"type":"wait\_element",

"value":"",

"locator":"//h1[contains(text(), 'Контакты')]",

"timeout":15

},

{

"description":"Событие GA",

"type":"test\_defaults\_ga",

"value":"",

"locator":"",

"timeout":5

}

]

}

*Created with the Personal Edition of HelpNDoc:* [*Easily create Qt Help files*](https://www.helpndoc.com/feature-tour)

**Использование условий**

**Использование условий**

Чтобы показать использование условий возьмем предудыщий пример [получения данных из элемента](#_topic_Newtopic21) и доработаем его.

Для программирования условий нужно использовать следующие [типы действий](#_topic_Newtopic9):

* if\_get\_text
* if\_get\_attribute\_
* else\_if\_get\_text
* else\_if\_get\_attribute\_
* else
* end\_if

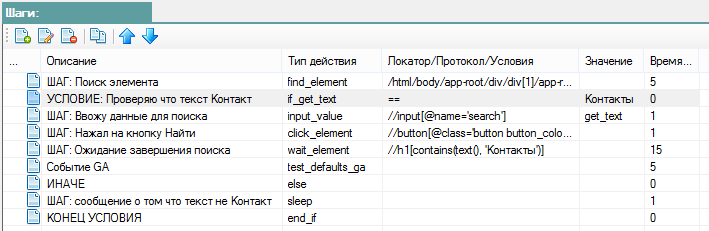
При использовании if\_get\_attribute\_ и else\_if\_get\_attribute\_ нужно дописывать имя атрибута после подчеркивания (например: if\_get\_attribute\_id и else\_if\_get\_attribute\_id)

Условия начинаются действием if\_get\_text или if\_get\_attribute\_

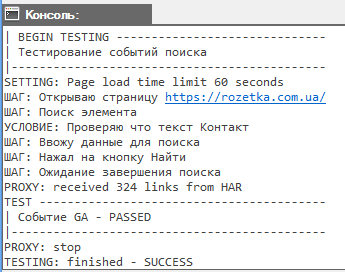
Дальше идут шаги действий или проверки.

Если нужно выполнять какие-то шаги в случае невыполнения условия используются действия else\_if\_get\_text , else\_if\_get\_attribute\_ и else

Завершается условие действием end\_if



Запустим автотест и убедимся что условие будет выполнено.



Полученный текст из элемента соответствует заданному тексту в условие и в результате тест был проведен.

Исходный код теста:

{

"description":"Тестирование событий поиска",

"port":9091,

"arguments":[

"--ignore-certificate-errors"

],

"har":"test.har",

"timeout":60,

"data":[

{

"title":"ШАГ: Открываю страницу https://rozetka.com.ua/",

"url":"https://rozetka.com.ua/",

"ga\_category":"Interactions",

"ga\_action":"click",

"ga\_label":"searchButton",

"ym\_code":""

}

],

"steps":[

{

"description":"ШАГ: Поиск элемента",

"type":"find\_element",

"value":"",

"locator":"/html/body/app-root/div/div[1]/app-rz-header/header/div/div[1]/ul[1]/li[1]/a",

"timeout":5

},

{

"description":"УСЛОВИЕ: Проверяю что текст Контакты",

"type":"if\_get\_text",

"value":"Контакты",

"locator":"==",

"timeout":0

},

{

"description":"ШАГ: Ввожу данные для поиска",

"type":"input\_value",

"value":"get\_text",

"locator":"//input[@name='search']",

"timeout":1

},

{

"description":"ШАГ: Нажал на кнопку Найти",

"type":"click\_element",

"value":"",

"locator":"//button[@class='button button\_color\_green button\_size\_medium search-form\_\_submit']",

"timeout":1

},

{

"description":"ШАГ: Ожидание завершения поиска",

"type":"wait\_element",

"value":"",

"locator":"//h1[contains(text(), 'Контакты')]",

"timeout":15

},

{

"description":"Событие GA",

"type":"test\_defaults\_ga",

"value":"",

"locator":"",

"timeout":5

},

{

"description":"ИНАЧЕ",

"type":"else",

"value":"",

"locator":"",

"timeout":0

},

{

"description":"ШАГ: сообщение о том что текст не Контакт",

"type":"sleep",

"value":"",

"locator":"",

"timeout":1

},

{

"description":"КОНЕЦ УСЛОВИЯ",

"type":"end\_if",

"value":"",

"locator":"",

"timeout":0

}

]

}

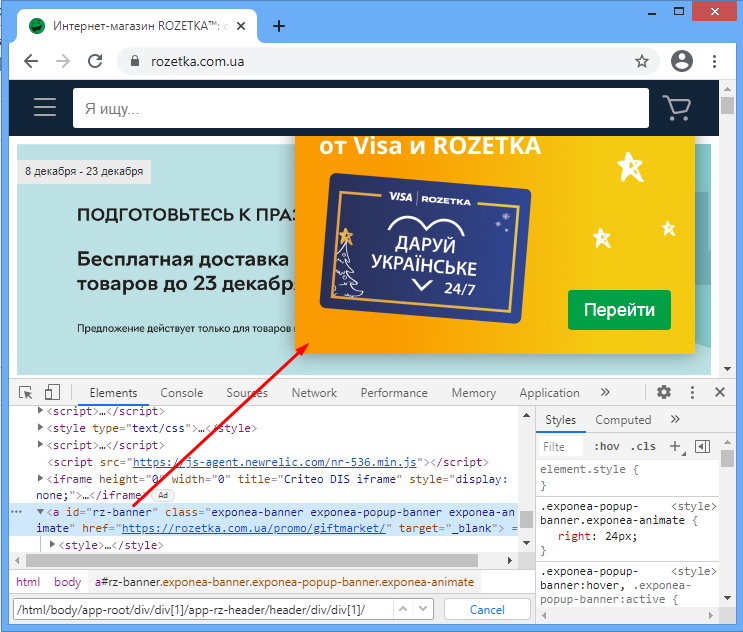
*Created with the Personal Edition of HelpNDoc:* [*Qt Help documentation made easy*](https://www.helpndoc.com/feature-tour/create-help-files-for-the-qt-help-framework)

**Применение JavaScript**

**Применение JavaScript**

Иногда бывает необходимо выполнить JavaScript код который поможет в процессе тестирования.

Допустим на сайте присутствует всплывающий баннер который может блокировать ход тестирования.



Чтобы выполнить JavaScript код воспользуйтесь специальным [действием](#_topic_Newtopic9) execute\_js

Добавим шаг выполняющий следующий JavaScript под:

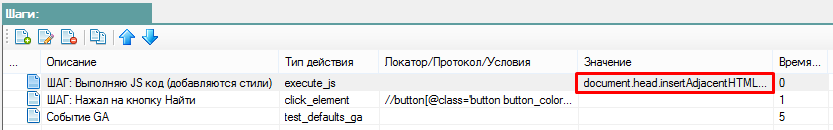
document.head.insertAdjacentHTML(\"beforeend\", '<style type=\"text/css\">body{overflow:auto!important;}#rz-banner{display:none!important}</style>');

или использовать встроенный на странице jQuery

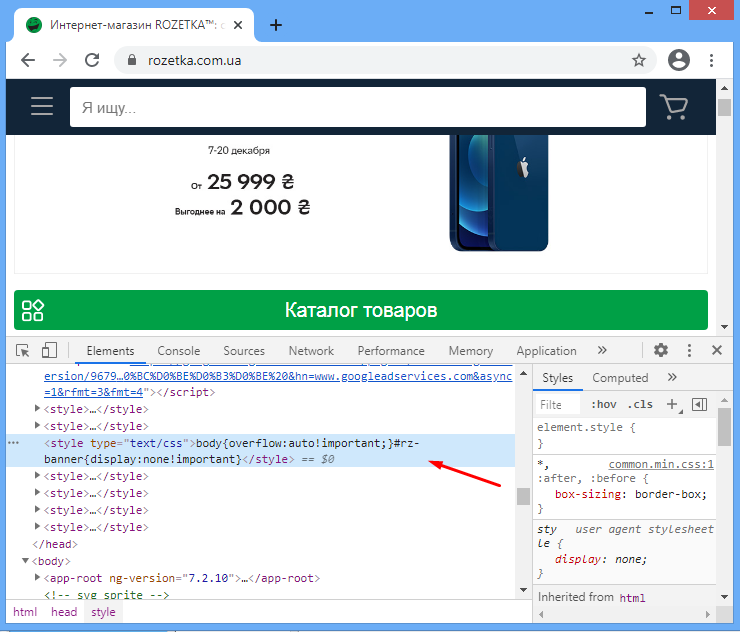
$(\"body\").find(\"#rz-banner\").css({display:\"none!important\"});

$(\"head\").prepend('<style type=\"text/css\">body{overflow:auto!important;}#rz-banner{display:none!important}</style>');

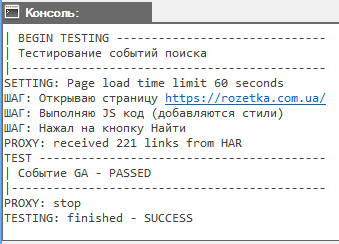
Обязательно нужно экранировать двойные кавычки чтобы тест корректно сохранился в json формате.



Запустим тест. Во время выполнения на странице будет добавлен слить которые удалит баннер и не даст ему появляться.



В консоли видим строку выполнения JavaScript кода и успешный результат проверки.



Исходный код теста:

{

"description":"Тестирование событий поиска",

"port":9091,

"arguments":[

"--ignore-certificate-errors"

],

"har":"test.har",

"timeout":60,

"data":[

{

"title":"ШАГ: Открываю страницу https://rozetka.com.ua/",

"url":"https://rozetka.com.ua/",

"ga\_category":"Interactions",

"ga\_action":"click",

"ga\_label":"searchButton",

"ym\_code":""

}

],

"steps":[

{

"description":"ШАГ: Выполняю JS код (добавляются стили)",

"type":"execute\_js",

"value":"document.head.insertAdjacentHTML(\"beforeend\", '<style type=\"text/css\">body{overflow:auto!important;}#rz-banner{display:none!important}</style>');",

"locator":"",

"timeout":0

},

{

"description":"ШАГ: Нажал на кнопку Найти",

"type":"click\_element",

"value":"",

"locator":"//button[@class='button button\_color\_green button\_size\_medium search-form\_\_submit']",

"timeout":1

},

{

"description":"Событие GA",

"type":"test\_defaults\_ga",

"value":"",

"locator":"",

"timeout":5

}

]

}

*Created with the Personal Edition of HelpNDoc:* [*Produce electronic books easily*](https://www.helpndoc.com/create-epub-ebooks)