

# Hat

## Table of contents

---

Русская версия .....	10
Описание .....	10
Быстрый старт .....	10
Системные требования .....	10
Установка и запуск .....	12
Создание простого проекта .....	15
Создание проекта совместимого с Visual Studio .....	18
Запуск автотеста и его результат .....	22
Кэш браузера .....	25
Очистка .....	25
Интерфейс .....	26
Основное меню .....	26
Панель инструментов .....	28
Окно Проект - вкладка Проводник .....	28
Окно Проект - вкладка Тест .....	32
Окно Проект - вкладка Библиотеки .....	33
Окно Проект - вкладка Почта .....	34
Окно система .....	35
Отчеты .....	36
Отчет работы автотеста .....	36
Лог файл .....	40
Почта .....	40
Отправить отчет о провале автотеста на почту .....	40
Консольный режим запуска (Jenkins) .....	42
Запуск из командной строки .....	42
Jenkins .....	45
1. Настройка агента .....	46
2. Настройка поддержки русского языка .....	52
3. Подключение автотеста к ноде. ....	54
Текстовый редактор кода .....	56
Встроенный редактор кода на C# .....	57
Использование Visual Studio для редактирования кода .....	60
Структура автотеста .....	64
Простой автотест .....	64
Паттерн PageObjects и StepObjects .....	67
Создание нового автотеста и паттернов .....	70
Встроенный фреймворк HatFramework .....	74
Класс: Tester .....	74
Конструктор .....	74
Tester .....	74
Константы .....	75
BY_CSS .....	75
BY_XPATH .....	75
IMAGE_STATUS_PROCESS .....	75
IMAGE_STATUS_PASSED .....	75
IMAGE_STATUS_FAILED .....	76
IMAGE_STATUS_MESSAGE .....	76

IMAGE_STATUS_WARNING .....	76
IMAGE_STATUS_DEBUG .....	76
COMPLETED .....	77
FAILED .....	77
PASSED .....	77
PROCESS .....	77
STOPPED .....	77
WARNING .....	78
DEBUG .....	78
DEFAULT .....	78
UTF8 .....	78
UTF8BOM .....	79
WINDOWS1251 .....	79
Переменные .....	79
BrowserView .....	79
BrowserWindow .....	80
Локаторы .....	80
Locator .....	80
AddLocator .....	80
GetLocators .....	81
GetLocator .....	81
GetLocatorValue .....	81
GetCountLocators .....	82
ClearLocators .....	82
RemoveLocator .....	82
Методы для работы с браузером .....	82
BrowserBasicAuthenticationAsync .....	82
BrowserClearNetworkAsync .....	83
BrowserCloseAsync .....	83
BrowserEnableSendMailAsync .....	83
BrowserFullScreenAsync .....	84
BrowserScreenshotAsync .....	84
BrowserSizeAsync .....	84
BrowserGetUserAgentAsync .....	85
BrowserSetUserAgentAsync .....	85
BrowserGetErrorsAsync .....	85
BrowserGetNetworkAsync .....	85
BrowserGoBackAsync .....	86
BrowserGoForwardAsync .....	86
BrowserPageReloadAsync .....	87
Методы для вывода сообщений .....	87
ConsoleMsg .....	88
ConsoleMsgError .....	88
ClearMessage .....	88
Description .....	88
DisableDebugInReport .....	89
SendMessage .....	89
SendMessageDebug .....	90
Методы для подготовки и завершения тестирования .....	91
TestBeginAsync .....	91

TestEndAsync .....	91
TestStopAsync .....	91
GetTestResult .....	91
DefineTestStop .....	92
Методы для выполнения действий .....	92
Атрибуты .....	92
GetAttributeFromElementAsync .....	93
GetAttributeFromElementByClassAsync .....	94
GetAttributeFromElementByIdAsync .....	95
GetAttributeFromElementByNameAsync .....	96
GetAttributeFromElementByTagAsync .....	97
GetAttributeFromElementsAsync .....	98
GetAttributeFromElementsByClassAsync .....	99
GetAttributeFromElementsByNameAsync .....	100
GetAttributeFromElementsByTagAsync .....	101
SetAttributeInElementAsync .....	102
SetAttributeInElementByClassAsync .....	103
SetAttributeInElementByIdAsync .....	104
SetAttributeInElementByNameAsync .....	105
SetAttributeInElementByTagAsync .....	106
SetAttributeInElementsAsync .....	107
SetAttributeInElementsByClassAsync .....	108
SetAttributeInElementsByNameAsync .....	109
SetAttributeInElementsByTagAsync .....	110
Значение .....	110
GetValueFromElementAsync .....	111
GetValueFromElementByClassAsync .....	112
GetValueFromElementByIdAsync .....	113
GetValueFromElementByNameAsync .....	114
GetValueFromElementByTagAsync .....	115
SetValueInElementAsync .....	116
SetValueInElementByClassAsync .....	117
SetValueInElementByIdAsync .....	118
SetValueInElementByNameAsync .....	119
SetValueInElementByTagAsync .....	120
Нажатие .....	120
ClickElementAsync .....	121
ClickElementByClassAsync .....	122
ClickElementByIdAsync .....	123
ClickElementByNameAsync .....	124
ClickElementByTagAsync .....	125
FocusElementAsync .....	126
IsClickableElementAsync .....	127
ScrollToElementAsync .....	128
Объекты .....	128
GetElementAsync .....	129
GetFrameAsync .....	131
GetCountElementsAsync .....	133
GetCountElementsByClassAsync .....	134
GetCountElementsByNameAsync .....	135

GetCountElementsByTagAsync .....	136
GetHtmlFromElementAsync .....	137
GetHtmlFromElementByClassAsync .....	138
GetHtmlFromElementByIdAsync .....	139
GetHtmlFromElementByNameAsync .....	140
GetHtmlFromElementByTagAsync .....	141
IsVisibleElementAsync .....	142
MakeElementVisibleAsync .....	143
SetHtmlInElementAsync .....	144
SetHtmlInElementByClassAsync .....	145
SetHtmlInElementByIdAsync .....	146
SetHtmlInElementByNameAsync .....	147
SetHtmlInElementByTagAsync .....	148
Ожидание .....	148
WaitAsync .....	149
WaitElementInDomAsync .....	150
WaitElementNotDomAsync .....	151
WaitNotVisibleElementAsync .....	152
WaitNotVisibleElementByClassAsync .....	153
WaitNotVisibleElementByIdAsync .....	154
WaitNotVisibleElementByNameAsync .....	155
WaitNotVisibleElementByTagAsync .....	156
WaitVisibleElementAsync .....	157
WaitVisibleElementByClassAsync .....	158
WaitVisibleElementByIdAsync .....	159
WaitVisibleElementByNameAsync .....	160
WaitVisibleElementByTagAsync .....	161
Поиск .....	161
FindElementAsync .....	162
FindElementByClassAsync .....	163
FindElementByIdAsync .....	164
FindElementByNameAsync .....	165
FindElementByTagAsync .....	166
FindVisibleElementAsync .....	167
FindVisibleElementByClassAsync .....	168
FindVisibleElementByIdAsync .....	169
FindVisibleElementByNameAsync .....	170
FindVisibleElementByTagAsync .....	171
Стили .....	171
GetStyleFromElementAsync .....	172
GetStyleFromElementByClassAsync .....	173
GetStyleFromElementByIdAsync .....	174
GetStyleFromElementByNameAsync .....	175
GetStyleFromElementByTagAsync .....	176
SetStyleInElementAsync .....	177
SetStyleInElementByClassAsync .....	178
SetStyleInElementByIdAsync .....	179
SetStyleInElementByNameAsync .....	180
SetStyleInElementByTagAsync .....	181
Страница .....	181

GetCookiesAsync .....	182
GetListRedirectUrlAsync .....	183
GetTitleAsync .....	184
GetUrlAsync .....	185
GetUrlResponseAsync .....	186
GoToUrlAsync .....	187
GoToUrlBaseAuthAsync .....	188
LoadPageAsync .....	189
Текст .....	189
GetTextFromElementAsync .....	190
GetTextFromElementByClassAsync .....	191
GetTextFromElementByIdAsync .....	192
GetTextFromElementByNameAsync .....	193
GetTextFromElementByTagAsync .....	194
SetTextInElementAsync .....	195
SetTextInElementByClassAsync .....	196
SetTextInElementByIdAsync .....	197
SetTextInElementByNameAsync .....	198
SetTextInElementByTagAsync .....	199
Методы для выполнения JavaScript .....	199
ExecuteJavaScriptAsync .....	199
Методы для выполнения Rest запросов .....	199
RestGetAsync .....	199
RestGetBasicAuthAsync .....	200
RestGetStatusCodeAsync .....	201
RestPostAsync .....	201
Методы для замера затраченного времени .....	202
TimerStart .....	202
TimerStop .....	202
Методы для отправки email и message .....	203
SendMsgToMailAsync .....	203
SendMsgToTelegramAsync .....	203
Методы для проверки результата .....	204
AssertEqualsAsync .....	204
AssertNotEqualsAsync .....	204
AssertTrueAsync .....	205
AssertFalseAsync .....	205
AssertNotNullAsync .....	205
AssertNullAsync .....	205
AssertNoErrorsAsync .....	206
AssertNetworkEventsAsync .....	206
Методы для работы с файлами .....	206
FileDownloadAsync .....	206
FileGetHashMD5Async .....	207
FileReadAsync .....	207
FileWriteAsync .....	207
Методы для разных задач .....	208
CreateHashMD5FromTextAsync .....	208
GetProjectPath .....	208
Класс: HTML_Element .....	208

Конструктор .....	208
HTMLElement .....	208
Константы .....	209
BY_INDEX .....	209
BY_TEXT .....	209
BY_VALUE .....	209
Переменные .....	209
Id .....	209
Name .....	210
Class .....	210
Type .....	210
Методы .....	211
ClickAsync .....	211
ClickMouseAsync .....	211
FocusAsync .....	211
GetAttributeAsync .....	211
GetHtmlAsync .....	212
GetLocatorAsync .....	212
GetOptionAsync .....	212
GetStyleAsync .....	213
GetTextAsync .....	213
GetValueAsync .....	213
IsClickableAsync .....	213
MakeVisibleAsync .....	214
ScrollToAsync .....	214
SelectOptionAsync .....	214
SetAttributeAsync .....	215
SetHtmlAsync .....	215
SetStyleAsync .....	215
SetTextAsync .....	216
SetValueAsync .....	216
WaitNotVisibleAsync .....	216
WaitVisibleAsync .....	216
Класс: FRAMEElement .....	217
Конструктор .....	217
FRAMEElement .....	217
Константы .....	217
BY_INDEX .....	217
BY_TEXT .....	217
BY_VALUE .....	218
Переменные .....	218
Name .....	218
Index .....	218
Index .....	218
Методы .....	219
ClickElementAsync .....	219
FindElementAsync .....	219
FindVisibleElementAsync .....	219
GetAttributeFromElementAsync .....	220
GetAttributeFromElementsAsync .....	220

GetCountElementsAsync .....	221
GetHtmlFromElementAsync .....	221
GetOptionAsync .....	221
GetStyleFromElementAsync .....	222
GetTextFromElementAsync .....	222
GetTitleAsync .....	223
GetUrlAsync .....	223
GetValueFromElementAsync .....	223
IsClickableElementAsync .....	223
IsVisibleElementAsync .....	224
MakeVisibleAsync .....	224
ScrollToElementAsync .....	224
SelectOptionAsync .....	225
SetAttributeInElementAsync .....	225
SetAttributeInElementsAsync .....	226
SetHtmlInElementAsync .....	226
SetStyleInElementAsync .....	227
SetTextInElementAsync .....	227
SetValueInElementAsync .....	227
WaitNotVisibleElementAsync .....	228
WaitVisibleElementAsync .....	228
Плагины .....	228
Плагин HatPluginMySql .....	228
Установка плагина .....	228
Подключение плагина к проекту .....	231
Пример автотеста .....	232
Класс: TesterMySql .....	237
Конструктор .....	237
TesterMySql .....	238
Методы .....	238
ConnectionOpenAsync .....	239
ConnectionCloseAsync .....	240
GetCountEntriesAsync .....	241
GetEntriesAsync .....	242
GetEntriesFromTableAsync .....	243
GetDataTableAsync .....	244
SetEntryAsync .....	245
EditEntryAsync .....	246
RemoveEntryAsync .....	247
FindEntryAsync .....	248
AssertHaveInTableAsync .....	249
AssertDontHaveInTableAsync .....	250
Практические примеры .....	250
Группа автотестов .....	250
Выполнение JavaScript кода .....	257
Обработка Json данных с помощью Newtonsoft .....	260
Обработка XML данных для проверки карты сайта Sitemap .....	262
Диалоговые окна alert, prompt и confirm .....	266
Базовая авторизация .....	270
Перехват событий google analytics и yandex metrika .....	272

Версия 1.3 (не актуальна) .....	275
Описание .....	275
SendMessage [изменён начиная с версии 1.3.0] .....	276
SendMessageDebug [изменён начиная с версии 1.3.0] .....	277
EditMessage [удалён начиная с версии 1.3.0] .....	278
EditMessageDebug [удалён начиная с версии 1.3.0].....	279
DefineTestStop [изменён начиная с версии 1.3.0].....	280

## Русская версия

---

Created with the Personal Edition of HelpNDoc: [Free EPub and documentation generator](#)

---

### Описание

# Hat

---

Selenium WebDriver. Hat HatFramework C# Visual Studio. Windows : Jenkins, TeamCity, GitLab CI/CD. : <https://somovstudio.github.io/ru/> GitFlic: <https://gitflic.ru/project/somovstudio/hat/> 15.04.2026) 1.5.1.4 (

---

Created with the Personal Edition of HelpNDoc: [Easily create Help documents](#)

---

### Быстрый старт

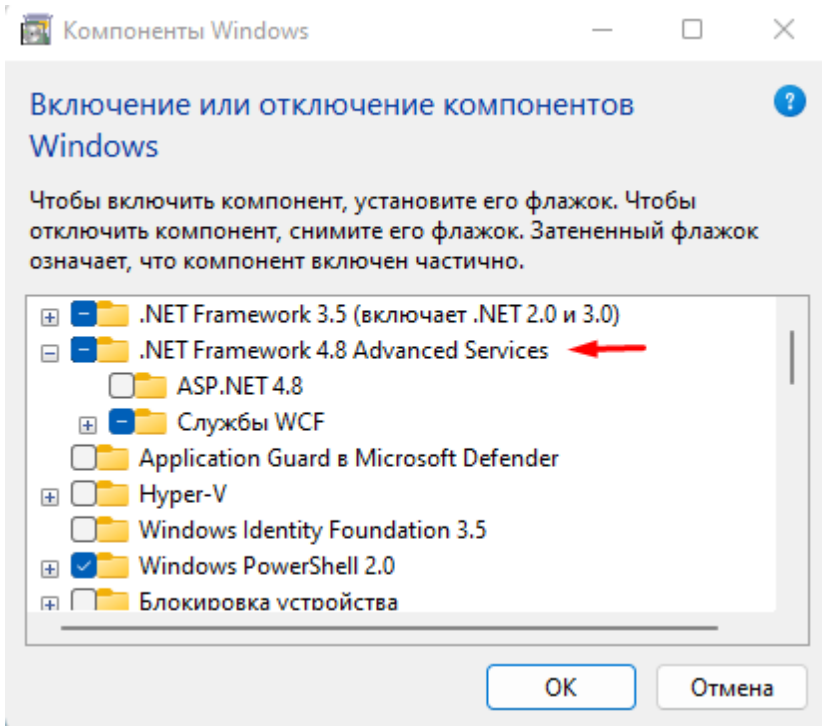
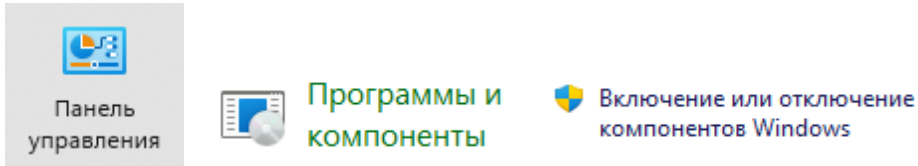
---

Created with the Personal Edition of HelpNDoc: [Easily create PDF Help documents](#)

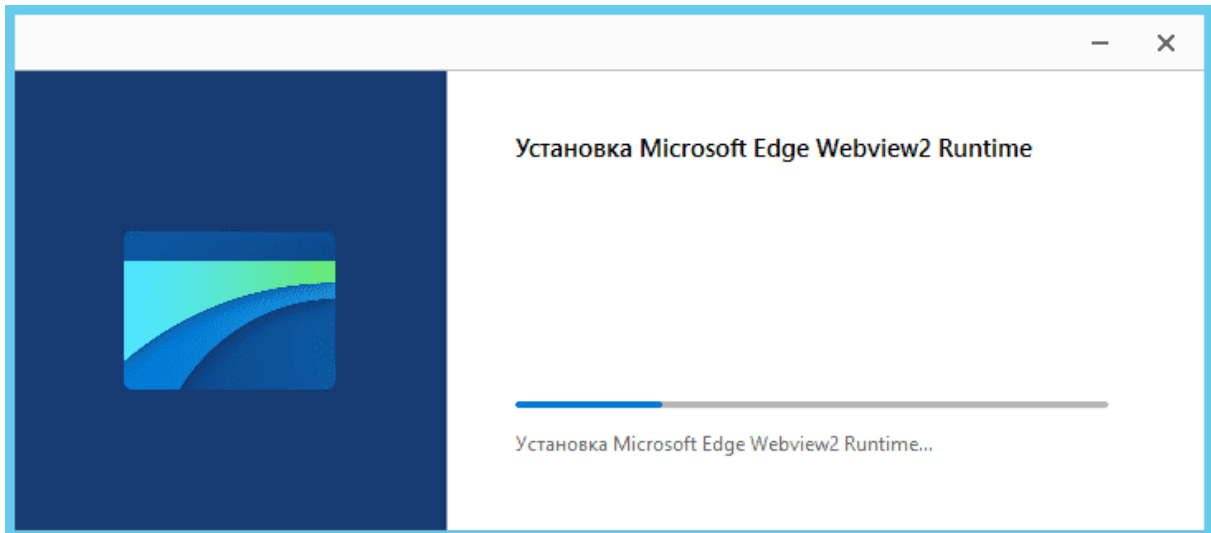
---

### Системные требования

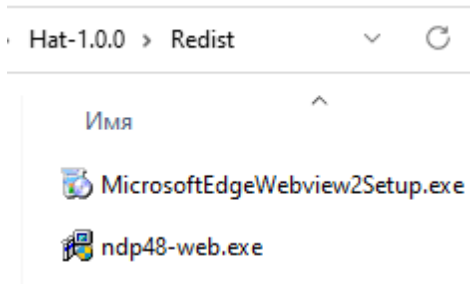
1. Windows 8, 8.1, 10, 11
2. Microsoft .NET Framework 4.8 (с [.NET Framework Runtime](#))



3. Microsoft Edge WebView2 (с [WebView2 Runtime](#))



Microsoft .NET Framework 4.8 : Redist  
Microsoft Edge WebView2



Created with the Personal Edition of HelpNDoc: [Easily create EPub books](#)

## Установка и запуск

### Hat

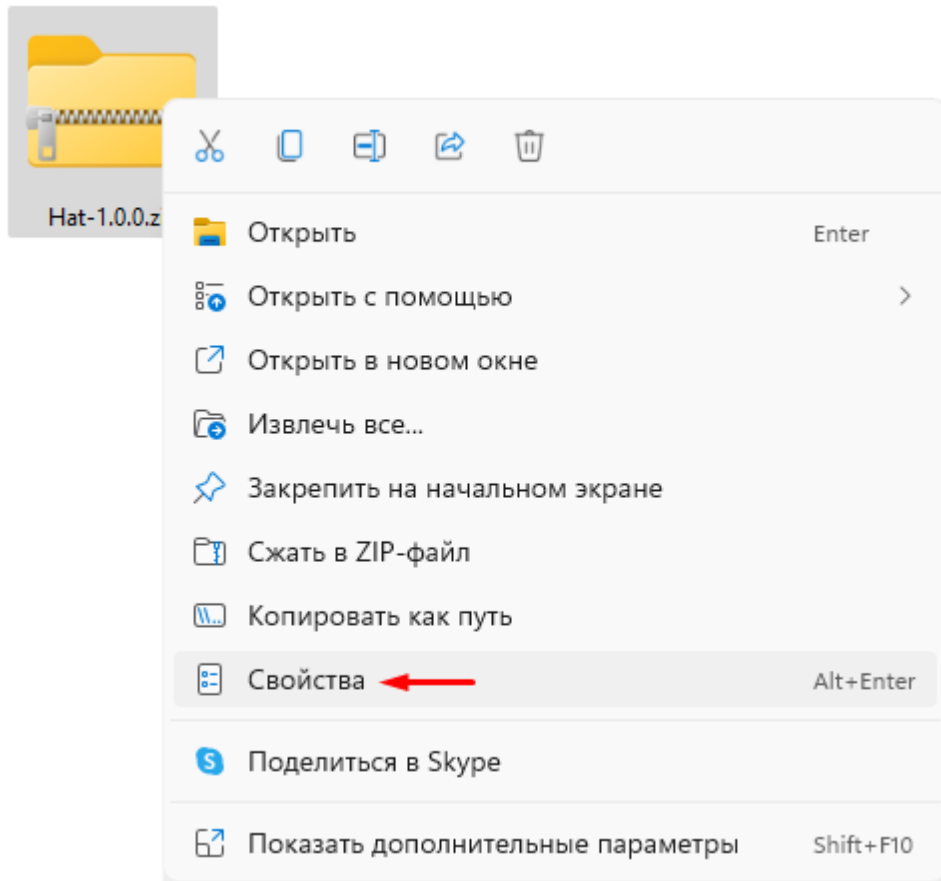
1. <https://github.com/SomovStudio/Hat/>

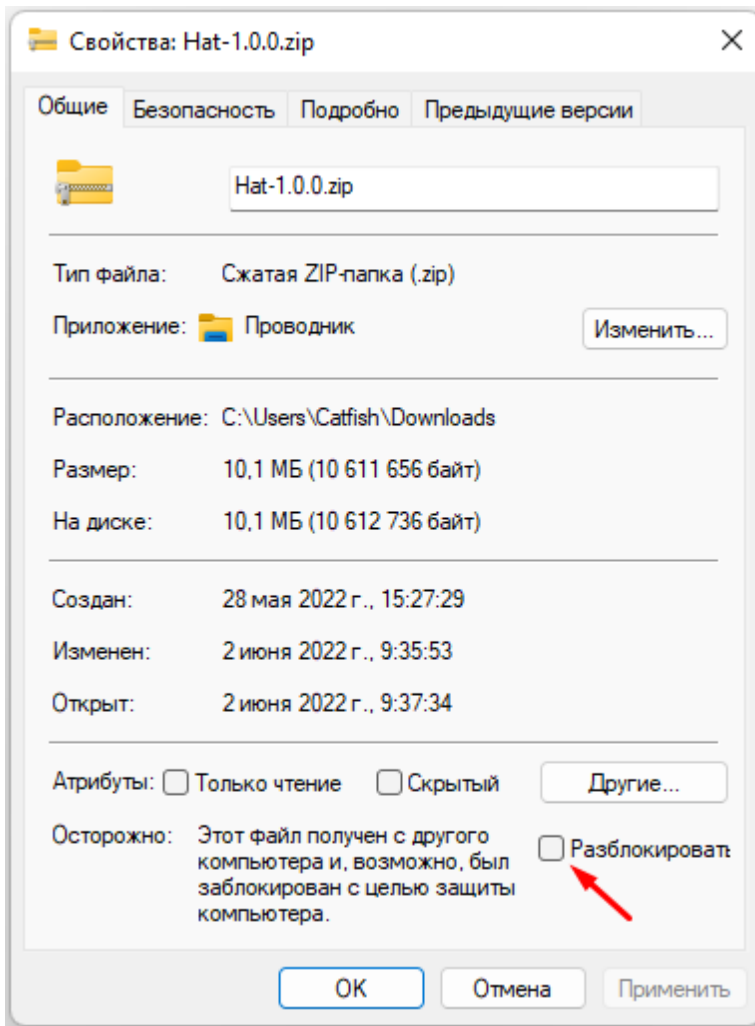
GitHub:



Hat-1.0.0.zip

” ”  
.

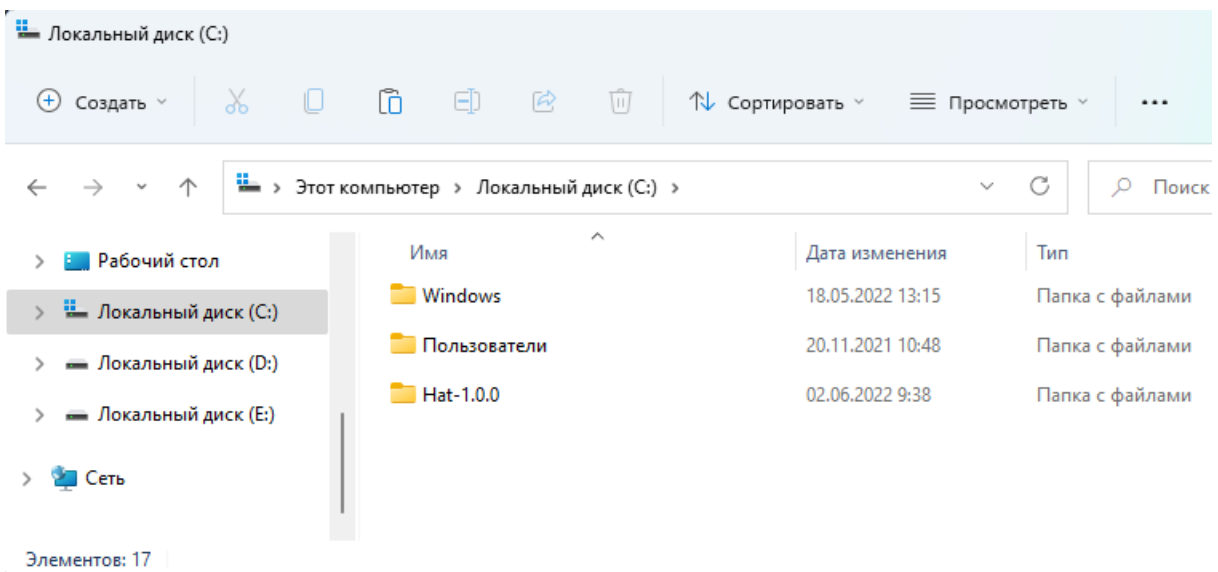


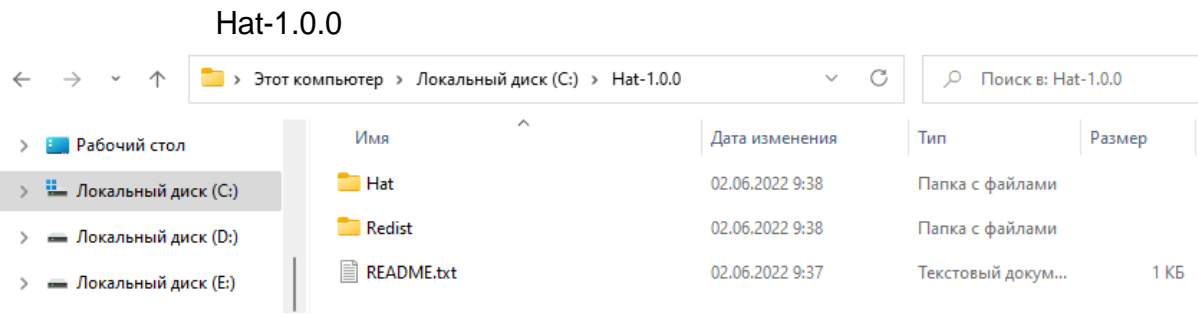


2.

:

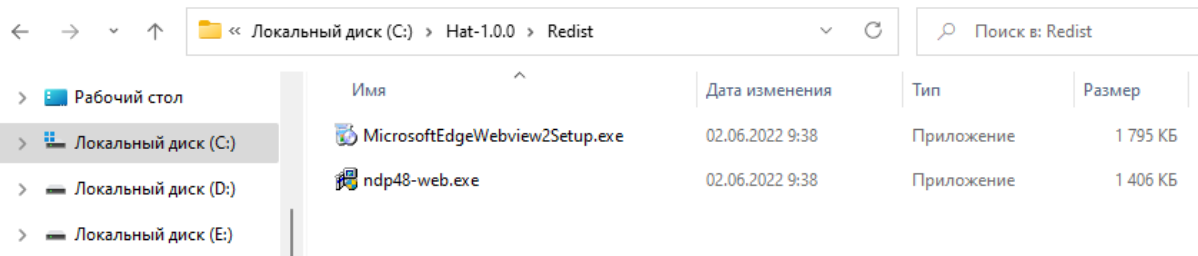
Program Files



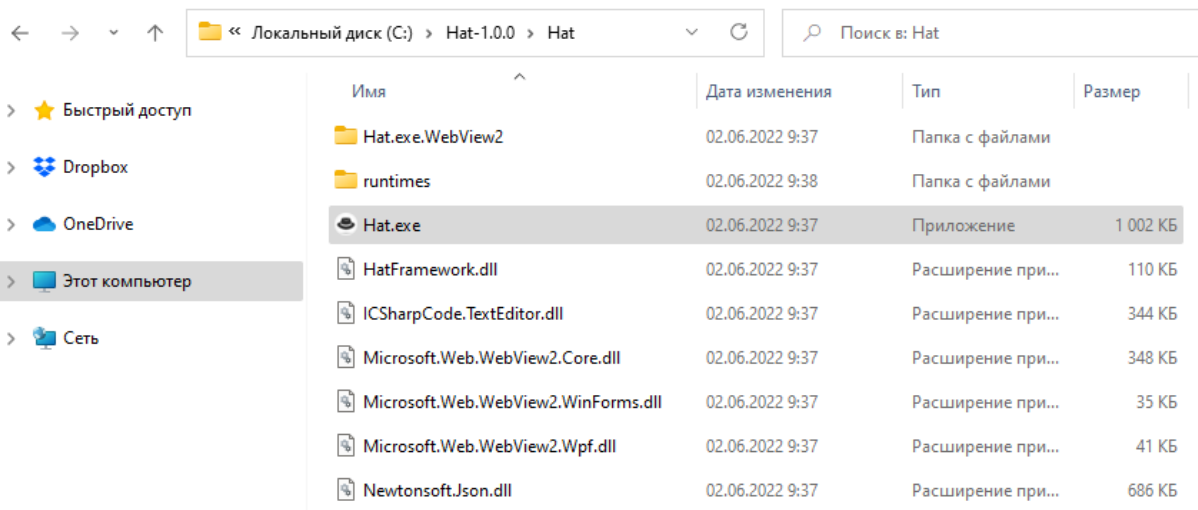


### 3. Redist Microsoft Edge WebView2

### Microsoft .NET Framework 4.8



### 4. Hat

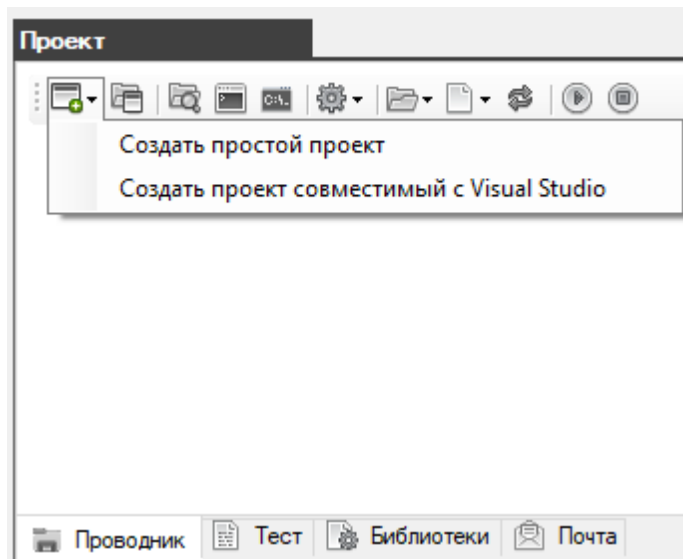
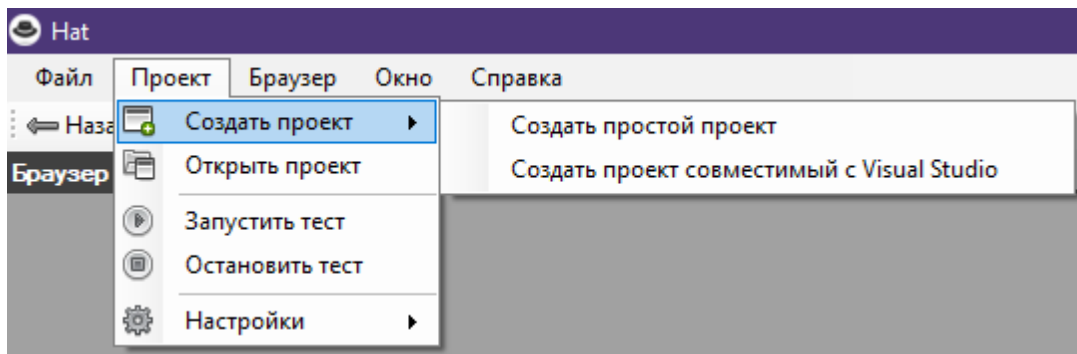


### Hat.exe

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

## Создание простого проекта

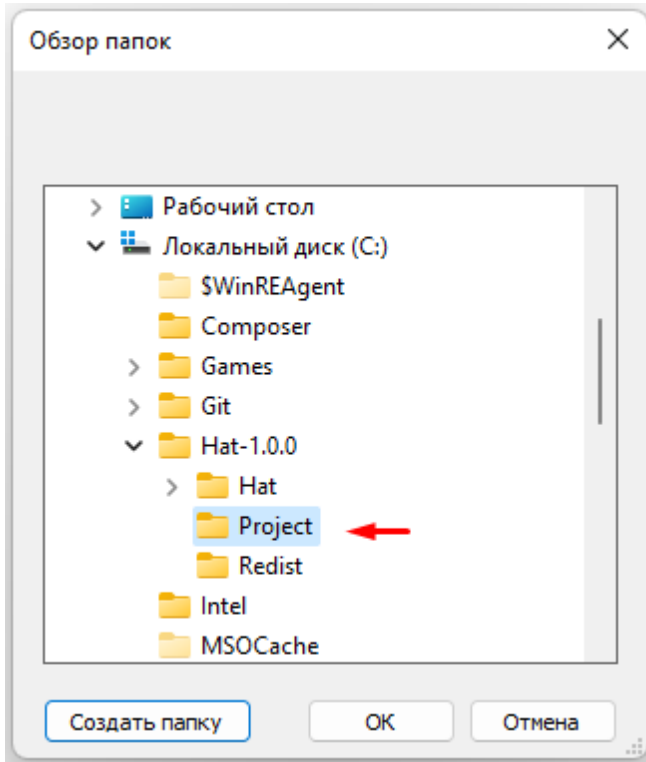
1. " " " " " "



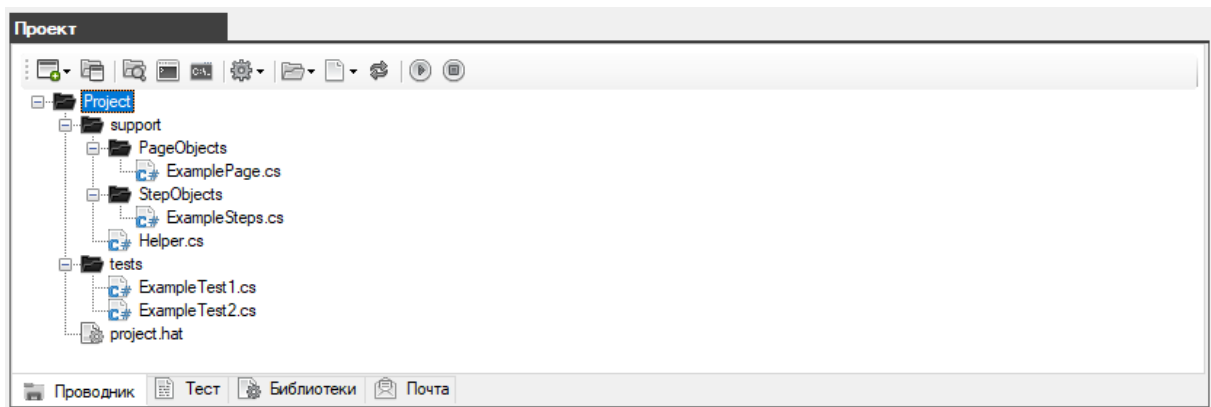
2.

Project  
OK

C:\Hat-1.0.0\



3. " " " "

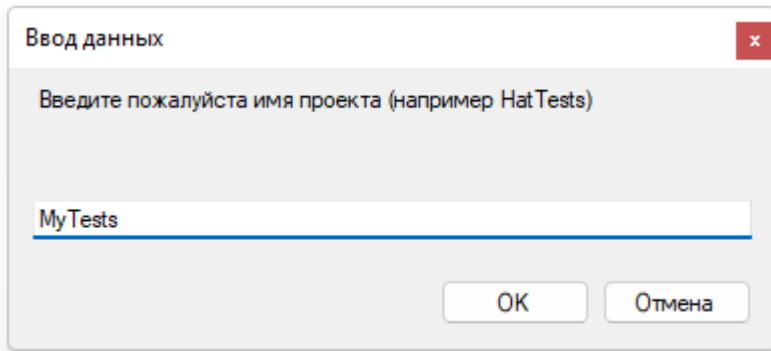


4. ExampleTest1.cs

61

BrowserCloseAsync()

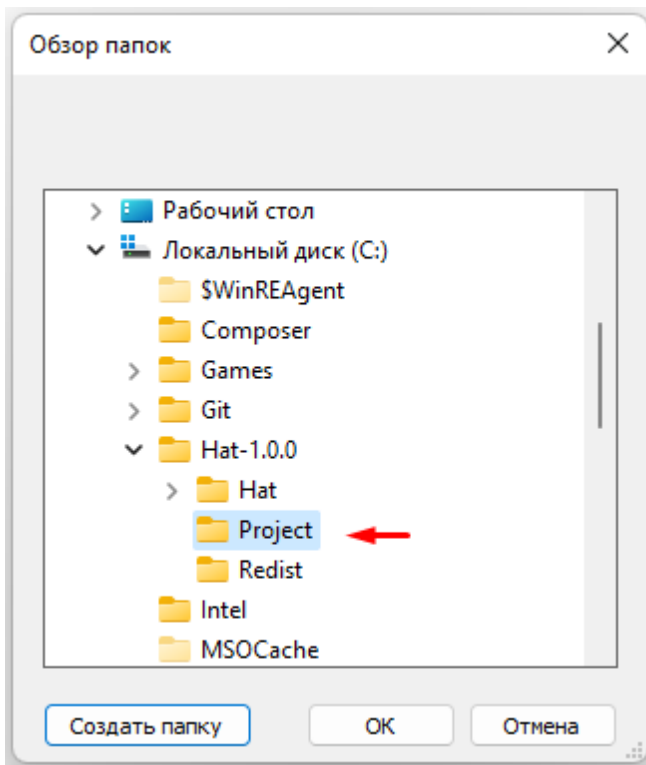




3.

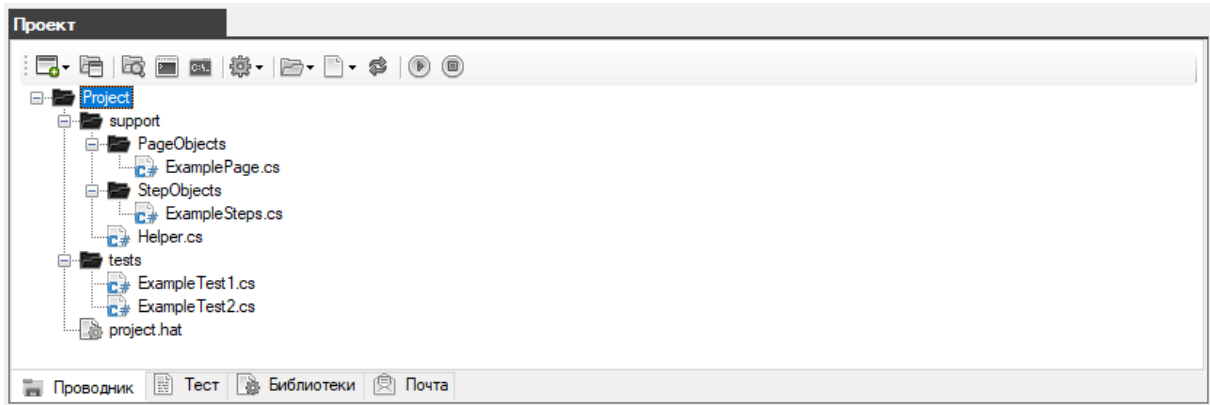
Project  
OK

C:\Hat-1.0.0\



4.

" " " "



5. Studio

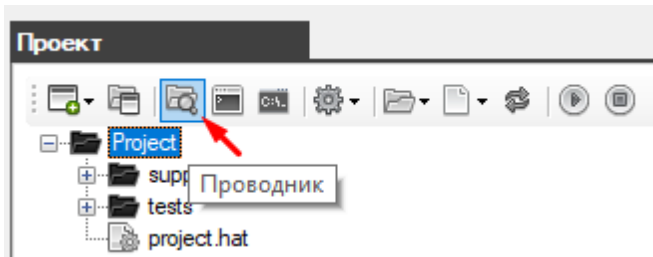
Visual Studio (

Visual

)  
Visual Studio

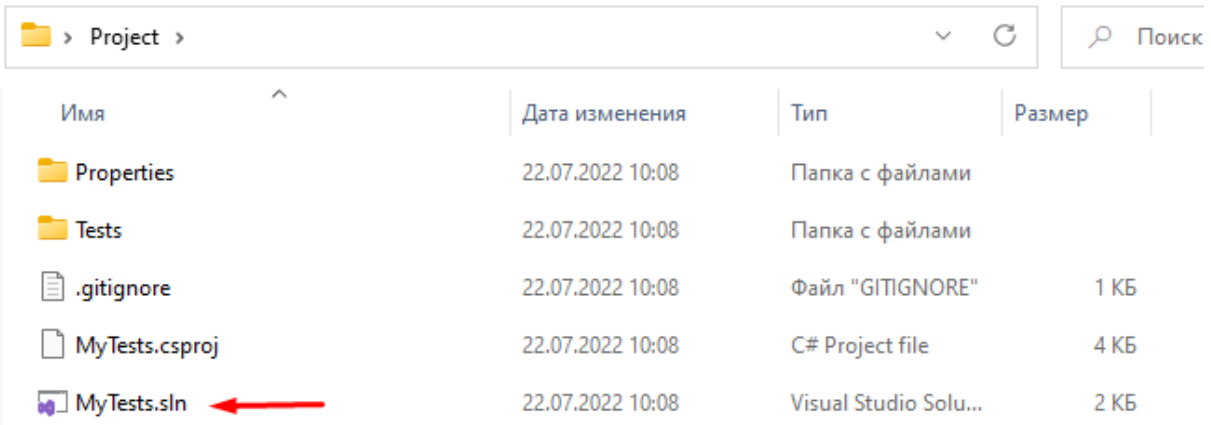
" "

" "



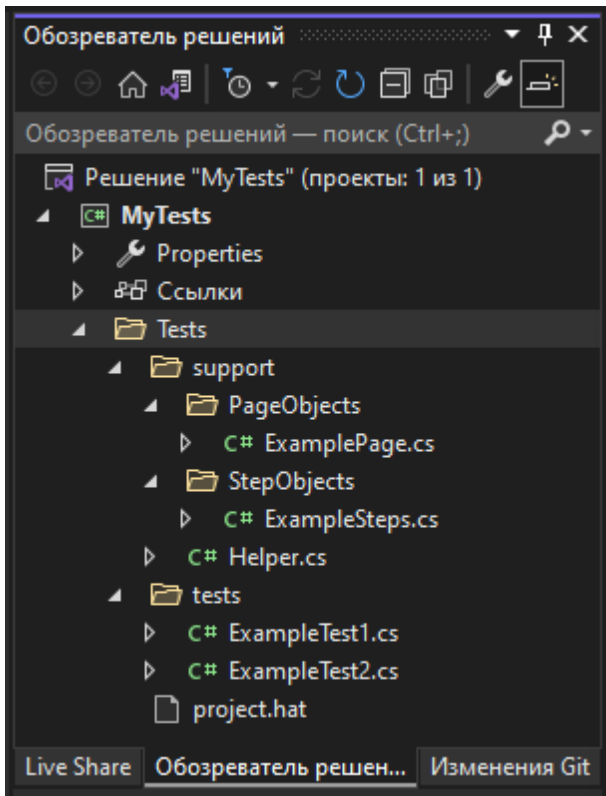
\*.sln (

MyTests.sln)



Visual Studio

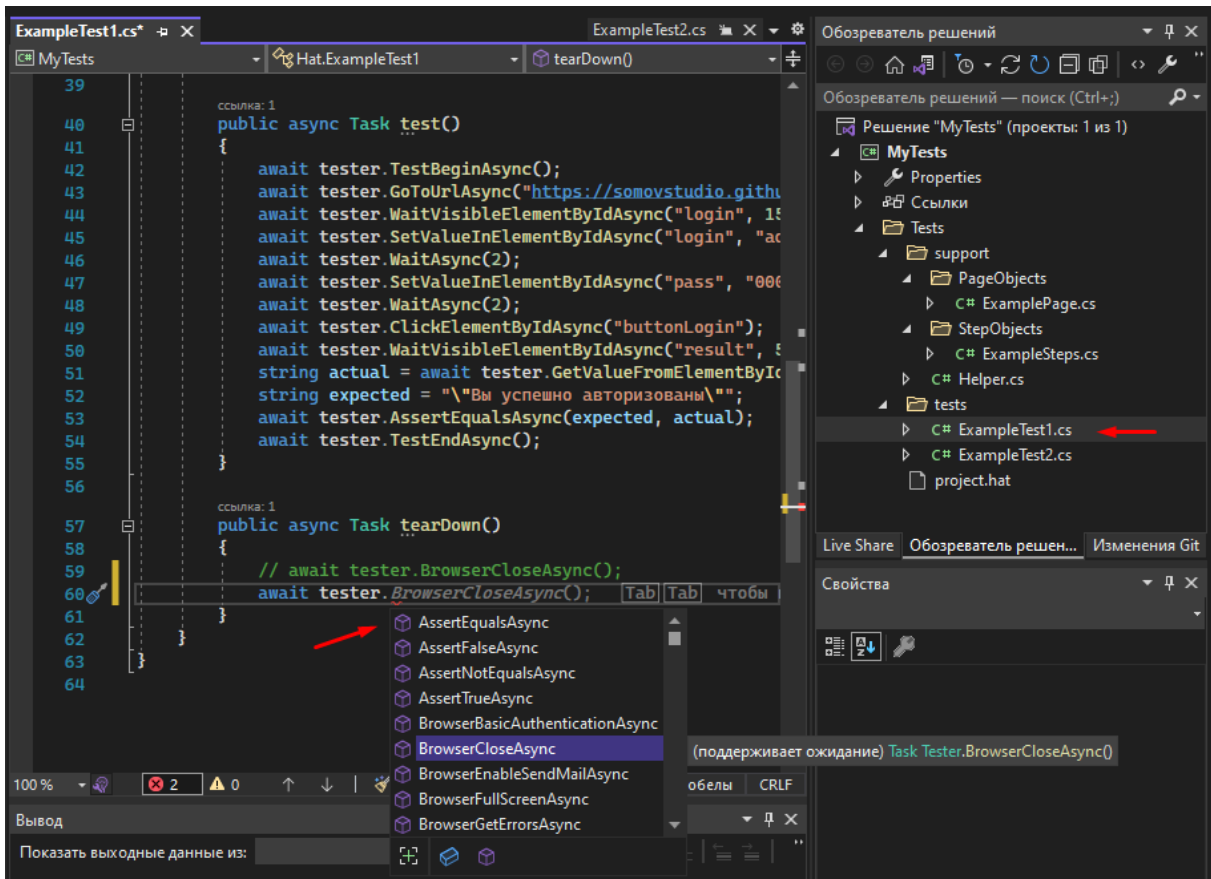
" "



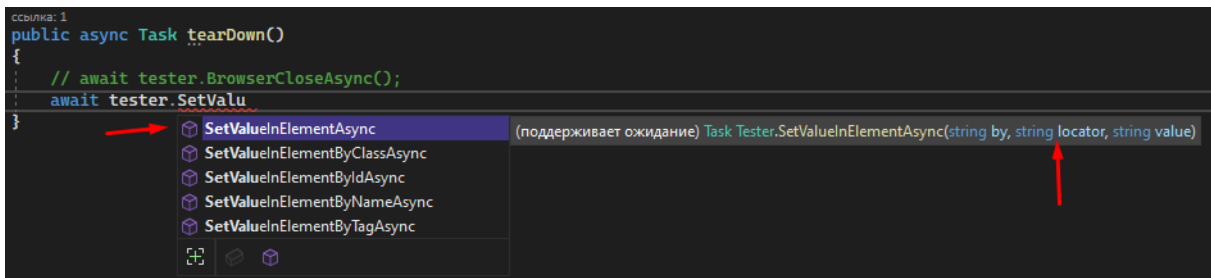
59

ExampleTest1.cs

BrowserCloseAsync()



Visual Studio



Created with the Personal Edition of HelpNDoc: [Write EPub books for the iPad](#)

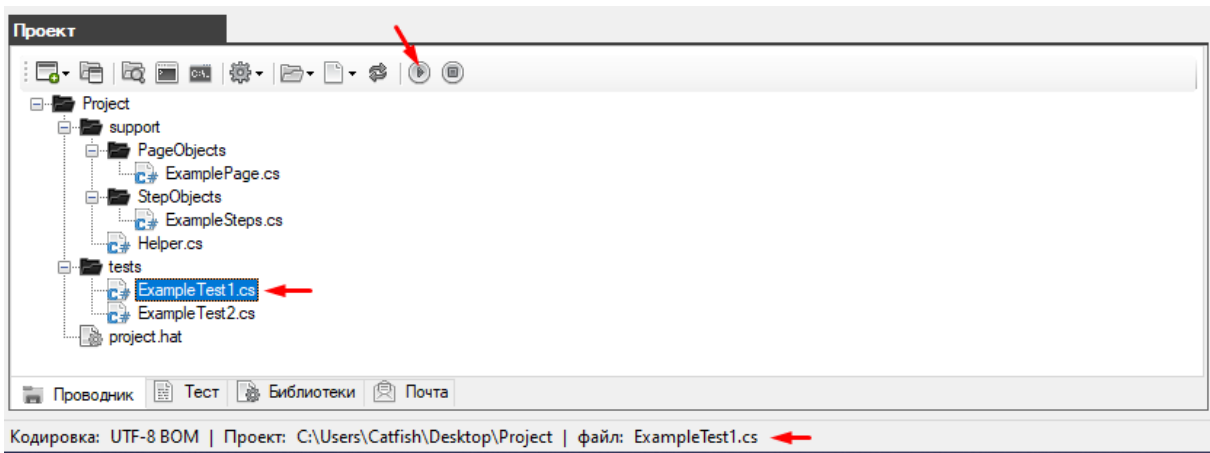
## Запуск автотеста и его результат

"

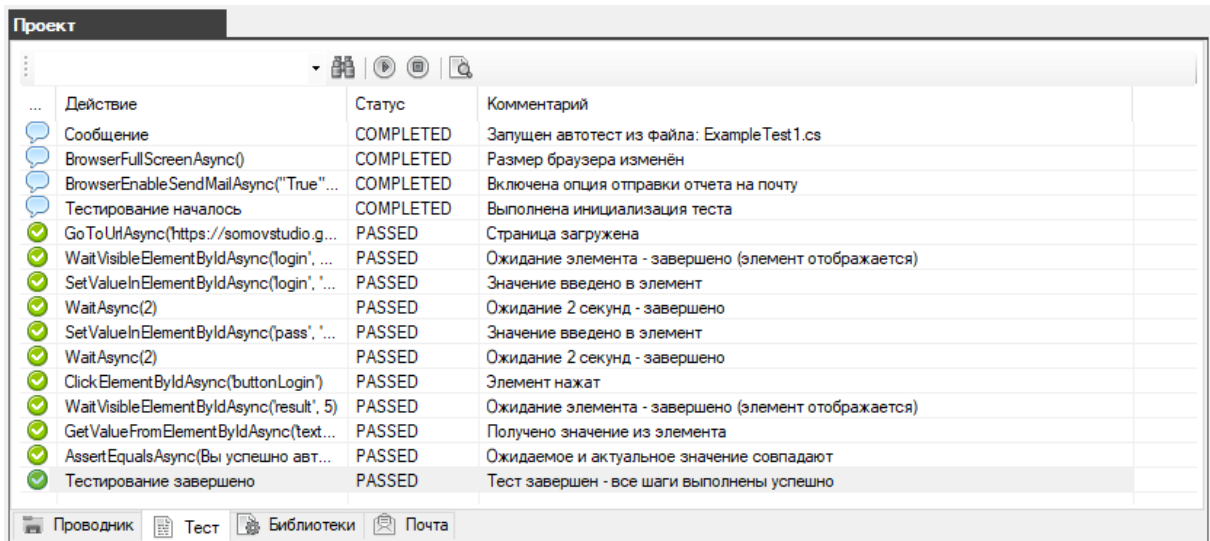
"

tests

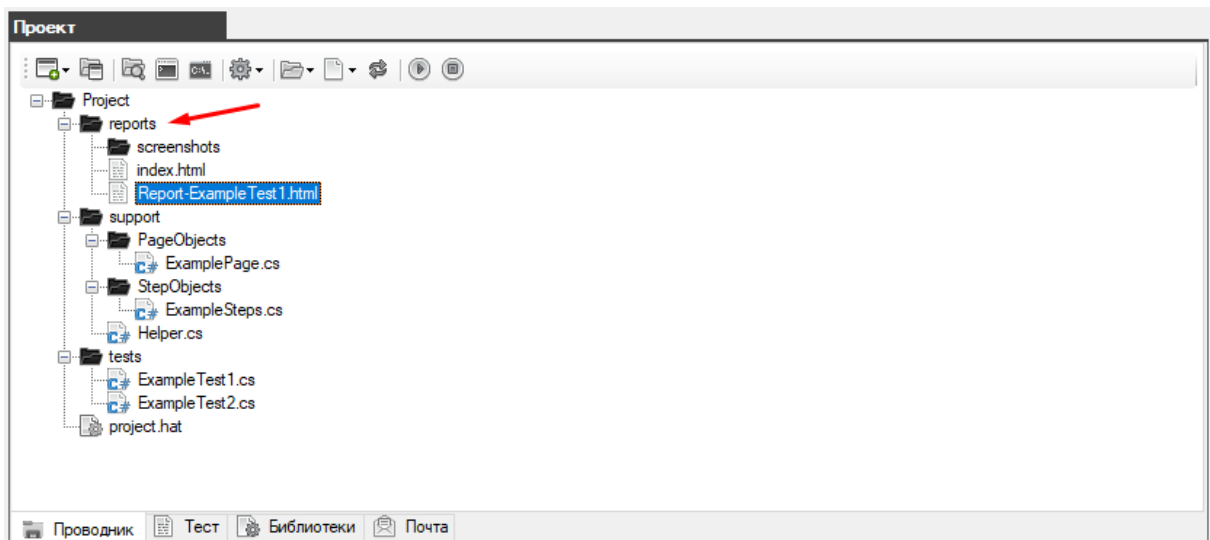
ExampleTest1.cs



" "



"report"



( Report-ExampleTest1.html)



Полный список результатов всех тестов

График результатов всех тестов в процентах:

Успех: 100%  
Неудача: 0%  
В работе: 0%

Всего тестов: 1  
Успешных тестов: 1  
Неудачных тестов: 0  
Тесты в работе: 0  
Дата отчета: 28.02.2023 13:13:26

Статус теста	Описание теста	Дата завершения	Файл	Отчет
Успех	Тест #1 проверяет авторизацию на сайте	28.02.2023 13:13:20	ExampleTest1.cs	<a href="#">Report-ExampleTest1.html</a>

Отчет о работе автотестера

Описание: Тест #1 проверяет авторизацию на сайте  
Файл: ExampleTest1.cs  
Дата: 28.02.2023 13:13:20

Результат: **Успешно**

Статус	Действие	Комментарий
Выполнено	Сообщение	Запущен автотест из файла: ExampleTest1.cs
Выполнено	Browser(FullScreenAsync)	Размер браузера изменён
Выполнено	Browser(EnableSendMailAsync("True", "True"))	Включена опция отправки отчета на почту
Выполнено	Тестирование началось	Выполнена инициализация теста
Успешно	GoToUrlAsync("https://somovstudio.github.io/test.html", 5)	Страница загружена
Успешно	WaitVisibleElementByIdAsync("login", 15)	Ожидание элемента - завершено (элемент отображается)
Успешно	SetValueInElementByIdAsync("login", "admin")	Значение введено в элемент
Успешно	WaitAsync(2)	Ожидание 2 секунд - завершено
Успешно	SetValueInElementByIdAsync("pass", "0000")	Значение введено в элемент
Успешно	WaitAsync(2)	Ожидание 2 секунд - завершено
Успешно	ClickElementByIdAsync("buttonLogin")	Элемент нажат
Успешно	WaitVisibleElementByIdAsync("result", 5)	Ожидание элемента - завершено (элемент отображается)
Успешно	GetValueFromElementByIdAsync("textarea")	Получено значение из элемента
Успешно	AssertEqualsAsync(Вы успешно авторизованы, Вы успешно авторизованы)	Ожидаемое и фактическое значение совпадают
Успешно	Тестирование завершено	Тест завершен - все шаги выполнены успешно

Success автотест ExampleTest1.cs - Тест #1 проверяет авторизацию на сайте

Browser Hat sep@zionic.ru Сегодня в 13:13  
Я >

## Отчет о работе автотеста

Описание: Тест #1 проверяет авторизацию на сайте

Файл: ExampleTest1.cs

Результат: **Успешно**

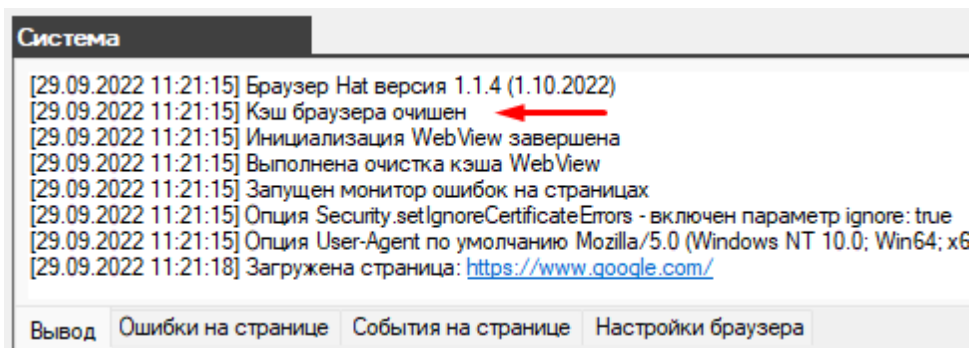
Статус	Действие	Комментарий
Выполнено	Сообщение	Запущен автотест из файла: ExampleTest1.cs
Выполнено	BrowserFullScreenAsync()	Размер браузера изменён
Выполнено	BrowserEnableSendMailAsync("True", "True")	Включена опция отправки отчета на почту
Выполнено	Тестирование началось	Выполнена инициализация теста
Успешно	GoToUrlAsync('https://somovstudio.github.io/test.html', 5)	Страница загружена
Успешно	WaitVisibleElementByIdAsync('login', 15)	Ожидание элемента - завершено (элемент отображается)
Успешно	SetValueInElementByIdAsync('login', 'admin')	Значение введено в элемент
Успешно	WaitAsync(2)	Ожидание 2 секунд - завершено
Успешно	SetValueInElementByIdAsync('pass', '0000')	Значение введено в элемент
Успешно	WaitAsync(2)	Ожидание 2 секунд - завершено
Успешно	ClickElementByIdAsync('buttonLogin')	Элемент нажат
Успешно	WaitVisibleElementByIdAsync('result', 5)	Ожидание элемента - завершено (элемент отображается)
Успешно	GetValueFromElementByIdAsync('textarea')	Получено значение из элемента
Успешно	AssertEqualsAsync(Вы успешно авторизованы, Вы успешно авторизованы)	Ожидаемое и фактическое значение совпадают
Успешно	Тестирование завершено	Тест завершен - все шаги выполнены успешно

Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

## Кэш браузера

Created with the Personal Edition of HelpNDoc: [Create cross-platform Qt Help files](#)

## Очистка



"Hat.exe.WebView2"

> Локальный диск (C:) > Hat

Поиск в: Hat

Имя	Дата изменения	Тип	Размер
Hat.exe.WebView2	29.09.2022 9:47	Папка с файлами	
runtimes	12.08.2022 13:00	Папка с файлами	
Hat.exe	06.09.2022 9:27	Приложение	1 482 КБ
HatFramework.dll	06.09.2022 9:17	Расширение при...	203 КБ
help.chm	06.09.2022 9:29	Скомпилирован...	1 990 КБ
ICSharpCode.TextEditor.dll	12.08.2022 10:37	Расширение при...	344 КБ
Microsoft.Web.WebView2.Core.dll	12.08.2022 10:39	Расширение при...	382 КБ
Microsoft.Web.WebView2.WinForms.dll	12.08.2022 10:39	Расширение при...	36 КБ
Microsoft.Web.WebView2.Wpf.dll	12.08.2022 10:39	Расширение при...	43 КБ
Newtonsoft.Json.dll	12.08.2022 10:37	Расширение при...	686 КБ

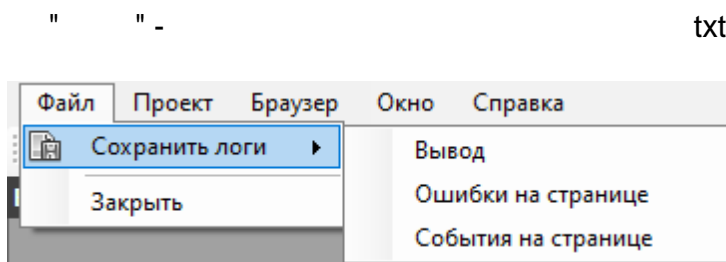
"Hat.exe.WebView2",

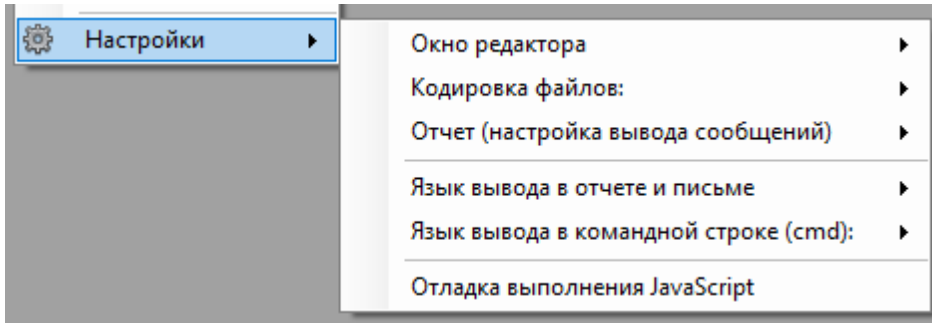
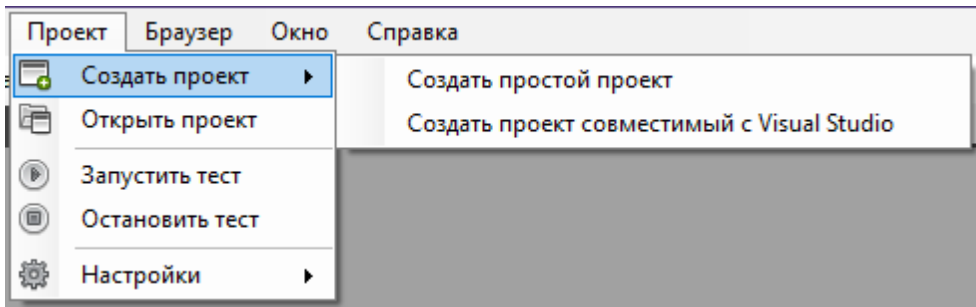
Created with the Personal Edition of HelpNDoc: [Full-featured Help generator](#)

## Интерфейс

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

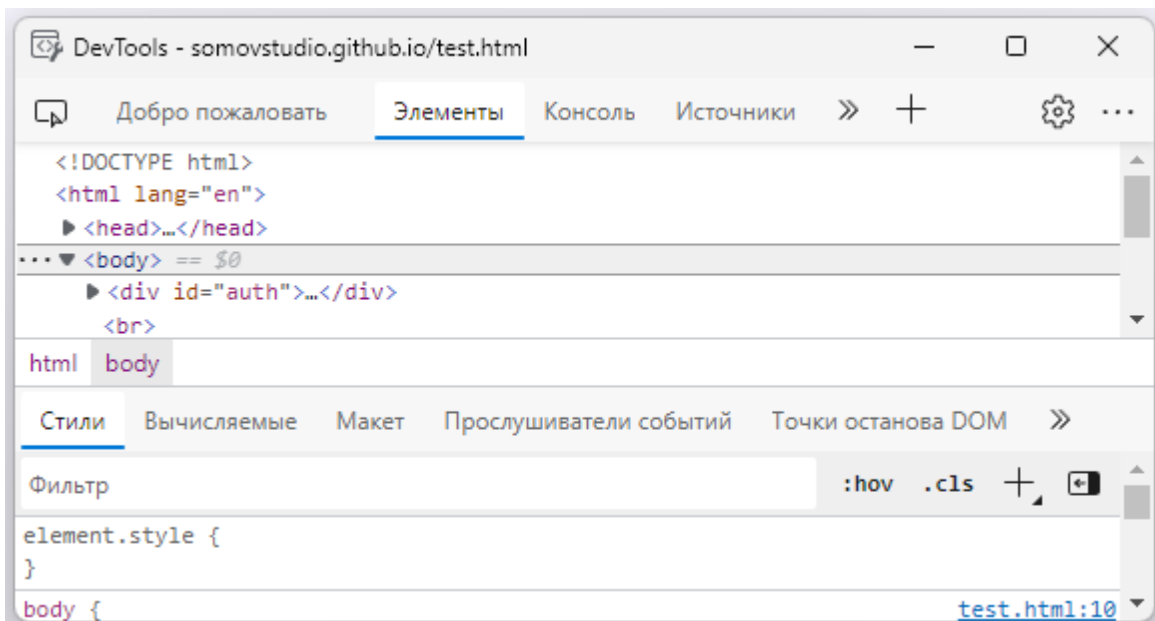
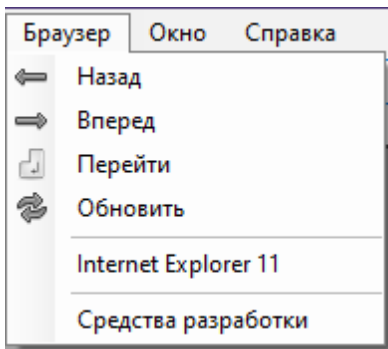
## Основное меню



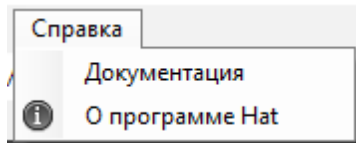
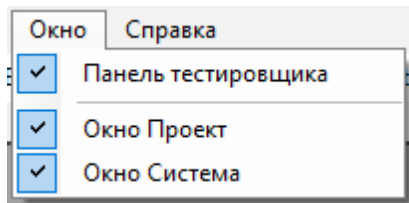


" " \_

Internet Explorer 11



" " \_



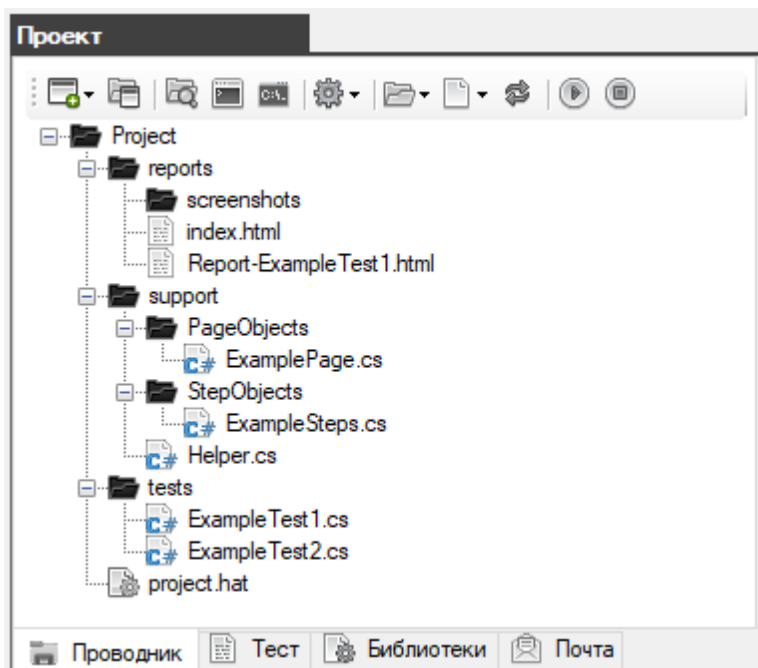
Created with the Personal Edition of HelpNDoc: [Easily create Help documents](#)

## Панель инструментов



Created with the Personal Edition of HelpNDoc: [iPhone web sites made easy](#)

## Окно Проект - вкладка Проводник

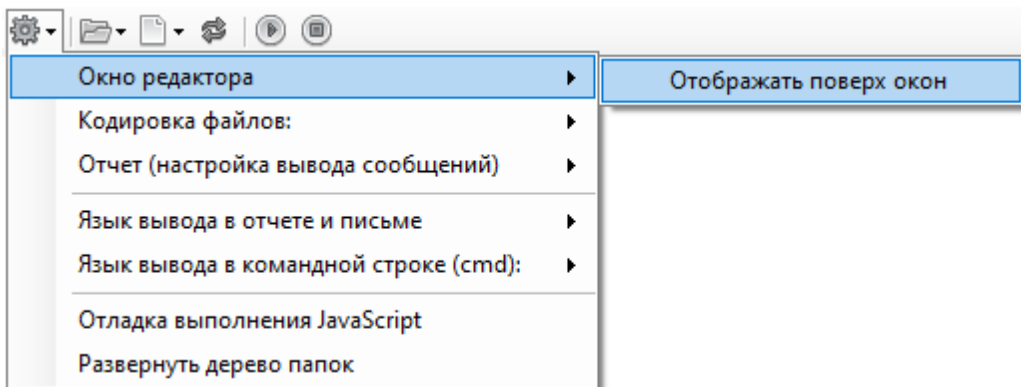


:

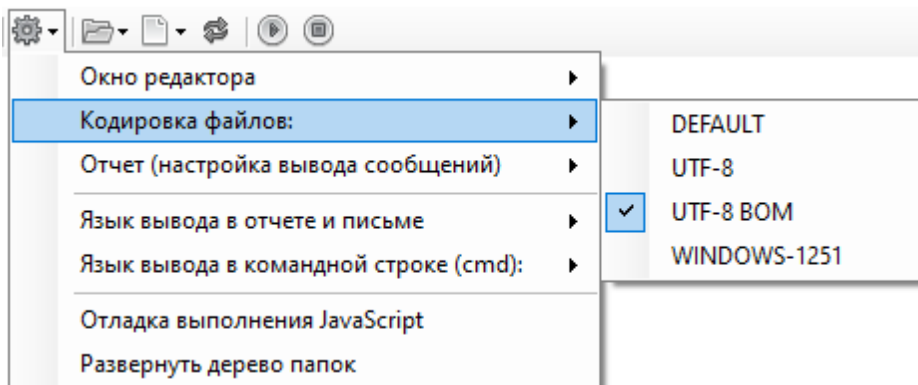


- - Visual Studio) (
- - Windows
- - Windows (cmd)
- - Windows
- (cmd) -
- -
- -
- -
- -
- -
- -
- -

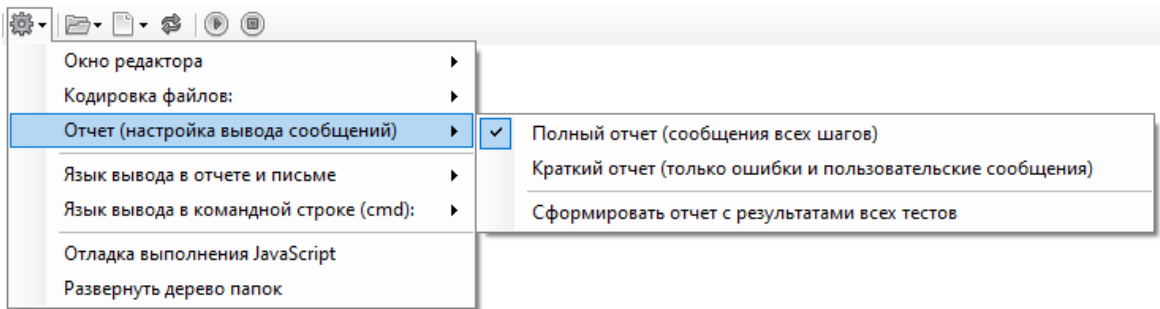
○



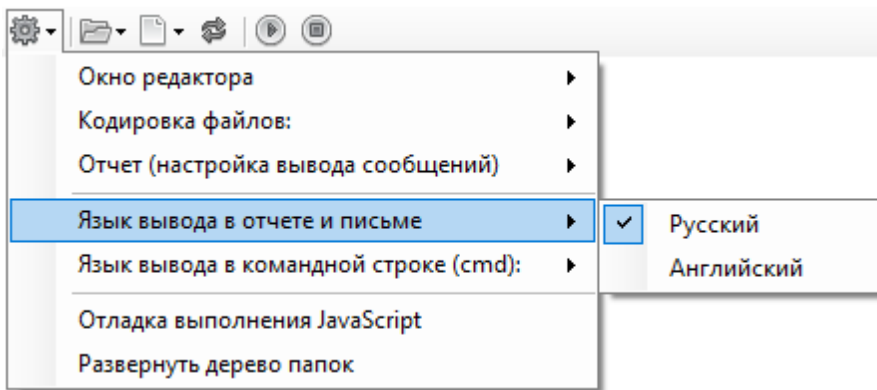
○



○

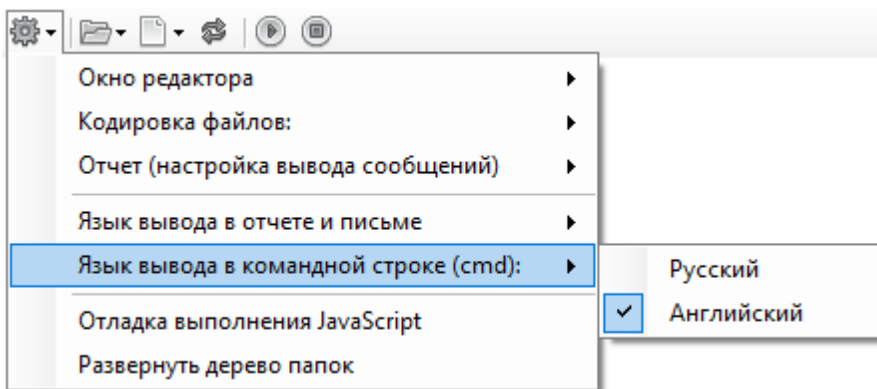


○



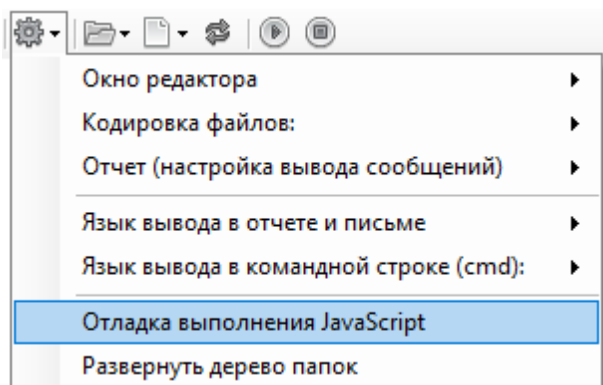
○

(cmd)

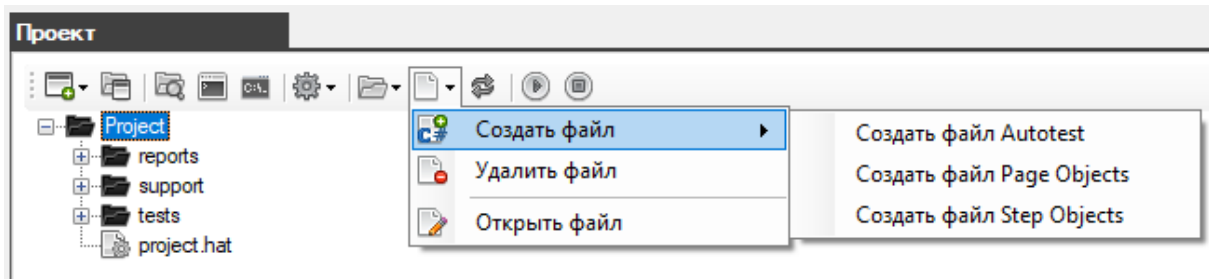
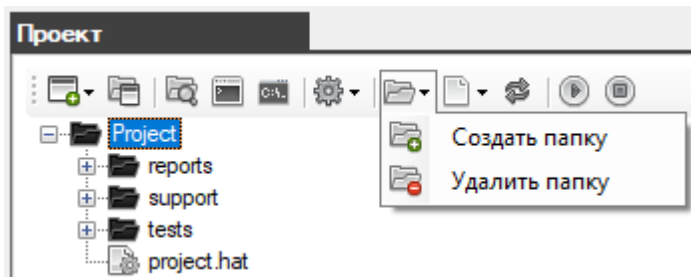
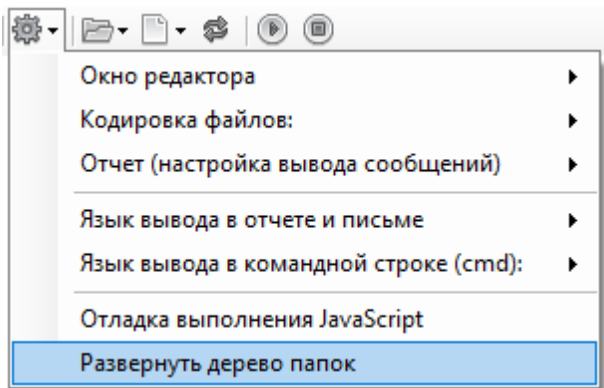


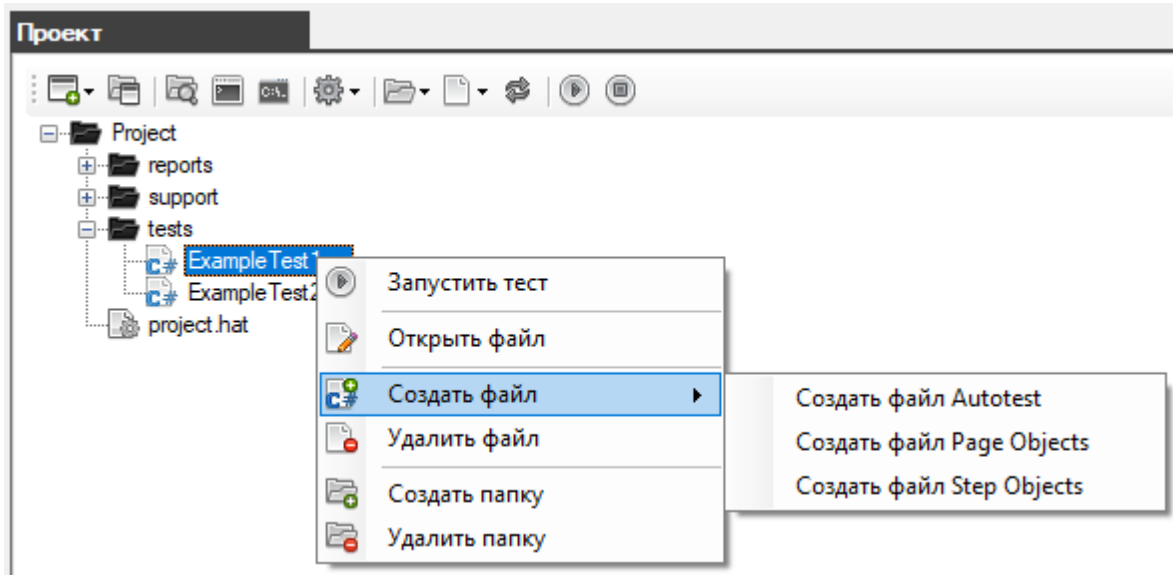
○

JavaScript



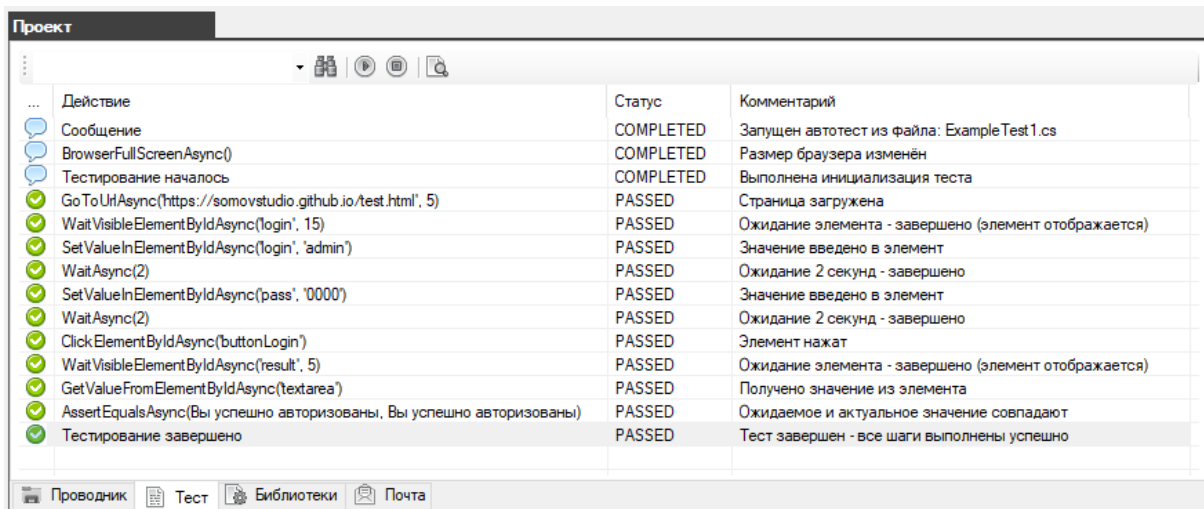
○





Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

## Окно Проект - вкладка Тест

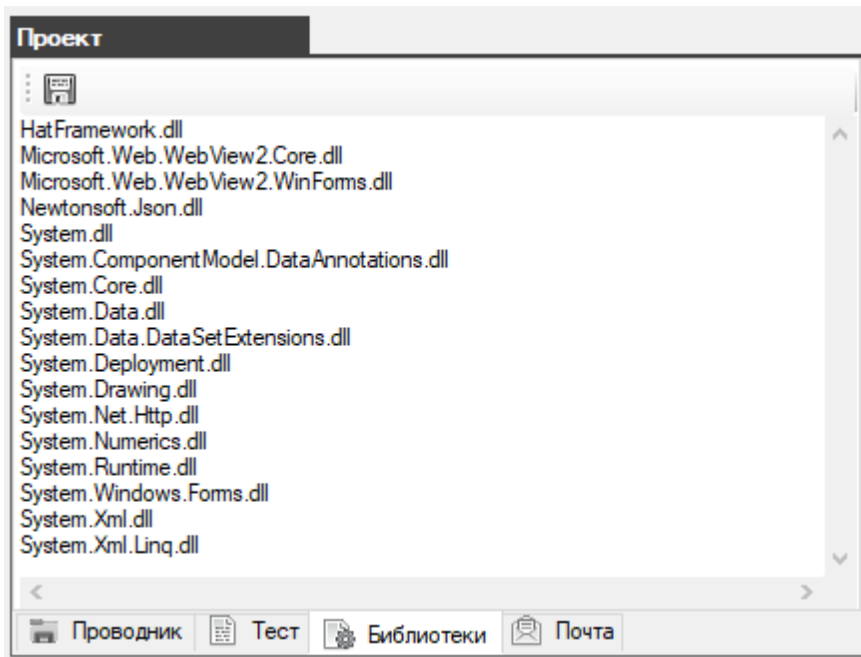


Действие	Статус	Комментарий
Сообщение	COMPLETED	Запущен автотест из файла: ExampleTest1.cs
BrowserFullScreenAsync()	COMPLETED	Размер браузера изменён
Тестирование началось	COMPLETED	Выполнена инициализация теста
Go To Uri Async (https://somovstudio.github.io/test.html, 5)	PASSED	Страница загружена
Wait Visible Element By Id Async (login, 15)	PASSED	Ожидание элемента - завершено (элемент отображается)
Set Value In Element By Id Async (login, 'admin')	PASSED	Значение введено в элемент
Wait Async (2)	PASSED	Ожидание 2 секунд - завершено
Set Value In Element By Id Async (pass, '0001')	PASSED	Значение введено в элемент
Wait Async (2)	PASSED	Ожидание 2 секунд - завершено
Click Element By Id Async (buttonLogin)	PASSED	Элемент нажат
Wait Visible Element By Id Async (result, 5)	PASSED	Ожидание элемента - завершено (элемент отображается)
Get Value From Element By Id Async (textarea)	PASSED	Получено значение из элемента
Assert Equals Async (Вы успешно авторизованы, Неверный логин или пароль)	FAILED	Ожидаемое и фактическое значение не совпадают
Тестирование завершено	FAILED	Тест завершен - шаги теста выполнены неуспешно



- -
- -
- -
- -
- -

### Окно Проект - вкладка Библиотеки



:



o

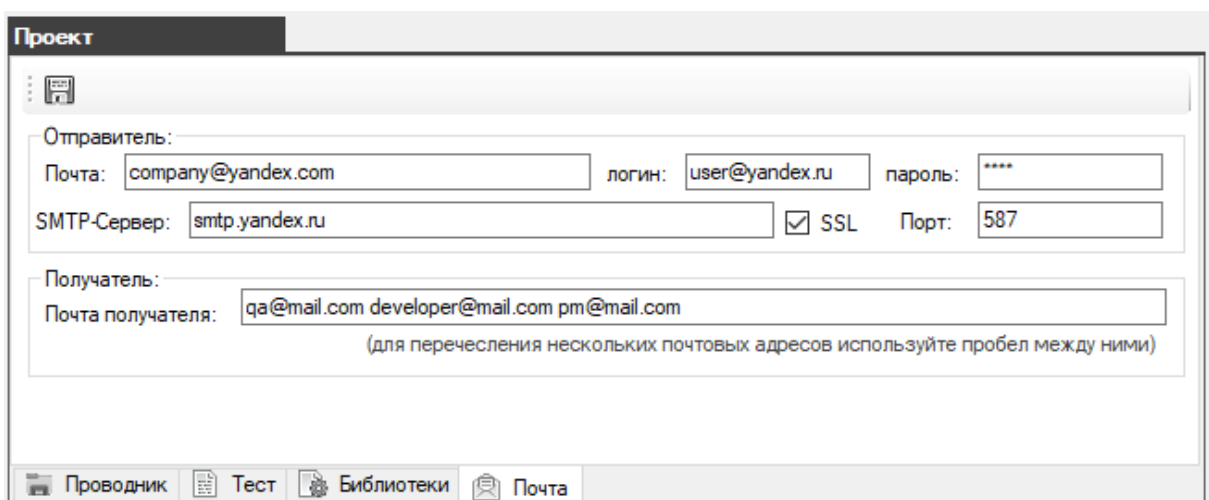
-

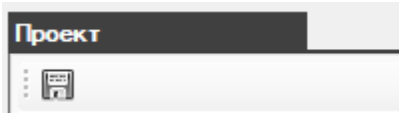
project.hat

Created with the Personal Edition of HelpNDoc: [iPhone web sites made easy](#)

## Окно Проект - вкладка Почта

-

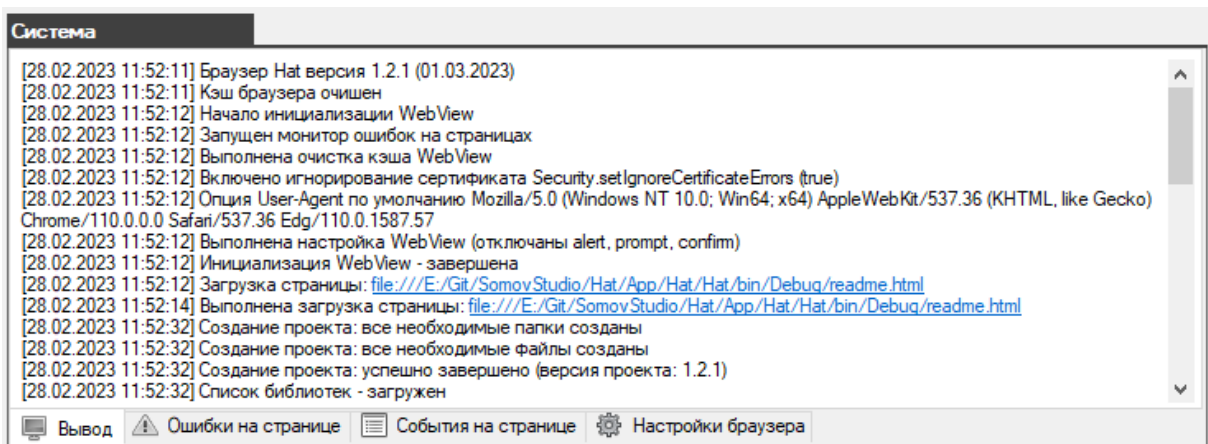




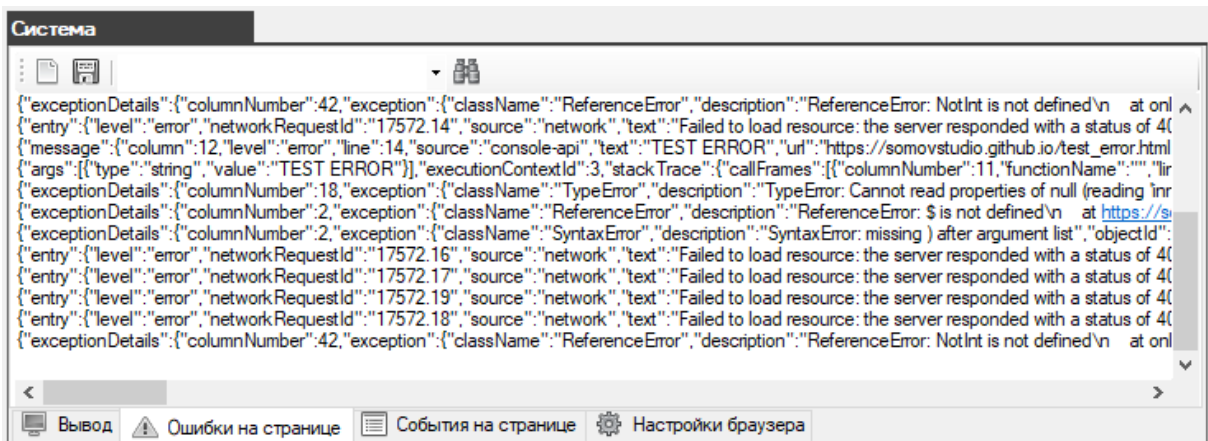
project.hat

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

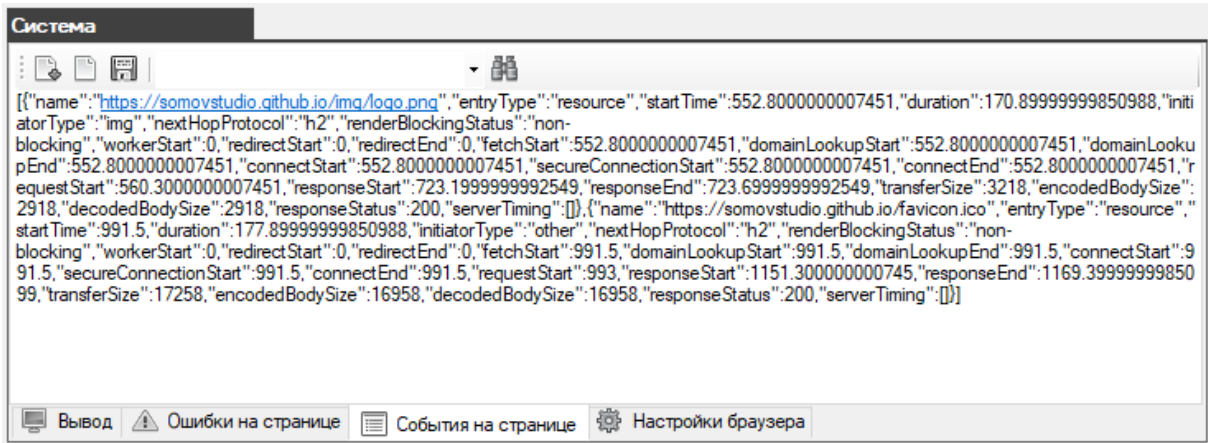
## Окно система



(console)

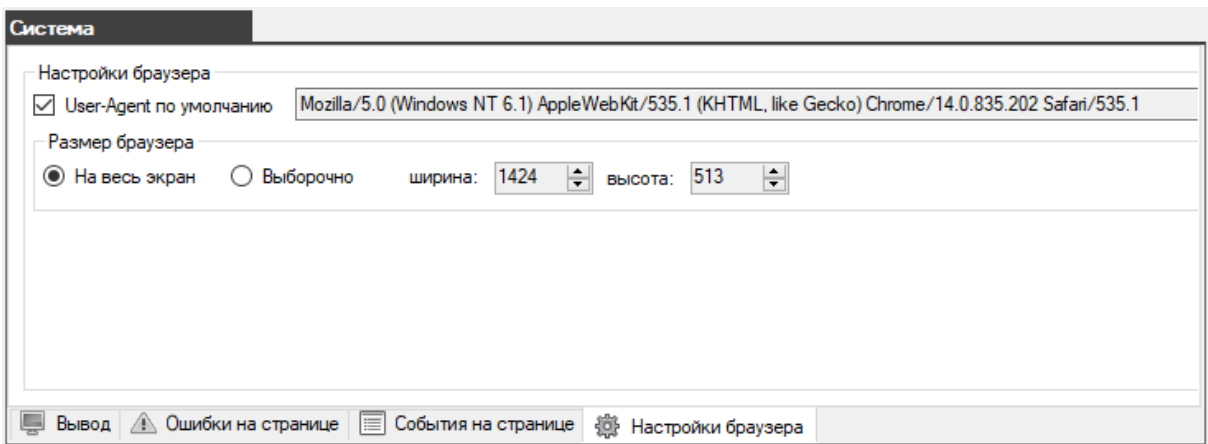


(network)



"

"



Created with the Personal Edition of HelpNDoc: [Free iPhone documentation generator](#)

## Отчеты

Created with the Personal Edition of HelpNDoc: [Easily create EBooks](#)

## Отчет работы автотеста

reports

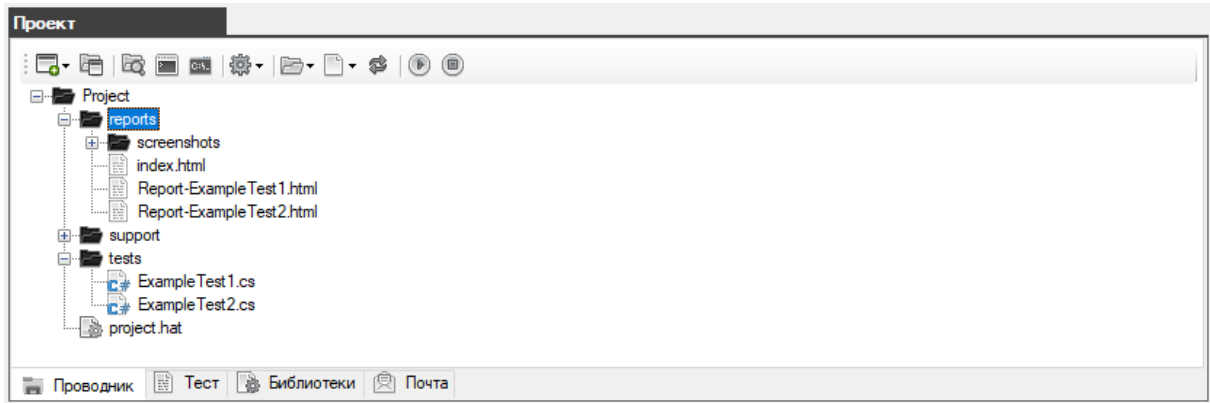
html.

Report

ExampleTest1.cs


Report-ExampleTest1.html

index.html



(index.html)

:



**Browser Hat**

**Полный список результатов всех тестов**

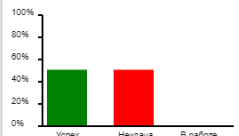


График результатов всех тестов в процентах:

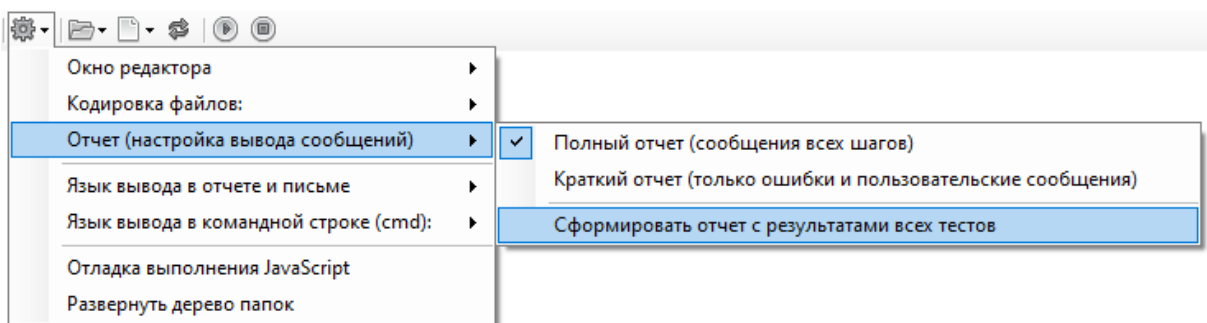
Успех: 50%  
Неудача: 50%  
В работе: 0%

Всего тестов: 2  
Успешных тестов: 1  
Неудачных тестов: 1  
Тесты в работе: 0  
Дата отчета: 28.02.2023 12:18:35

Статус теста	Описание теста	Дата завершения	Файл	Отчет
Неудача	Тест #1 проверяет авторизацию на сайте	28.02.2023 12:18:19	ExampleTest1.cs	<a href="#">Report-ExampleTest1.html</a>
Успех	Тест #2 проверяет авторизацию на сайте	28.02.2023 12:18:29	ExampleTest2.cs	<a href="#">Report-ExampleTest2.html</a>

Browser Hat 1.2.1

(index.html)



:

### Отчет о работе автотеста

Описание: Тест #2 проверяет авторизацию на сайте  
 Файл: ExampleTest2.cs  
 Дата: 28.02.2023 12:18:29

Результат: **Успешно**

Статус	Действие	Комментарий
Выполнено	Сообщение	Запущен автотест из файла: ExampleTest2.cs
Выполнено	BrowserFullScreenAsync()	Размер браузера изменён
Выполнено	Тестирование началось	Выполнена инициализация теста
Успешно	GoToUrlAsync('https://somovstudio.github.io/test.html', 5)	Страница загружена
Успешно	WaitVisibleElementByIdAsync('login', 15)	Ожидание элемента - завершено (элемент отображается)
Успешно	SetValueInElementByIdAsync('login', 'admin')	Значение введено в элемент
Успешно	WaitAsync(2)	Ожидание 2 секунд - завершено
Успешно	SetValueInElementByIdAsync('pass', '0000')	Значение введено в элемент
Успешно	WaitAsync(2)	Ожидание 2 секунд - завершено
Успешно	ClickElementByIdAsync('buttonLogin')	Элемент нажат
Успешно	WaitVisibleElementByIdAsync('result', 5)	Ожидание элемента - завершено (элемент отображается)
Успешно	GetValueFromElementByIdAsync('textarea')	Получено значение из элемента
Успешно	AssertEqualsAsync(Вы успешно авторизованы, Вы успешно авторизованы)	Ожидаемое и фактическое значение совпадают
Успешно	Тестирование завершено	Тест завершен - все шаги выполнены успешно

:

### Отчет о работе автотеста

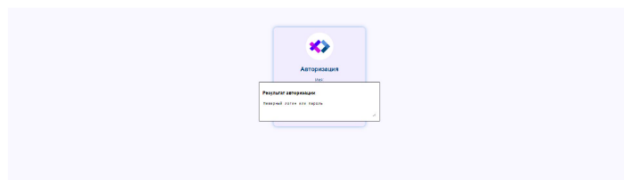
Описание: Тест #1 проверяет авторизацию на сайте  
 Файл: ExampleTest1.cs  
 Дата: 28.02.2023 12:18:19

Результат: **Провально**

Статус	Действие	Комментарий
Выполнено	Сообщение	Запущен автотест из файла: ExampleTest1.cs
Выполнено	BrowserFullScreenAsync()	Размер браузера изменён
Выполнено	Тестирование началось	Выполнена инициализация теста
Успешно	GoToUrlAsync('https://somovstudio.github.io/test.html', 5)	Страница загружена
Успешно	WaitVisibleElementByIdAsync('login', 15)	Ожидание элемента - завершено (элемент отображается)
Успешно	SetValueInElementByIdAsync('login', 'admin')	Значение введено в элемент
Успешно	WaitAsync(2)	Ожидание 2 секунд - завершено
Успешно	SetValueInElementByIdAsync('pass', '0001')	Значение введено в элемент
Успешно	WaitAsync(2)	Ожидание 2 секунд - завершено
Успешно	ClickElementByIdAsync('buttonLogin')	Элемент нажат
Успешно	WaitVisibleElementByIdAsync('result', 5)	Ожидание элемента - завершено (элемент отображается)
Успешно	GetValueFromElementByIdAsync('textarea')	Получено значение из элемента
Провально	AssertEqualsAsync(Вы успешно авторизованы, Неверный логин или пароль)	Ожидаемое и фактическое значение не совпадают
Провально	Тестирование завершено	Тест завершен - шаги теста выполнены неуспешно

Скриншот

Файл: [image-28-18-2023-12-18-25.jpeg](#)



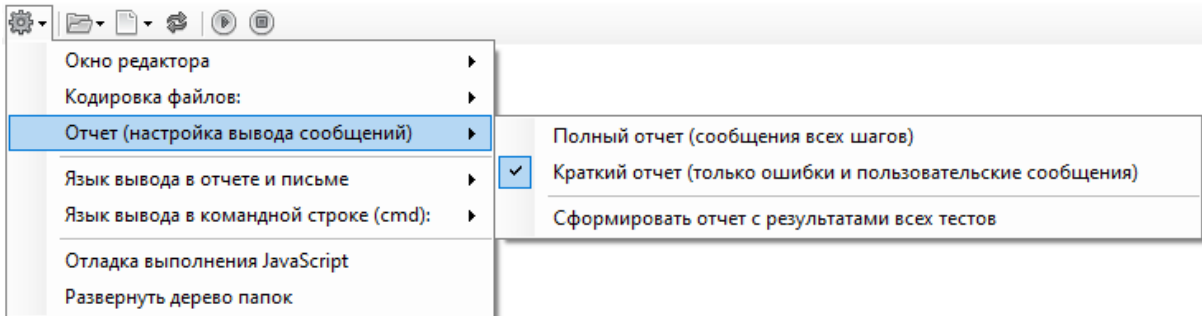
:

Отчет о работе автотеста

Файл: ExampleTest1.cs

Результат: **Провально**

Статус	Действие	Комментарий
ОШИБКА		c:\Users\Catfish\Desktop\Hat\projecttests\ExampleTest1.cs(50,19) : error CS1502: Наиболее подходящий перегруженный метод для "HatFramework.Tester.ClickElementByIdAsync(string)" имеет несколько недопустимых аргументов
ОШИБКА		c:\Users\Catfish\Desktop\Hat\projecttests\ExampleTest1.cs(50,48) : error CS1503: Аргумент "1": преобразование типа из "int" в "string" невозможно
ОШИБКА		c:\Users\Catfish\Desktop\Hat\projecttests\ExampleTest1.cs(58,27) : warning CS1998: У этого асинхронного метода нет операторов "await", поэтому он будет выполняться синхронно. Рассмотрите возможность использования оператора "await" для ожидания неблокирующих вызовов API или конструкции "await Task.Run(...)" для выполнения фонового потока, ограниченного производительностью процессора.
ОШИБКА		c:\Users\Catfish\Desktop\Hat\projecttests\ExampleTest2.cs(53,27) : warning CS1998: У этого асинхронного метода нет операторов "await", поэтому он будет выполняться синхронно. Рассмотрите возможность использования оператора "await" для ожидания неблокирующих вызовов API или конструкции "await Task.Run(...)" для выполнения фонового потока, ограниченного производительностью процессора.
ОШИБКА		c:\Users\Catfish\Desktop\Hat\projecttests\ExampleTest3.cs(64,27) : warning CS1998: У этого асинхронного метода нет операторов "await", поэтому он будет выполняться синхронно. Рассмотрите возможность использования оператора "await" для ожидания неблокирующих вызовов API или конструкции "await Task.Run(...)" для выполнения фонового потока, ограниченного производительностью процессора.
Ошибок:	5	



Отчет о работе автотеста

Описание: Тест #2 проверяет авторизацию на сайте

Файл: ExampleTest2.cs

Дата: 28.02.2023 12:48:32

Результат: **Успешно**

Статус	Действие	Комментарий
Выполнено	Тестирование началось	Выполнена инициализация теста
Успешно	Тестирование завершено	Тест завершен - все шаги выполнены успешно

Отчет о работе автотеста

Описание: Тест #1 проверяет авторизацию на сайте

Файл: ExampleTest1.cs

Дата: 28.02.2023 12:50:05

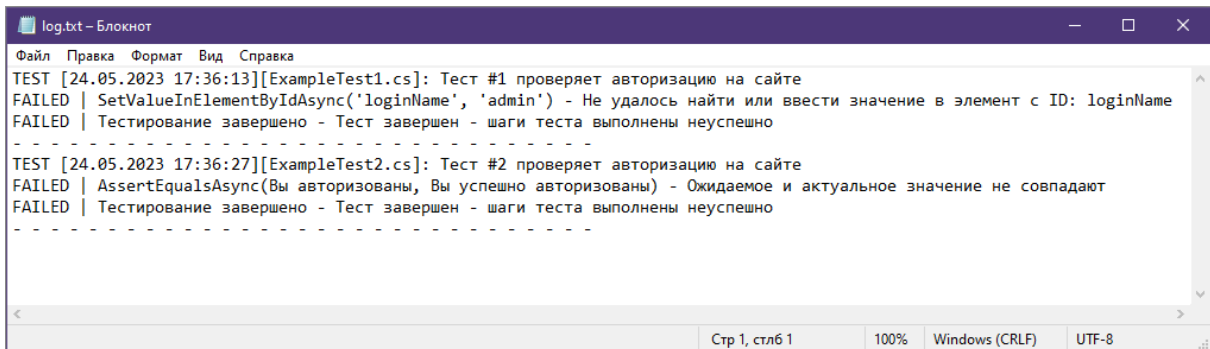
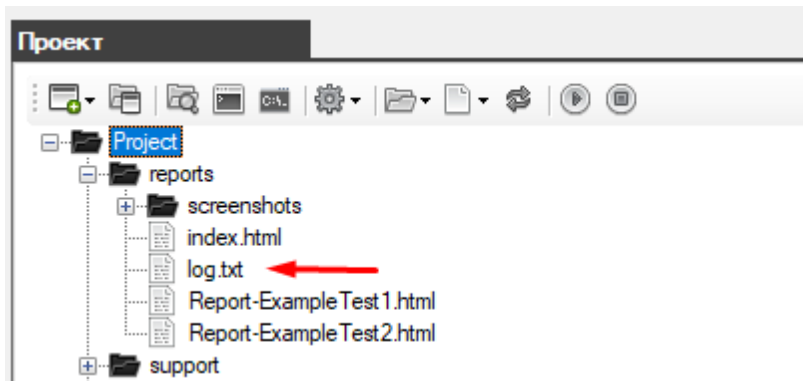
Результат: **Провально**

Статус	Действие	Комментарий
Выполнено	Тестирование началось	Выполнена инициализация теста
Провально	AssertEqualsAsync(Вы успешно авторизованы, Неверный логин или пароль)	Ожидаемое и фактическое значение не совпадают
Провально	Тестирование завершено	Тест завершен - шаги теста выполнены неуспешно
Скриншот	Файл: <a href="#">image-28-50-2023-12-50-11.jpeg</a>	

## Лог файл

(log.txt)

log.txt



## Почта

## Отправить отчет о провале автотеста на почту

" "

Yandex.

**Проект**

Отправитель:

Почта:  логин:  пароль:

SMTP-Сервер:   SSL Порт:

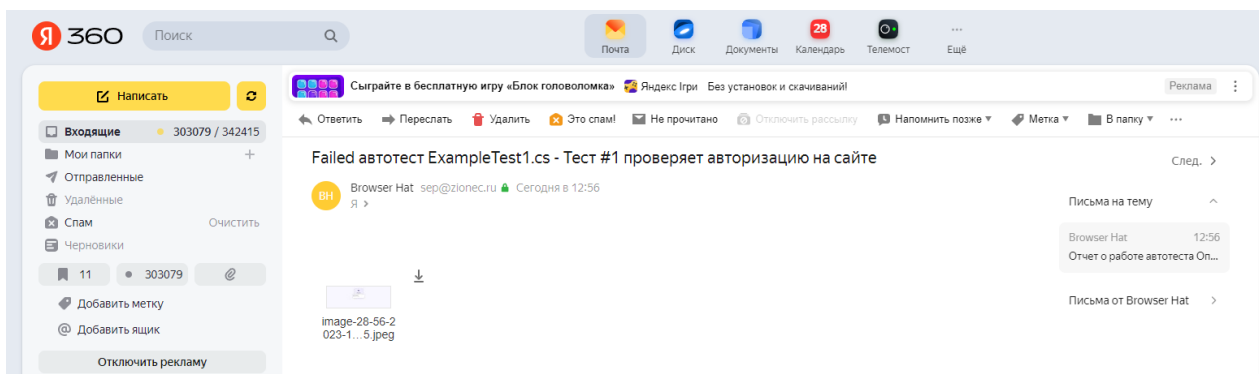
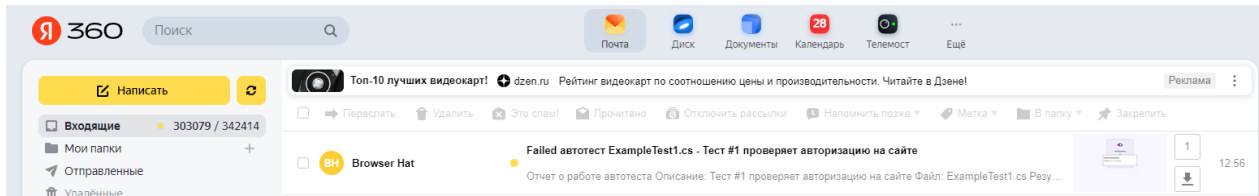
Получатель:

Почта получателя:   
(для перечисления нескольких почтовых адресов используйте пробел между ними)

Проводник Тест Библиотеки Почта

## BrowserEnableSendMailAsync

```
public async Task setUp()
{
    await tester.BrowserEnableSendMailAsync(true, false); //
    await tester.BrowserEnableSendMailAsync(false, true); //
    await tester.BrowserEnableSendMailAsync(); //
}
```

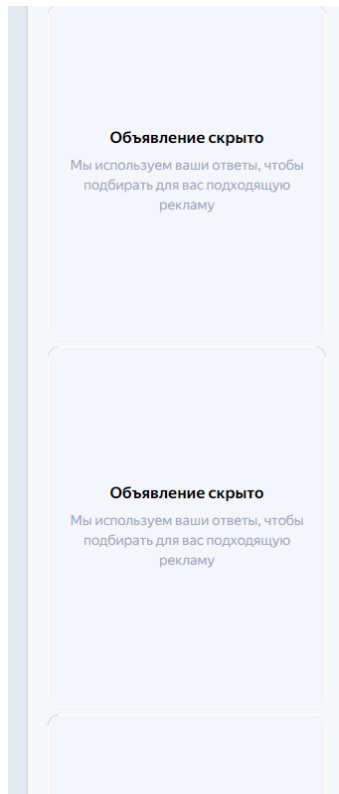


### Отчет о работе автотеста

Описание: Тест #1 проверяет авторизацию на сайте

Файл: ExampleTest1.cs

Результат: **Провально**



Статус	Действие	Комментарий
Выполнено	Сообщение	Запущен автотест из файла: ExampleTest1.cs
Выполнено	BrowserFullScreenAsync()	Размер браузера изменён
Выполнено	BrowserEnableSendMailAsync("True", "True")	Включена опция отправки отчета на почту
Выполнено	Тестирование началось	Выполнена инициализация теста
Успешно	GoToUrlAsync("https://somovstudio.github.io/test.html", 5)	Страница загружена
Успешно	WaitVisibleElementByIdAsync('login', 15)	Ожидание элемента - завершено (элемент отображается)
Успешно	SetValueInElementByIdAsync('login', 'admin')	Значение введено в элемент
Успешно	WaitAsync(2)	Ожидание 2 секунд - завершено
Успешно	SetValueInElementByIdAsync('pass', '0001')	Значение введено в элемент
Успешно	WaitAsync(2)	Ожидание 2 секунд - завершено
Успешно	ClickElementByIdAsync('buttonLogin')	Элемент нажат
Успешно	WaitVisibleElementByIdAsync('result', 5)	Ожидание элемента - завершено (элемент отображается)
Успешно	GetValueFromElementByIdAsync('textarea')	Получено значение из элемента
Провально	AssertEqualsAsync(Вы успешно авторизованы, Неверный логин или пароль)	Ожидаемое и фактическое значение не совпадают
Провально	Тестирование завершено	Тест завершен - шаги теста выполнены неуспешно
Скриншот	Файл: image-28-56-2023-12-56-35.jpeg	

### SendMsgToMailAsync

```
public async Task tearDown()
{
    if (tester.GetTestResult() == Tester.PASSED)
    {
        await tester.SendMsgToMailAsync(" ", " ");
    }
}
```

Created with the Personal Edition of HelpNDoc: [Free EPub producer](#)

### Консольный режим запуска (Jenkins)

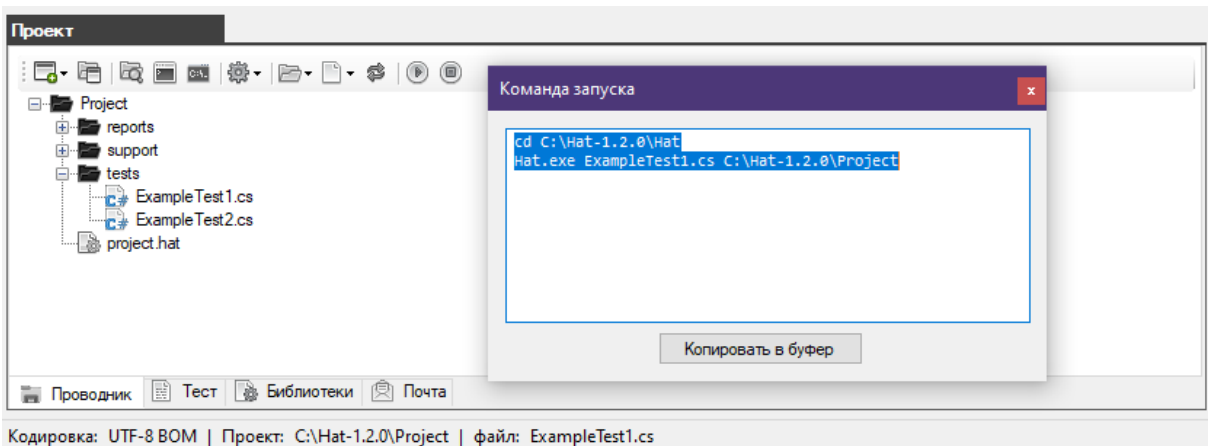
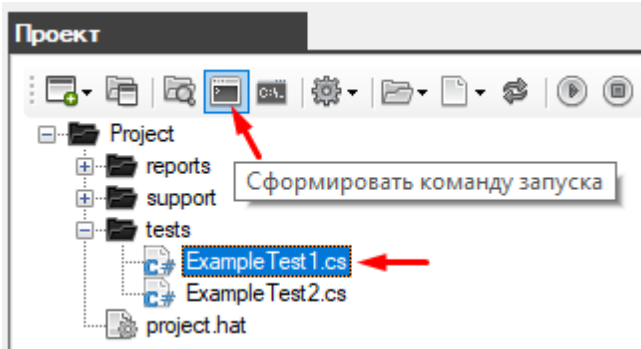
Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

### Запуск из командной строки

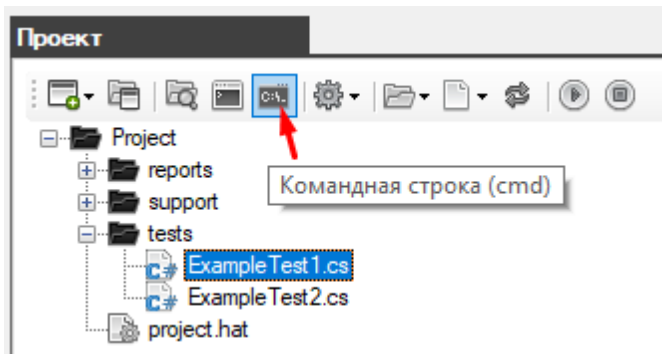
Windows.



" ( " )  
.  
"



- : cd C:\Hat-1.2.0\Hat - Hat.exe
  - : Hat.exe ExampleTest1.cs C:\Hat-1.2.0\Project -
- Hat.exe [ \_ \_ ] [ \_ \_ \_ \_ ]



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.2604]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Hat-1.2.0\Hat>cd C:\Hat-1.2.0\Hat
C:\Hat-1.2.0\Hat>Hat.exe ExampleTest1.cs C:\Hat-1.2.0\Project
```

```

C:\Windows\System32\cmd.exe
Browser Hat version 1.2.3 (1.04.2023)
The browser cache has been cleared
Launching the browser...
Проект открыт (версия проекта: 1.2.3)
Обновлен файл отчета Report-ExampleTest1.html

Создан отчет с результатами всех тестов

Запущен файл автотеста: ExampleTest1.cs
Шаг[в процессе]: Запуск автотеста
Шаг[выполнено]: Запущен автотест из файла: ExampleTest1.cs
Описание: Тест #1 проверяет авторизацию на сайте
Шаг[в процессе]: Изменяется размер браузера
Шаг[выполнено]: Размер браузера изменён
Шаг[в процессе]: Инициализация теста
Шаг[выполнено]: Выполнена инициализация теста

-- Тест начинается -----
Шаг[в процессе]: Загрузка страницы
Шаг[успешно]: Страница загружена
Шаг[в процессе]: Ожидание элемента 15 секунд
Шаг[успешно]: Ожидание элемента - завершено (элемент отображается)
Шаг[в процессе]: Ввод значения в элемент
Шаг[успешно]: Значение введено в элемент
Шаг[в процессе]: Ожидание 2 секунд
Шаг[успешно]: Ожидание 2 секунд - завершено
Шаг[в процессе]: Ввод значения в элемент
Шаг[успешно]: Значение введено в элемент
Шаг[в процессе]: Ожидание 2 секунд
Шаг[успешно]: Ожидание 2 секунд - завершено
Шаг[в процессе]: Нажатие на элемент
Шаг[успешно]: Элемент нажат
Шаг[в процессе]: Ожидание элемента 5 секунд
Шаг[успешно]: Ожидание элемента - завершено (элемент отображается)
Шаг[в процессе]: Получение значения из элемент
Шаг[успешно]: Получено значение из элемента
Шаг[в процессе]: Проверка совпадения ожидаемого и актуального значения
Шаг[неудача]: Ожидаемое и актуальное значение не совпадают AssertEqualsAsync(Вы успешно авторизованы, Неверный логин или пароль)

Тест завершен - провально
-----
Скриншот image-20-01-2023-02-01-26.jpeg

Обновлен файл отчета Report-ExampleTest1.html

Создан отчет с результатами всех тестов

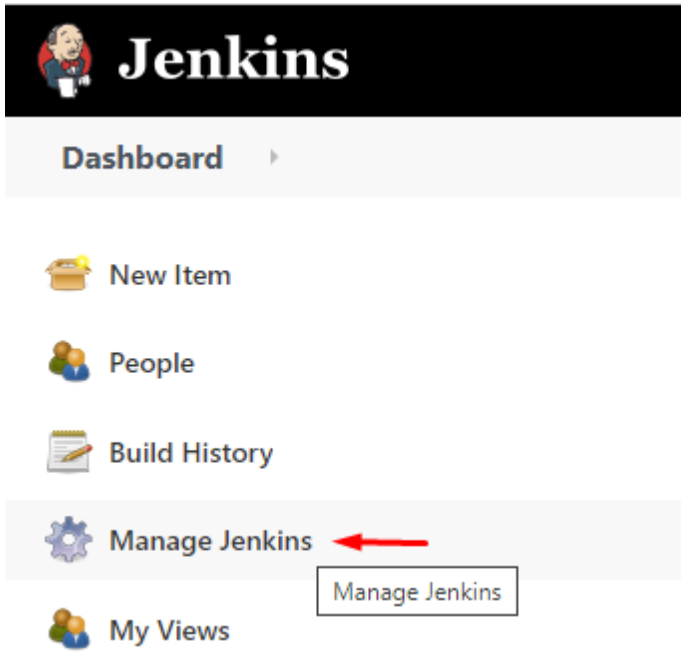
=====
Тестирование завершено ПРОВАЛЬНО

```

## 1. Настройка агента


1.

( ) slave .  
1) Jenkins Dashboard. Manage Jenkins




2) Configure Global Security

## Security

 **Configure Global Security**  
Secure Jenkins; define who is allowed to access/use the system.

## Security

 **Глобальные настройки безопасности**  
Настройка безопасности Jenkins, конфигурация прав для доступа и использования системы.

Agents

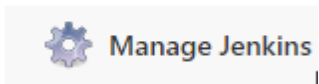
Random

### Agents

TCP port for inbound agents  Fixed :   Random  Disable

[Agent protocols...](#)

3) [Dashboard.](#) [Manage Jenkins](#)



### Manage Nodes and Clouds



#### Manage Nodes and Clouds

Add, remove, control and monitor the various nodes that Jenkins runs jobs on.



#### Управление средами сборки

Позволяет добавлять, удалять, контролировать и анализировать среды сборки.

4) [New Node](#)

**Jenkins**

Dashboard ▸ Nodes ▸

- Back to Dashboard
- Manage Jenkins
- New Node** ←
- Configure Clouds
- Node Monitoring

5) [Permanent Agent](#)

**Jenkins**

Dashboard > Nodes >

- Back to Dashboard
- Manage Jenkins
- New Node**
- Configure Clouds
- Node Monitoring

Node name: slave

**Permanent Agent**  
Adds a plain, permanent agent... example such as when you are...

**Copy Existing Node**  
Copy from: [ ]

OK

6)

"# of executors" -  
"Labels" -

Name	slave ←
Description	
# of executors	1
Remote root directory	C:\Autotests\t\jenkins\slave ←
Labels	slave ←
Usage	Use this node as much as possible
Launch method	Launch agent by connecting it to the master
	<input type="checkbox"/> Disable WorkDir
Custom WorkDir path	C:\Autotests\t\jenkins\slave ←
Internal data directory	remoting
	<input type="checkbox"/> Fail if workspace is missing
	<input type="checkbox"/> Use WebSocket
Availability	Keep this agent online as much as possible

### Node Properties

- Disable deferred wipeout on this node
- Environment variables
- Tool Locations

**Save** ←

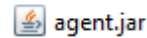
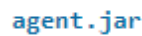
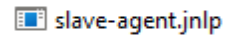
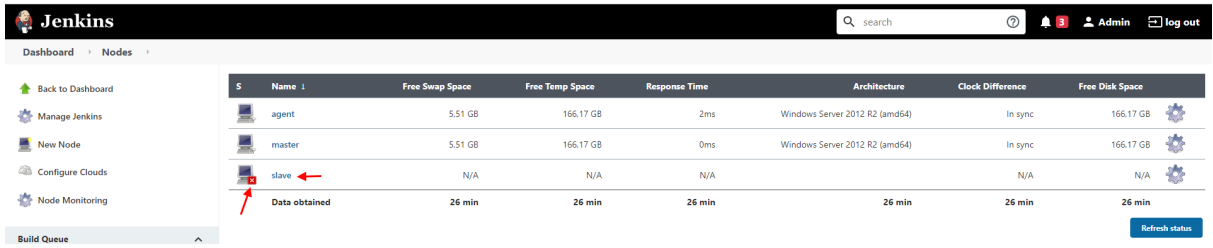
Usage

Only build jobs with label expressions matching this node

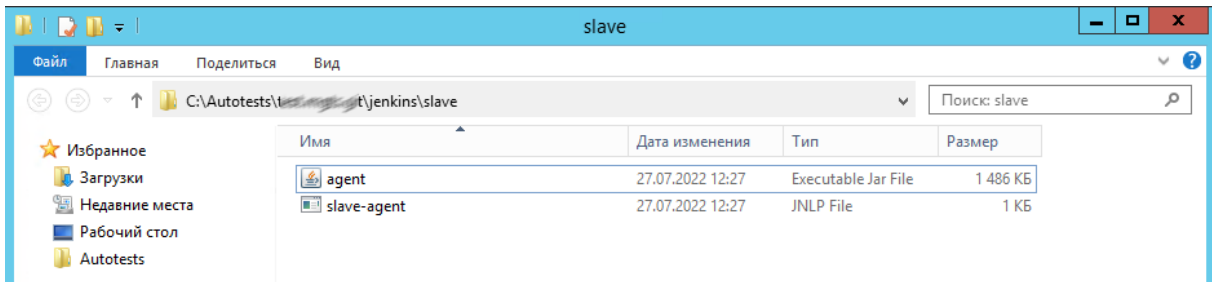
Save.

7)

slave



C:\Autotests\#####\jenkins\slave



( )

Run from agent command line:

```
java -jar agent.jar -jnlpUrl
http://192.168.201.1:8081/computer/slave/slave-agent.jnlp -
secret
aaaaaaaaaaaaabbbbbbbbbbbbbbccccccccccccccccddddd -
workDir "C:\Autotests\#####\jenkins\slave"
```

Run from agent command line, with the secret stored in a file:

```
echo aaaaaaaaaaaaaabbbbbbbbbbbbbbccccccccccccccccddddd
> secret-file java -jar agent.jar -jnlpUrl
```

```
http://192.168.201.1:8081/computer/slave/slave-agent.jnlp
secret @secret-file -workDir "C:\Autotests\#####\jenkins\slave"
```

### Dfile.encoding=UTF8

```
java -Dfile.encoding=UTF8 -jar agent.jar -jnlpUrl
http://192.168.201.1:8081/computer/slave/slave-agent.jnlp
secret
aaaaaaaaaaaaabbbbbbbbbbbbbbccccccccccccccccddddd
workDir "C:\Autotests\#####\jenkins\slave"
```



### Agent slave

Пометить

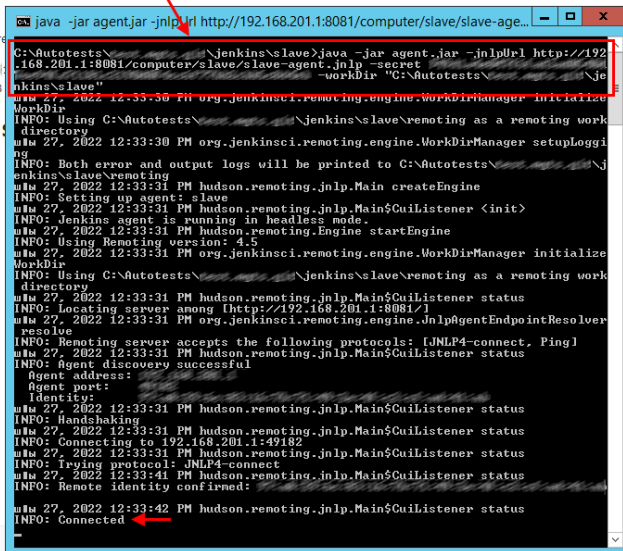
Connect agent to Jenkins one of these ways:

- Launch Launch agent from browser
- Выполнить в командной строке подчиненного узла

```
java -jar agent.jar -jnlpUrl http://192.168.201.1:8081/computer/slave/slave-agent.jnlp -secret -workDir
"C:\Autotests\#####\jenkins\slave"
Run from agent command line, with the secret stored in a file:
echo @secret-file > secret.txt
java -jar agent.jar -jnlpUrl http://192.168.201.1:8081/computer/slave/slave-agent.jnlp -secretFile secret.txt -workDir
"C:\Autotests\#####\jenkins\slave"
```

### Проекты, привязанные к...

Нет информации



Connector

slave

slave	5.03 GB	166.15 GB	86ms	Windows Server 2012 R2 (amd64)	In sync	166.15 GB
Data obtained	2 min 8 sec	2 min 9 sec	2 min 9 sec	2 min 9 sec	2 min 9 sec	2 min 9 sec

Refresh status

## 2. Настройка поддержки русского языка

### 2.

#### Console Output

```

Started by user Admin
Running as SYSTEM
Собирается удаленно на slave in workspace C:\Autotests\██████████\jenkins\slave\workspace\Hat - Тест
[Hat - Тест] $ cmd /c call C:\Users\sep\AppData\Local\Temp\2\jenkins1271870775057954817.bat

C:\Autotests\██████████\jenkins\slave\workspace\Hat - 'Г6в>cd C:\Hat

C:\Hat>Hat.exe HomeOrders_DeployTest.cs C:\Autotests\██████████\hat\Tests

Р°СЪР°СЪР·РμСЪ Hat PIPμСЪСЪР°СЪ 1.1.1 (19.07.2022)
Р-Р°Р°СЪСЪСЪР° Р±СЪР°СЪР·РμСЪР°... ←
РμСЪР°РμР°С, Р°С,Р°СЪС<С, (PIРμСЪСЪР°СЪ Р°СЪР°РμР°С,Р°: 1.1.1)
-----
РμСЪР°Р°Р°·Р°СЪР°Р° Р°СЪР°Р°Р°Р°Р°:
System.UnauthorizedAccessException: Р°С,Р°Р°Р°·Р°Р°Р°Р° Р° Р°Р°СЪС,СЪР°Р° Р°Р° Р°СЪС,Р° "C:\Autotests\██████████\hat\Tests\reports\Report-HomeOrders_DeployTest.html".
  PI System.IO.__Error.WinIOError(Int32 errorCode, String maybeFullPath)
  PI System.IO.File.InternalDelete(String path, Boolean checkHost)
  PI System.IO.File.Delete(String path)
  PI Hat.Report.SaveReport(Boolean testSuccess)
-----

```

#### **-Dfile.encoding=UTF8**

```

java -Dfile.encoding=UTF8 -jar agent.jar -jnlpUrl
http://192.168.201.1:8081/computer/slave/slave-agent.jnlp -
secret
aaaaaaaaaaaaabbbbbbbbbbbbbccccccccccccccccddddd -
workDir "C:\Autotests\#####\jenkins\slave"

```

#### Jenkins

Jenkins

:

arguments C:\Program Files\Jenkins\jenkins.xml  
 -Dfile.encoding=UTF8

```

<service>
...
<arguments>
-Xrs -Xmx256m -
Dhudson.lifecycle=udson.lifecycle.WindowsServiceLifecycle -
Dfile.encoding=UTF8 -jar "%BASE%\jenkins.war" --
httpPort=8080 --webroot="%BASE%\war"

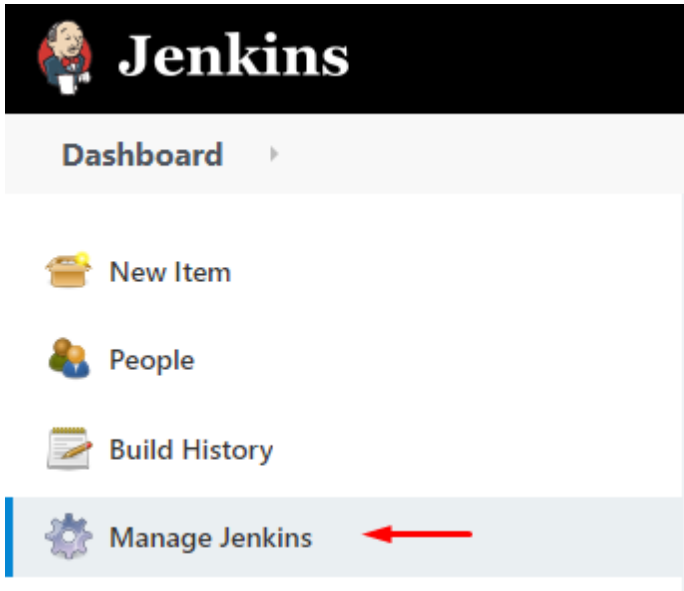
```

```
</arguments>  
...  
</service>
```

Jenkins

Manage Jenkins

System Configuration



## System Configuration



### Configure System

Configure global settings and paths.

" "

:

Environment variables

Name: **JAVA\_TOOL\_OPTIONS**  
Value: **-Dfile.encoding=UTF8**

## Global properties

 Disable deferred wipeout on this node ?
 Environment variables ?

List of variables

Name  ?Value 

Delete

Name Value 

Delete

Add

 Tool Locations

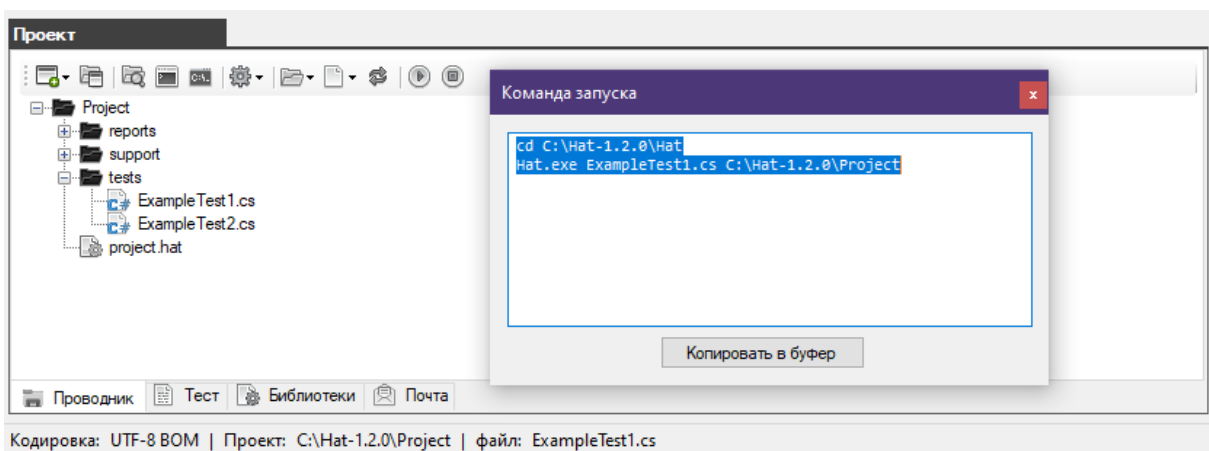
---

Created with the Personal Edition of HelpNDoc: [Free EPub and documentation generator](#)



---

### 3. Подключение автотеста к ноде.

#### 3.



Jenkins

S	W	Name ↓
		Hat - Tect

"Discard old builds"  
"Max # of builds to keep"

Discard old builds ?

Strategy:

Days to keep builds:

if not empty, build records are only kept up to this number of days

Max # of builds to keep:

if not empty, only up to this number of build records are kept

"Build"  
"Command"  
Hat.

"Execute Windows batch command".

Build

Execute Windows batch command X ?

Command:

See [the list of available environment variables](#)

slave ( "Restrict where this project can be run" )

Restrict where this project can be run ?

Label Expression:

[Label slave](#) matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Apply Save



S	W	Name ↓	Last Success	Last Failure	Last Duration
		Hat - Tectr	2 hr 11 min - #29	2 hr 12 min - #28	12 sec

	slave
1	Idle
2	Hat - Tectr #30



```

Hat          1.1.1 (19.07.2022)          ...
(           : 1.1.1)
-----
: System.UnauthorizedAccessException:
  "C:
\Autotests\#####\hat\Tests\reports\Report-
HomeOrders_DeployTest.html".
System.IO.__Error.WinIOError(Int32 errorCode, String
maybeFullPath)      System.IO.File.InternalDelete(String path,
Boolean checkHost)   System.IO.File.Delete(String path)
Hat.Report.SaveReport(Boolean testSuccess) - - - - -
-----
)
(
)

```



Created with the Personal Edition of HelpNDoc: [Full-featured EBook editor](#)

### Текстовый редактор кода

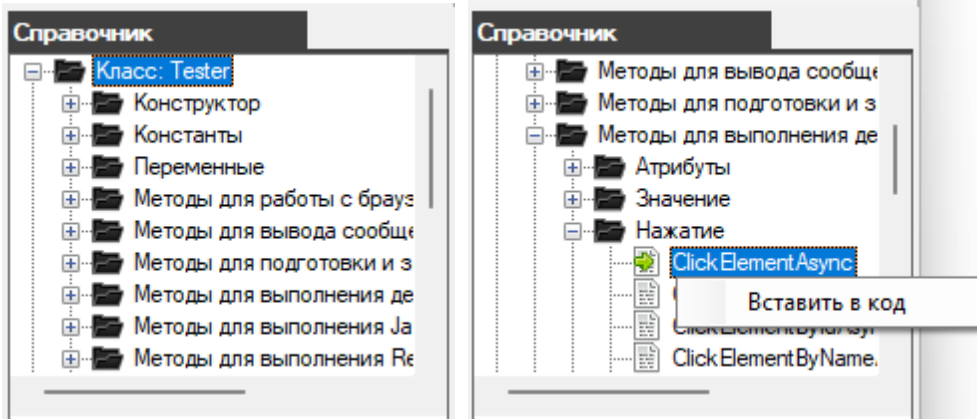


Created with the Personal Edition of HelpNDoc: [Full-featured EPub generator](#)



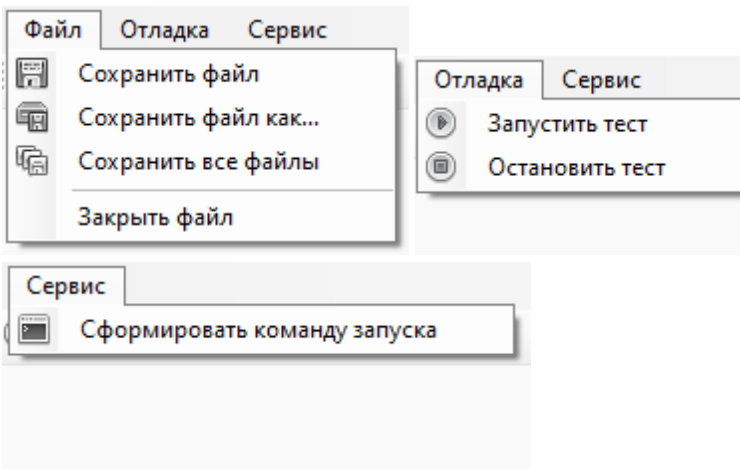
# HatFramework.

( )



- -
- -
- -
- -
- -
- -
- -

Windows (cmd)

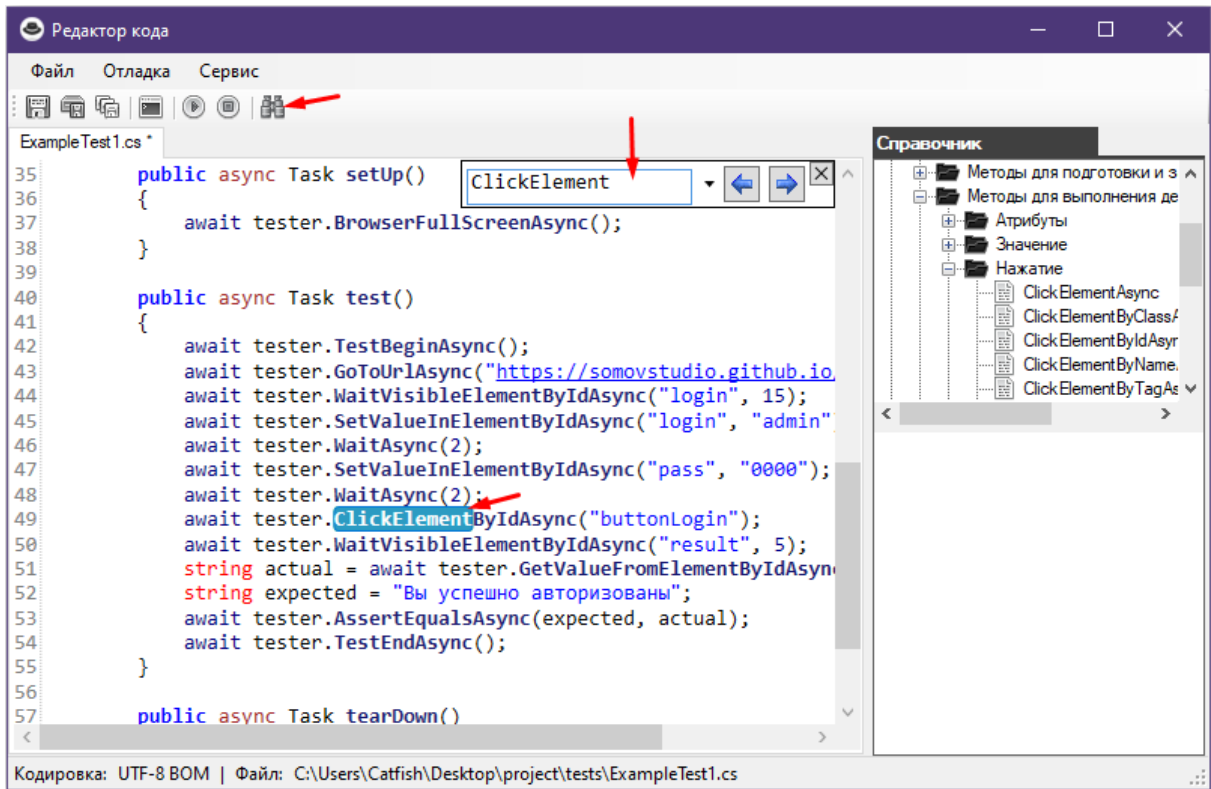


- -
- -

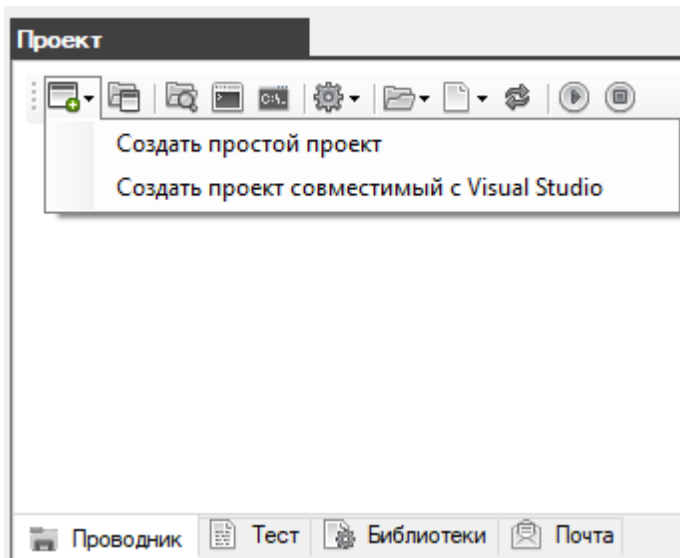
- 
- 
- :
- 
- 
- :
- 

Windows (cmd)

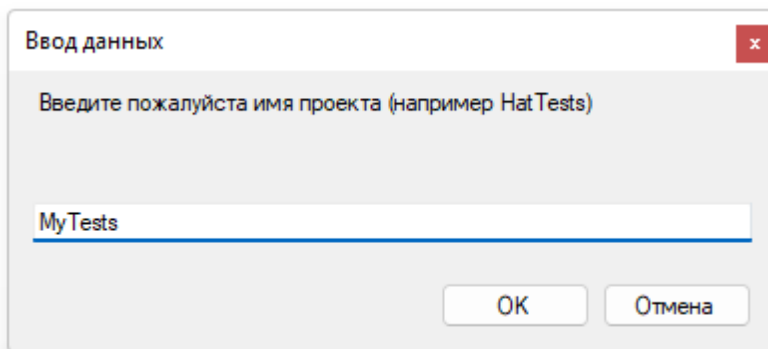
CTRL+F.







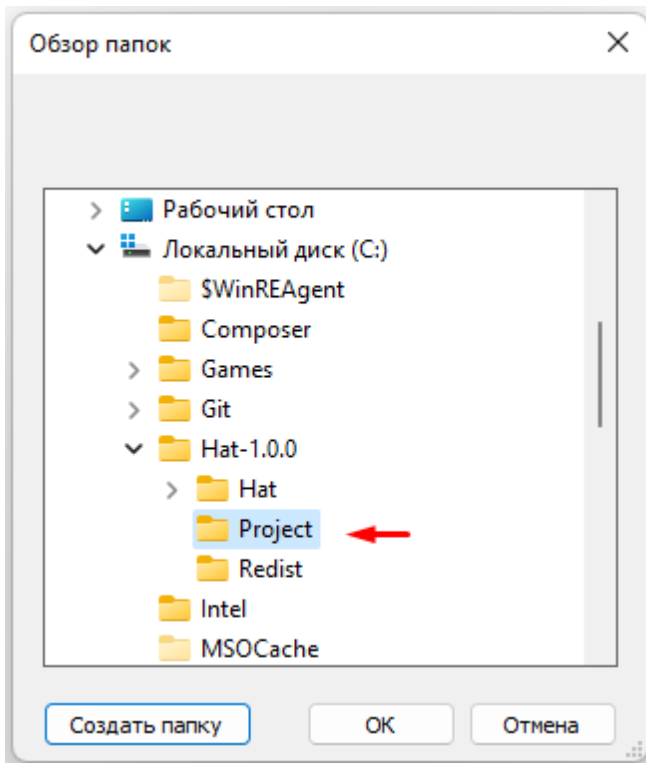
2.



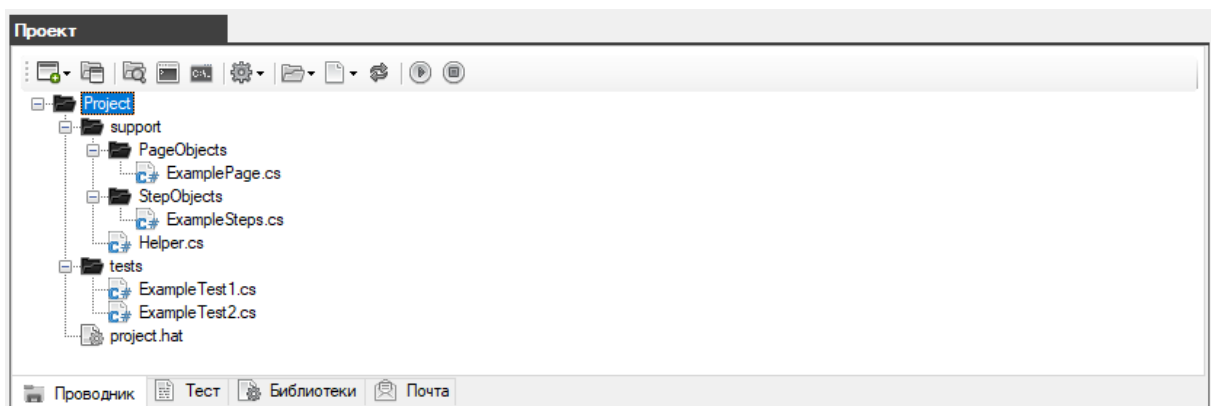
3.

Project  
OK

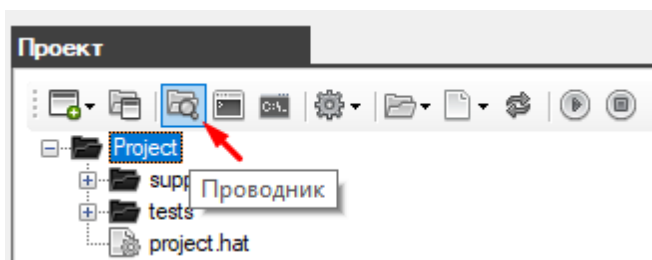
C:\Hat-1.0.0\



4. " " " "



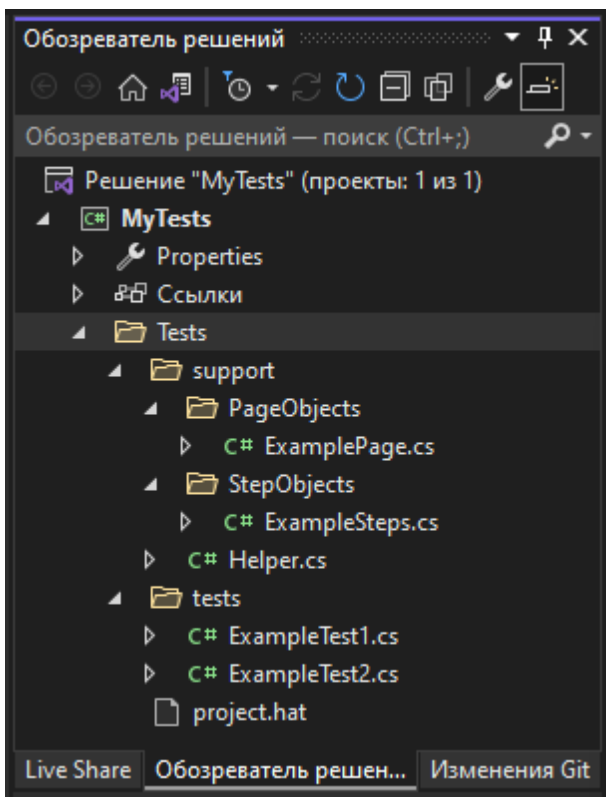
5. Studio Visual Studio ( Visual Studio " "



\*.sln ( MyTests.sln)

Имя	Дата изменения	Тип	Размер
Properties	22.07.2022 10:08	Папка с файлами	
Tests	22.07.2022 10:08	Папка с файлами	
.gitignore	22.07.2022 10:08	Файл "GITIGNORE"	1 КБ
MyTests.csproj	22.07.2022 10:08	C# Project file	4 КБ
MyTests.sln	22.07.2022 10:08	Visual Studio Solu...	2 КБ

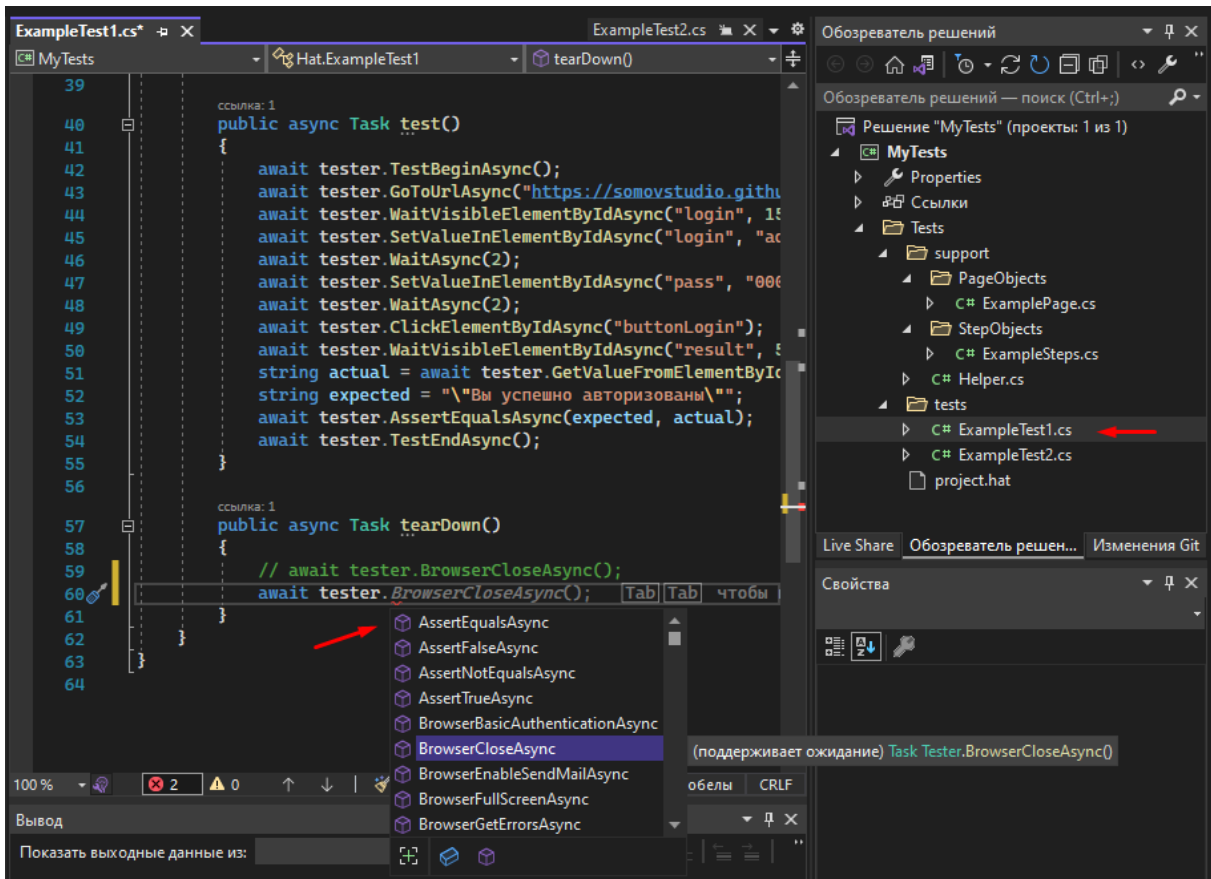
Visual Studio



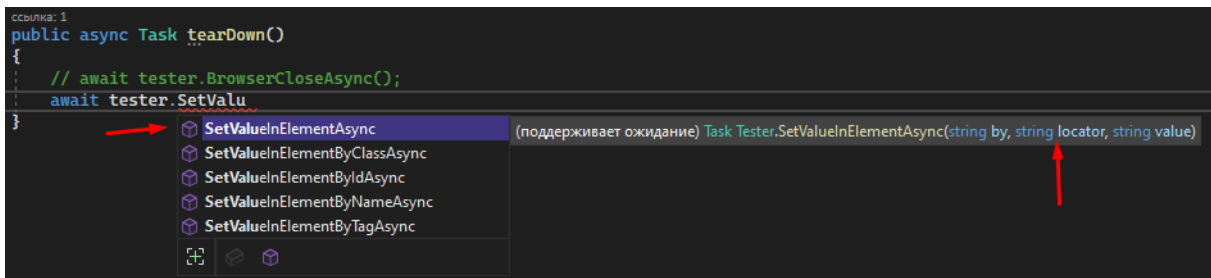
59

ExampleTest1.cs

BrowserCloseAsync()



Visual Studio



Visual Studio.

Created with the Personal Edition of HelpNDoc: [Free EPub and documentation generator](#)

## Структура автотеста

Created with the Personal Edition of HelpNDoc: [Produce electronic books easily](#)

## Простой автотест

C#.

:

```
using System;
using
System
.Collections.Generic;
using
System.ComponentModel;
using
System.Windows.Forms;
using System.Threading;
using
System.Threading.Tasks;
using System.IO;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using
System
.Text
.RegularExpressions;
using System.Net;
using System.Net.Http;
using
System.Net.Http.Headers;
using System.Reflection;
using Newtonsoft.Json;
using HatFramework;
```

HatFramework -

```
namespace Hat
{
    public class
    ExampleTest1
    {
    }
}
```

```
Tester tester;

public async void
Main(Form browserWindow)
{
    tester = new
    Tester(browserWindow);

    await setUp();
    await test();
    await
    tearDown();
}
```

Tester

Tester.

```
public async Task
setUp()
{
}

public async Task test()
{
    await
```

- setUp -
- task -
- tearDown -

( , )

```

tester.TestBeginAsync();

        await
tester.TestEndAsync();
}

public async Task
tearDown()
{
}

```

```

:
TestBeginAsync()
TestEndAsync()

```

```

:

```

```

: ExampleTest1.cs

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
using System.IO;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Reflection;
using Newtonsoft.Json;
using HatFramework;

namespace Hat
{
    public class ExampleTest1
    {
        Tester tester;

        public async void Main(Form browserWindow)
        {
            tester = new Tester(browserWindow);

            await setUp();
            await test();
            await tearDown();
        }

        public async Task setUp()
        {
            await tester.BrowserFullScreenAsync();
        }

        public async Task test()
        {
            await tester.TestBeginAsync();

```

```

        await
tester.GoToUrlAsync("https://somovstudio.github.io/test.html", 5);
        await tester.WaitVisibleElementByIdAsync("login", 15);
        await tester.SetValueInElementByIdAsync("login", "admin");
        await tester.WaitAsync(2);
        await tester.SetValueInElementByIdAsync("pass", "0000");
        await tester.WaitAsync(2);
        await tester.ClickElementByIdAsync("buttonLogin");
        await tester.WaitVisibleElementByIdAsync("result", 5);
        string actual = await
tester.GetValueFromElementByIdAsync("textarea");
        string expected = "                ";
        await tester.AssertEqualsAsync(expected, actual);
        await tester.TestEndAsync();
    }

    public async Task tearDown()
    {
        await tester.BrowserCloseAsync();
    }
}
}

```

Created with the Personal Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

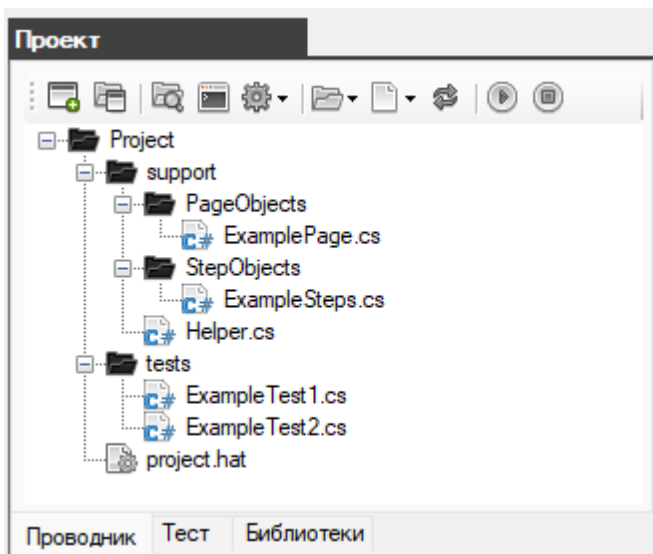
## Паттерн PageObjects и StepObjects

### PageObjects StepObjects

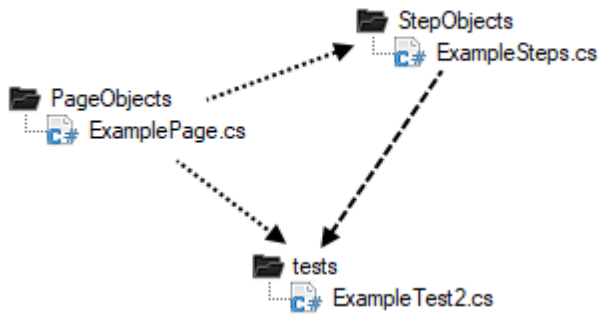
-

PageObjects

StepObjects.



:



- PageObjects ( ExamplePage.cs) -
- StepObjects ( ExampleSteps.cs) -  
Tester ( HatFramerowk)
- ( ExampleTest2.cs) -  
( PageObjects StepObjects.  
tester tester ExampleSteps  
Tester)

#### ExamplePage.cs

```

using System;
using HatFramework;

namespace Hat
{
    public static class ExamplePage
    {
        public static string URL =
@"https://somovstudio.github.io/test.html";
        public static string InputLogin = "login";
        public static string InputPass = "pass";
        public static string ButtonLogin = "buttonLogin";
        public static string Result = "result";
        public static string Textarea = "textarea";
    }
}

```

#### ExampleSteps.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
using System.IO;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;

```

```

using System.Text.RegularExpressions;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Reflection;
using Newtonsoft.Json;
using HatFramework;

namespace Hat
{
    public class ExampleSteps : Tester
    {
        public ExampleSteps(Form browserWindow): base(browserWindow) {}

        public async Task FillForm()
        {
            await this.WaitVisibleElementByIdAsync(ExamplePage.InputLogin,
15);
            await this.SetValueInElementByIdAsync(ExamplePage.InputLogin,
"admin");
            await this.WaitAsync(2);
            await this.SetValueInElementByIdAsync(ExamplePage.InputPass,
"0000");
            await this.WaitAsync(2);
        }
    }
}

```

#### ExampleTest2.cs

```

using System;
using System.IO;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;
using HatFramework;

namespace Hat
{
    public class ExampleTest2
    {
        ExampleSteps tester;

        public async void Main(Form browserWindow)
        {
            tester = new ExampleSteps(browserWindow);

            await setUp();
            await test();
            await tearDown();
        }
    }
}

```

```

public async Task setUp()
{
    await tester.BrowserFullScreenAsync();
}

public async Task test()
{
    await tester.TestBeginAsync();
    await tester.GoToUrlAsync(ExamplePage.URL, 5);
    await tester.FillForm();
    await tester.ClickElementByIdAsync(ExamplePage.ButtonLogin);
    await tester.WaitVisibleElementByIdAsync(ExamplePage.Result, 5);
    string actual = await
tester.GetValueFromElementByIdAsync(ExamplePage.Textarea);
    string expected = "                ";
    await tester.AssertEqualsAsync(expected, actual);
    await tester.TestEndAsync();
}

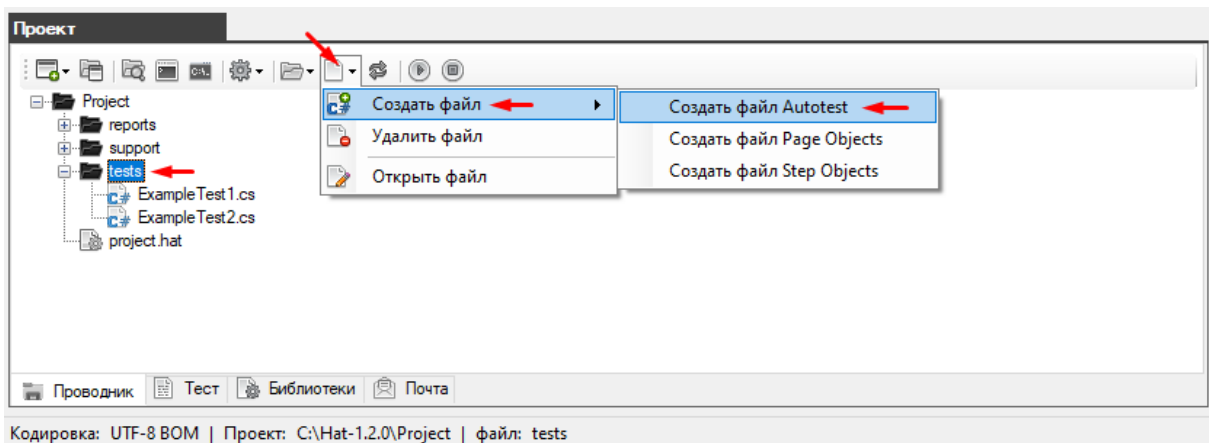
public async Task tearDown()
{
    await tester.BrowserCloseAsync();
}
}
}

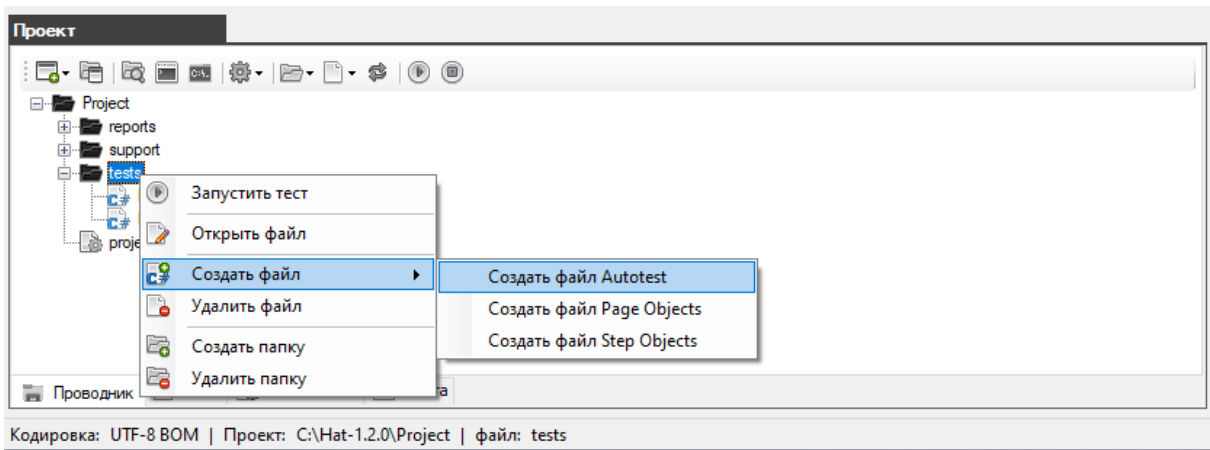
```

Created with the Personal Edition of HelpNDoc: [Easily create Qt Help files](#)

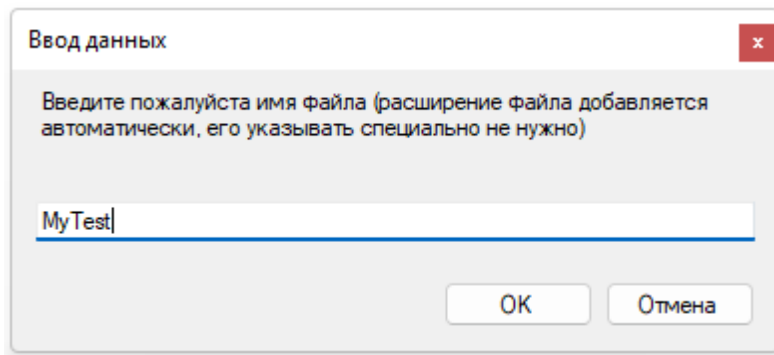
## Создание нового автотеста и паттернов

" ( " " )  
 " , " " "

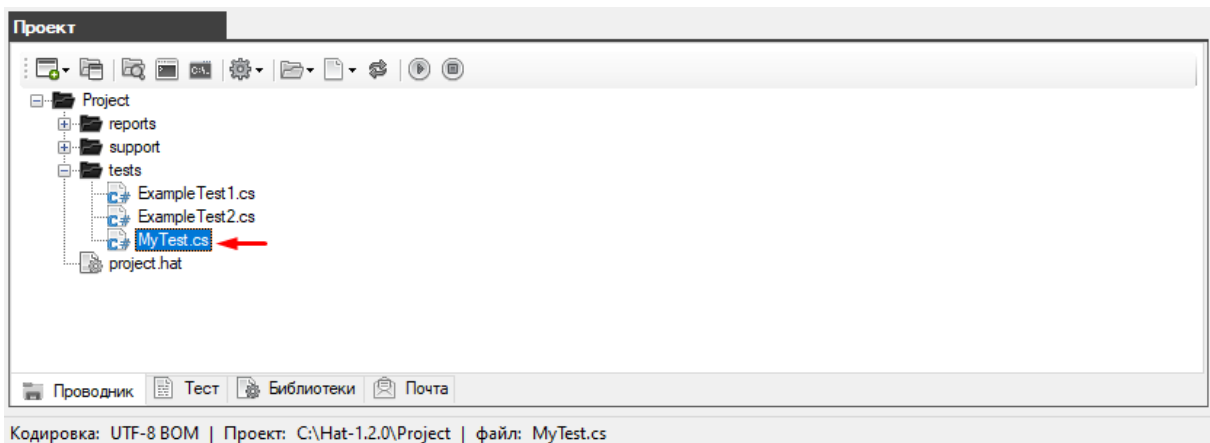




) (



" ( " ).



```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
using System.IO
    
```

```
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Reflection;
using Newtonsoft.Json;
using HatFramework;

namespace Hat
{
    public class MyTest
    {
        Tester tester;

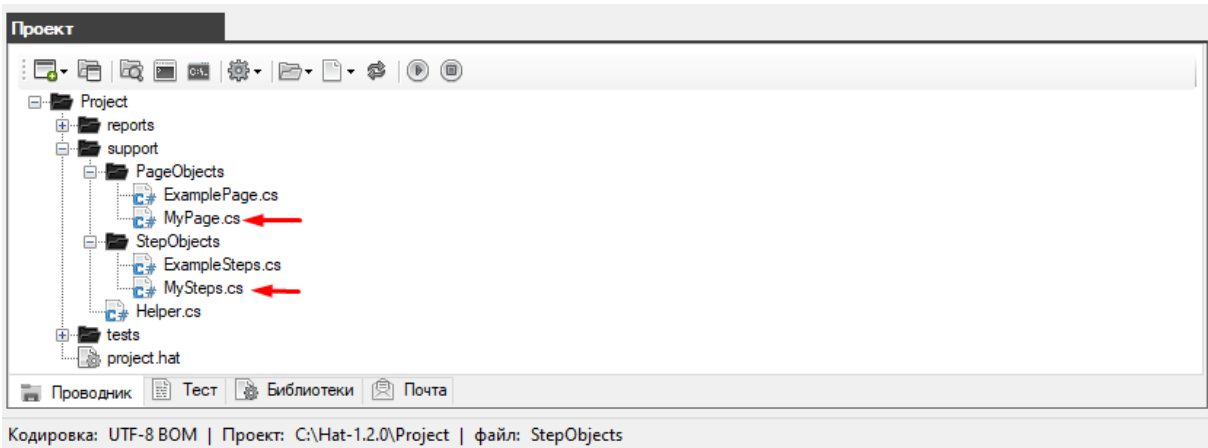
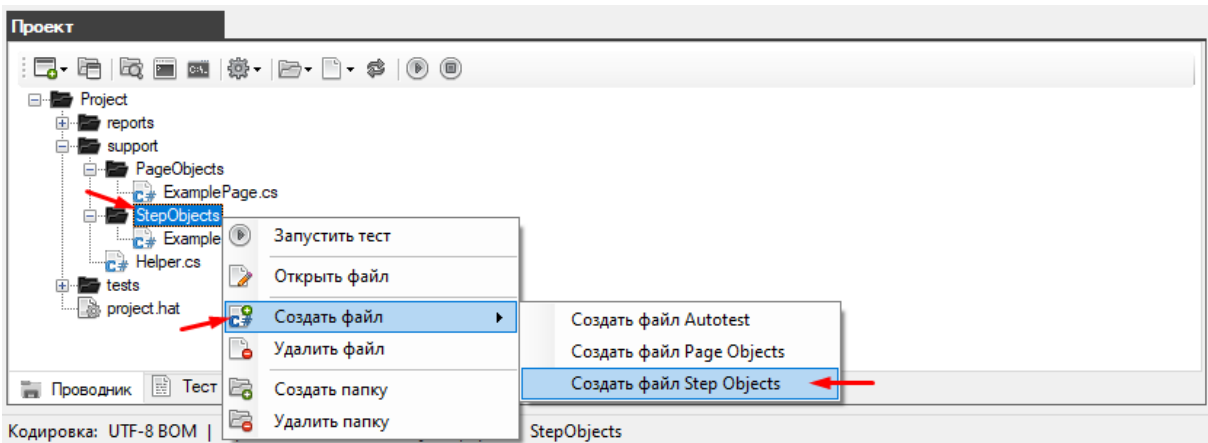
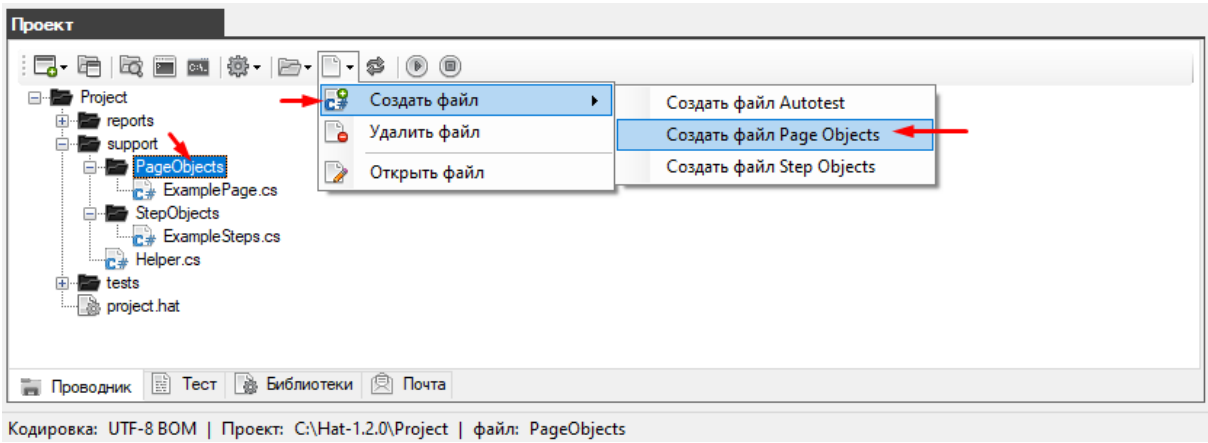
        public async void Main(Form browserWindow)
        {
            tester = new Tester(browserWindow);
            await setUp();
            await test();
            await tearDown();
        }

        public async Task setUp()
        {
        }

        public async Task test()
        {
        }

        public async Task tearDown()
        {
        }
    }
}
```

PageObjects StepObjects



: MyPage.cs	: MySteps.cs
<pre> using System; using HatFramework;  namespace Hat {     public static class MyPage     {         public static string URL = "https://test.com/";     } }                 </pre>	<pre> using System; using System.Collections.Generic; using System.ComponentModel; using System.Windows.Forms; using System.Threading; using System.Threading.Tasks; using System.IO; using System.Data; using System.Drawing;                 </pre>

<pre>         public static string ButtonLogin = "buttonLogin";     } } </pre>	<pre> using System.Linq; using System.Text; using System.Text.RegularExpressions; using System.Net; using System.Net.Http; using System.Net.Http.Headers; using System.Reflection; using Newtonsoft.Json; using HatFramework;  namespace Hat {     public class MySteps : Tester     {         public MySteps(Form browserWindow): base(browserWindow) {}          public async Task Test()         {             await this.AssertTrueAsync(true);         }     } } </pre>
--	--

---

Created with the Personal Edition of HelpNDoc: [Free help authoring tool](#)

---

## Встроенный фреймворк HatFramework

---

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

---

### Класс: Tester

---

Created with the Personal Edition of HelpNDoc: [Easily create Help documents](#)

---

#### Конструктор

---

Created with the Personal Edition of HelpNDoc: [Free CHM Help documentation generator](#)

---

Tester

Tester

**Описание:** основной класс работы автотестов

**Синтаксис:** Tester(Form browserForm)

**Пример:**

```

Tester tester;
public async void Main(Form browserWindow)
{
    tester = new Tester(browserWindow);
}

```

---

Created with the Personal Edition of HelpNDoc: [Produce electronic books easily](#)

---

## Константы

---

Created with the Personal Edition of HelpNDoc: [Easily create EPub books](#)

---

BY\_CSS

BY\_CSS

**Описание:** константа обозначает тип вводимого локатора

**Синтаксис:** BY\_CSS = "BY\_CSS"

**Пример:**

Tester.BY\_CSS

---

Created with the Personal Edition of HelpNDoc: [News and information about help authoring tools and software](#)

---

BY\_XPATH

BY\_XPATH

**Описание:** константа обозначает тип вводимого локатора

**Синтаксис:** BY\_XPATH = "BY\_XPATH"

**Пример:**

Tester.BY\_XPATH

---

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

---

IMAGE\_STATUS\_PROCESS

IMAGE\_STATUS\_PROCESS

**Описание:** индекс картинки которая обозначает статус в процессе

**Синтаксис:** IMAGE\_STATUS\_PROCESS = 0

**Пример:**

Tester.IMAGE\_STATUS\_PROCESS

---

Created with the Personal Edition of HelpNDoc: [Full-featured EPub generator](#)

---

IMAGE\_STATUS\_PASSED

IMAGE\_STATUS\_PASSED

**Описание:** индекс картинки которая обозначает статус успешно

**Синтаксис:** IMAGE\_STATUS\_PASSED = 1

**Пример:**

Tester.IMAGE\_STATUS\_PASSED

## IMAGE\_STATUS\_FAILED

### IMAGE\_STATUS\_FAILED

**Описание:** индекс картинки которая обозначает статус провальню

**Синтаксис:** IMAGE\_STATUS\_FAILED = 2

**Пример:**

Tester.IMAGE\_STATUS\_FAILED

## IMAGE\_STATUS\_MESSAGE

### IMAGE\_STATUS\_MESSAGE

**Описание:** индекс картинки которая обозначает статус сообщение

**Синтаксис:** IMAGE\_STATUS\_MESSAGE = 3

**Пример:**

Tester.IMAGE\_STATUS\_MESSAGE

## IMAGE\_STATUS\_WARNING

### IMAGE\_STATUS\_WARNING

**Описание:** индекс картинки которая обозначает статус предупреждение

**Синтаксис:** IMAGE\_STATUS\_WARNING = 4

**Пример:**

Tester.IMAGE\_STATUS\_WARNING

## IMAGE\_STATUS\_DEBUG

### IMAGE\_STATUS\_DEBUG

**Описание:** индекс картинки которая обозначает статус отладки

**Синтаксис:** IMAGE\_STATUS\_DEBUG = 5

**Пример:**

Tester.IMAGE\_STATUS\_DEBUG

COMPLETED

COMPLETED

**Описание:** константа обозначает статус выполнено

**Синтаксис:** COMPLETED = "COMPLETED"

**Пример:**

Tester.COMPLETED

---

Created with the Personal Edition of HelpNDoc: [Produce electronic books easily](#)

---

FAILED

FAILED

**Описание:** константа обозначает статус провальноно

**Синтаксис:** FAILED = "FAILED"

**Пример:**

Tester.FAILED

---

Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

---

PASSED

PASSED

**Описание:** константа обозначает статус успешно

**Синтаксис:** PASSED = "PASSED"

**Пример:**

Tester.PASSED

---

Created with the Personal Edition of HelpNDoc: [Generate Kindle eBooks with ease](#)

---

PROCESS

PROCESS

**Описание:** константа обозначает статус выполняется

**Синтаксис:** PROCESS = "PROCESS"

**Пример:**

Tester.PROCESS

---

Created with the Personal Edition of HelpNDoc: [Easily create EPub books](#)

---

STOPPED

STOPPED

**Описание:** константа обозначает статус остановлено

**Синтаксис:** STOPPED = "STOPPED"

**Пример:**

Tester.STOPPED

---

Created with the Personal Edition of HelpNDoc: [Easily create HTML Help documents](#)

---

WARNING

WARNING

**Описание:** константа обозначает статус предупреждение

**Синтаксис:** WARNING = "WARNING"

**Пример:**

Tester.WARNING

---

Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

---

DEBUG

DEBUG

**Описание:** константа обозначает статус отладка

**Синтаксис:** DEBUG = "DEBUG"

**Пример:**

Tester.DEBUG

---

Created with the Personal Edition of HelpNDoc: [Free help authoring environment](#)

---

DEFAULT

DEFAULT

**Описание:** константа обозначает тип кодировки для файла

**Синтаксис:** DEFAULT = "DEFAULT"

**Пример:**

```
string text = await tester.FileReadAsync(Tester.DEFAULT, "C:\\Hat\\file.txt");  
await tester.FileWriteAsync(text, Tester.DEFAULT, "C:\\Hat\\file_copy.txt");
```

---

Created with the Personal Edition of HelpNDoc: [Free EPub producer](#)

---

UTF8

UTF8

**Описание:** константа обозначает тип кодировки для файла

**Синтаксис:** UTF8 = "UTF-8"

**Пример:**

```
string text = await tester.FileReadAsync(Tester.UTF8, "C:\\Hat\\file.txt");
await tester.FileWriteAsync(text, Tester.UTF8, "C:\\Hat\\file_copy.txt");
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured Help generator](#)

---

## UTF8BOM

## UTF8BOM

**Описание:** константа обозначает тип кодировки для файла

**Синтаксис:** UTF8BOM = "UTF-8 BOM"

**Пример:**

```
string text = await tester.FileReadAsync(Tester.UTF8BOM, "C:\\Hat\\file.txt");
await tester.FileWriteAsync(text, Tester.UTF8BOM, "C:\\Hat\\file_copy.txt");
```

---

Created with the Personal Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

---

## WINDOWS1251

## WINDOWS1251

**Описание:** константа обозначает тип кодировки для файла

**Синтаксис:** WINDOWS1251 = "WINDOWS-1251"

**Пример:**

```
string text = await tester.FileReadAsync(Tester.WINDOWS1251, "C:\\Hat\\file.txt");
await tester.FileWriteAsync(text, Tester.WINDOWS1251, "C:\\Hat\\file_copy.txt");
```

---

Created with the Personal Edition of HelpNDoc: [Produce online help for Qt applications](#)

---

## Переменные

---

Created with the Personal Edition of HelpNDoc: [Free EBook and documentation generator](#)

---

## BrowserView

## BrowserView

**Описание:** переменная ссылается на объект представляющий область браузера

**Синтаксис:** WebView2 BrowserView

**Пример:**

```
tester.BrowserView.Refresh();
tester.BrowserView.Reload();
tester.BrowserView.Source = new Uri(url);
tester.BrowserView.Update();
```

---

Created with the Personal Edition of HelpNDoc: [Easily create EBooks](#)

---

BrowserWindow

[BrowserWindow](#)

**Описание:** переменная ссылается на окно приложения

**Синтаксис:** Form BrowserWindow

**Пример:**

```
Tester tester;
public async void Main(Form browserWindow)
{
    tester = new Tester(browserWindow);
    tester.BrowserWindow.Text = "Браузер Hat";
    tester.BrowserWindow.Width = 800;
    tester.BrowserWindow.Height = 600;
    tester.BrowserWindow.Close();
}
```

---

Created with the Personal Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

---

[Локаторы](#)

---

Created with the Personal Edition of HelpNDoc: [Full-featured EBook editor](#)

---

Locator

[Locator](#)

**Описание:** вспомогательный класс для создания локаторов

**Синтаксис:**

```
public class Locator
{
    public string name { get; set; }
    public string description { get; set; }
    public string type { get; set; }
    public string value { get; set; }
}
```

**Пример:**

```
Locator locator = new Locator();
locator.name = "inputLogin";
locator.description = "Поле для ввода логина";
locator.type = Tester.BY_ХПАТН;
locator.value = "//input[@id='login']";
```

---

Created with the Personal Edition of HelpNDoc: [Easily create Web Help sites](#)

---

AddLocator

[AddLocator](#)

**Описание:** метод создает локатор и добавляет его в общее хранилище локаторов

**Синтаксис:** AddLocator(string name, string type, string value, string description)

**Возвращаемое значение:** отсутствует

**Пример:**

```
tester.AddLocator("inputLogin", Tester.BY_ХПАТН, "//input[@id='login']", "Поле для ввода  
логина");
```

---

Created with the Personal Edition of HelpNDoc: [Free EBook and documentation generator](#)

---

## GetLocators

### GetLocators

**Описание:** метод возвращает хранилища локаторов

**Синтаксис:** GetLocators()

**Возвращаемое значение:** Dictionary<string, Locator>

**Пример:**

```
Dictionary<string, Locator> locators = tester.GetLocators();
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured Help generator](#)

---

## GetLocator

### GetLocator

**Описание:** метод возвращает локатор из хранилища локаторов в виде объекта класс которого Locator

**Синтаксис:** GetLocator(string name)

**Возвращаемое значение:** объект (Locator)

**Пример:**

```
await tester.WaitVisibleElementAsync(tester.GetLocator("inputLogin"), 25);
```

---

Created with the Personal Edition of HelpNDoc: [Easily create PDF Help documents](#)

---

## GetLocatorValue

### GetLocatorValue

**Описание:** метод возвращает значение локатор из хранилища локаторов

**Синтаксис:** GetLocatorValue(string name)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string value = tester.GetLocatorValue("inputLogin");
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured EBook editor](#)

---

## GetCountLocators

### GetCountLocators

**Описание:** метод возвращает количество локатор в хранилище локаторов

**Синтаксис:** GetCountLocators()

**Возвращаемое значение:** число (int)

**Пример:**

```
int count = tester.GetCountLocators();
```

---

Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

---

## ClearLocators

### ClearLocators

**Описание:** метод очищает хранилище локаторов

**Синтаксис:** ClearLocators()

**Возвращаемое значение:** отсутствует

**Пример:**

```
tester.ClearLocators();
```

---

Created with the Personal Edition of HelpNDoc: [Write EPub books for the iPad](#)

---

## RemoveLocator

### RemoveLocator

**Описание:** метод удаляет локатора из хранилища локаторов и возвращает логический результат true или false

**Синтаксис:** RemoveLocator(string name)

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
bool result = tester.RemoveLocator("inputLogin");
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured Kindle eBooks generator](#)

---

## Методы для работы с браузером

---

Created with the Personal Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

---

## BrowserBasicAuthenticationAsync

### BrowserBasicAuthenticationAsync

**Описание:** метод навигации браузера выполняет действие вперед с указанным ожиданием в секундах

**Синтаксис:** BrowserBasicAuthenticationAsync(string user, string pass)

**Возвращаемое значение:** отсутствует

**Пример:**

```
Tester tester = new Tester(browserForm);
await tester.TestBeginAsync();
await tester.BrowserBasicAuthenticationAsync("user", "pass");
await tester.GoToUrlAsync("http://test.ru/basic_auth.html", 5);
await tester.TestEndAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Produce electronic books easily](#)

---

**BrowserClearNetworkAsync****BrowserClearNetworkAsync**

**Описание:** метод выполняет очистку событий network в браузере

**Синтаксис:** BrowserClearNetworkAsync()

**Возвращаемое значение:** строка (string)

**Пример:**

```
string events = await tester.BrowserClearNetworkAsync();
tester.ConsoleMsg(events);
```

---

Created with the Personal Edition of HelpNDoc: [Easy CHM and documentation editor](#)

---

**BrowserCloseAsync****BrowserCloseAsync**

**Описание:** метод закрывает окно браузера

**Синтаксис:** BrowserCloseAsync()

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.BrowserCloseAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Easy CHM and documentation editor](#)

---

**BrowserEnableSendMailAsync****BrowserEnableSendMailAsync**

**Описание:** метод включает опцию отправки отчета на почту в случае провала в работе автотеста

**Синтаксис:** BrowserEnableSendMailAsync(bool byFailure = true, bool bySuccess = true)

**Возвращаемое значение:** отсутствует

**Пример:**

```
Tester tester = new Tester(browserForm);
await tester.BrowserEnableSendMailAsync(true, false); // только в случае провала
await tester.BrowserEnableSendMailAsync(false, true); // только в случае успеха
await tester.BrowserEnableSendMailAsync(); // в обоих случаях
```

```
await tester.TestBeginAsync();  
...  
await tester.TestEndAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured Help generator](#)

---

## BrowserFullScreenAsync

### BrowserFullScreenAsync

**Описание:** метод устанавливает полноэкранный размер браузера

**Синтаксис:** BrowserFullScreenAsync()

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.BrowserFullScreenAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Easy CHM and documentation editor](#)

---

## BrowserScreenshotAsync

### BrowserScreenshotAsync

**Описание:** метод делает снимок экрана браузера

**Синтаксис:** BrowserScreenshotAsync(string filename)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string screenshot = await tester.BrowserScreenshotAsync(null);  
screenshot = await tester.BrowserScreenshotAsync("");
```

```
screenshot = await tester.BrowserScreenshotAsync("screenshot.jpg");  
screenshot = await tester.BrowserScreenshotAsync("C:\\Users\\User\\Desktop\\MyTests\\  
\\reports\\screenshots\\test_screenshot.jpg");
```

```
tester.ConsoleMsg(screenshot);
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

---

## BrowserSizeAsync

### BrowserSizeAsync

**Описание:** метод устанавливает размер браузера

**Синтаксис:** BrowserSizeAsync(int width, int height)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.BrowserSizeAsync(800, 600);
```

---

Created with the Personal Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

---

## BrowserGetUserAgentAsync

### BrowserGetUserAgentAsync

**Описание:** метод возвращает строчное значение User-Agent браузера

**Синтаксис:** BrowserGetUserAgentAsync()

**Возвращаемое значение:** строка (string)

**Пример:**

```
string ua = await tester.BrowserGetUserAgentAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

---

## BrowserSetUserAgentAsync

### BrowserSetUserAgentAsync

**Описание:** метод устанавливает значение User-Agent для браузера

**Синтаксис:** BrowserSetUserAgent(string value)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.TestBeginAsync();
await tester.BrowserSetUserAgent("my user-agent");
await tester.TestEndAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Free Qt Help documentation generator](#)

---

## BrowserGetErrorsAsync

### BrowserGetErrorsAsync

**Описание:** метод возвращает список ошибок и предупреждений браузера

**Синтаксис:** BrowserGetErrorsAsync()

**Возвращаемое значение:** список (List)

**Пример:**

```
List<string> errors = await tester.BrowserGetErrorsAsync();
foreach (string error in errors)
{
    tester.ConsoleMsg(error);
}
```

---

Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

---

## BrowserGetNetworkAsync

### BrowserGetNetworkAsync

**Описание:** метод возвращает в формате json все текущие сообщения из network

**Синтаксис:** BrowserGetNetworkAsync()

**Возвращаемое значение:** строка (string)

**Пример:**

```
string events = await tester.BrowserGetNetworkAsync();
tester.ConsoleMsg(events);
```

---

Created with the Personal Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

---

**BrowserGoBackAsync****BrowserGoBackAsync**

**Описание:** метод навигации браузера выполняет действие назад с указанным ожиданием в секундах

Флаг abortLoadAfterTime принудительно завершает загрузку страницы после завершения таймера

**Синтаксис:** BrowserGoBackAsync(int sec, bool abortLoadAfterTime = false)

**Возвращаемое значение:** отсутствует

**Пример:**

```
Tester tester = new Tester(browserForm);
await tester.TestBeginAsync();
await tester.GoToUrlAsync("https://www.yahoo.com/", 5);
string currentUrl = await tester.GetUrlAsync();
await tester.AssertEqualsAsync("https://www.yahoo.com/", currentUrl);
```

```
await tester.GoToUrlAsync("https://yandex.ru/", 5);
currentUrl = await tester.GetUrlAsync();
await tester.AssertEqualsAsync("https://yandex.ru/", currentUrl);
```

```
await tester.BrowserGoBackAsync(10);
currentUrl = await tester.GetUrlAsync();
await tester.AssertEqualsAsync("https://www.yahoo.com/", currentUrl);
```

```
await tester.BrowserGoForwardAsync(10);
currentUrl = await tester.GetUrlAsync();
await tester.AssertEqualsAsync("https://yandex.ru/", currentUrl);
await tester.TestEndAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

---

**BrowserGoForwardAsync****BrowserGoForwardAsync**

**Описание:** метод навигации браузера выполняет действие вперед с указанным ожиданием в секундах

Флаг abortLoadAfterTime принудительно завершает загрузку страницы после завершения таймера

**Синтаксис:** BrowserGoForwardAsync(int sec, bool abortLoadAfterTime = false)

**Возвращаемое значение:** отсутствует

**Пример:**

```
Tester tester = new Tester(browserForm);
await tester.TestBeginAsync();
await tester.GoToUrlAsync("https://www.yahoo.com/", 5);
string currentUrl = await tester.GetUrlAsync();
await tester.AssertEqualsAsync("https://www.yahoo.com/", currentUrl);
```

```
await tester.GoToUrlAsync("https://yandex.ru/", 5);
currentUrl = await tester.GetUrlAsync();
await tester.AssertEqualsAsync("https://yandex.ru/", currentUrl);
```

```
await tester.BrowserGoBackAsync(10);
currentUrl = await tester.GetUrlAsync();
await tester.AssertEqualsAsync("https://www.yahoo.com/", currentUrl);
```

```
await tester.BrowserGoForwardAsync(10);
currentUrl = await tester.GetUrlAsync();
await tester.AssertEqualsAsync("https://yandex.ru/", currentUrl);
await tester.TestEndAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Easily create Help documents](#)

---

## BrowserPageReloadAsync

### BrowserPageReloadAsync

**Описание:** метод выполняет перезагрузку открытой страницы в браузере  
Флаг abortLoadAfterTime принудительно завершает загрузку страницы после завершения таймера

**Синтаксис:** BrowserPageReloadAsync(int sec, bool abortLoadAfterTime = false)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.BrowserPageReloadAsync(25);
```

---

Created with the Personal Edition of HelpNDoc: [Write EPub books for the iPad](#)

---

## Методы для вывода сообщений

---

Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

---

## ConsoleMsg

### ConsoleMsg

**Описание:** метод выводит сообщение в консоли браузера

**Синтаксис:**

ConsoleMsg(string message)

ConsoleMsg(string messageRus, string messageEng)

**Возвращаемое значение:** отсутствует

**Пример:**

```
tester.ConsoleMsg("Текст на Русском языке", "The text is in English");
```

---

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

## ConsoleMsgError

### ConsoleMsgError

**Описание:** метод выводит сообщение об ошибке в системную консоль и консоль браузера

**Синтаксис:** ConsoleMsgError(string message)

**Возвращаемое значение:** отсутствует

**Пример:**

```
try
{
}
catch (Exception ex)
{
    tester.ConsoleMsgError(ex.ToString());
}
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured EBook editor](#)

## ClearMessage

### ClearMessage

**Описание:** метод очищает сообщения в таблице вывода процесса выполнения тесте

**Синтаксис:** ClearMessage()

**Возвращаемое значение:** отсутствует

**Пример:**

```
tester.ClearMessage();
```

---

Created with the Personal Edition of HelpNDoc: [Create iPhone web-based documentation](#)

## Description

### Description

**Описание:** метод устанавливает описание теста которое потом выводится в отчете и письме

**Синтаксис:** Description(string text)

**Возвращаемое значение:** отсутствует

**Пример:**

```
public async Task setUp()
{
    tester.Description("Тест проверяет авторизацию на сайте");
}
```

---

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

---

## DisableDebugInReport

### DisableDebugInReport

**Описание:** метод отключает вывод отладочных сообщений SendMessageDebug в отчет

**Синтаксис:** DisableDebugInReport()

**Возвращаемое значение:** отсутствует

**Пример:**

```
tester.DisableDebugInReport();
```

---

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

---

## SendMessage

### SendMessage

**Описание:** метод выводит сообщения в таблице процесса выполнения теста  
При выполнении в системную консоль выводится action если status равен FAILED, WARNING, "", null, в противном случае выводится только status и comment (без action).

**Синтаксис:** SendMessage(string action, string status, string comment)

**Пример:**

```
tester.SendMessage("текст действия", Tester.PROCESS, "текст комментария");
```

```
/* Примеры вывода в консоль */
```

```
// действие комментарий
```

```
tester.SendMessage("действия", null, "комментарий");
```

```
tester.SendMessage("действия", "", "комментарий");
```

```
// Шаг[в процессе]: комментарий
```

```
tester.SendMessage("действия", Tester.PROCESS, "комментарий");
```

```
// Шаг[успешно]: комментарий
```

```
tester.SendMessage("действия", Tester.PASSED, "комментарий");
```

```
// Шаг[выполнено]: комментарий
tester.SendMessage("действия", Tester.COMPLETED, "комментарий");
// Шаг[остановлено]: комментарий
tester.SendMessage("действия", Tester.STOPPED, "комментарий");

// Шаг[неудача]: действие комментарий
tester.SendMessage("действия", Tester.FAILED, "комментарий");
// Шаг[предупреждение]: действие комментарий
tester.SendMessage("действия", Tester.WARNING, "комментарий");
```

---

Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

---

## SendMessageDebug

### SendMessageDebug

**Описание:** метод выводит отладочное сообщение которое может быть отключено для вывода в отчет и письмо

При выполнении в системную консоль выводится action если status равен FAILED, WARNING, "" , null, в противном случае выводится только status и comment (без action).

**Синтаксис:** SendMessageDebug(string actionRus, string actionEng, string status, string commentRus, string commentEng, int image)

**Пример:**

```
// Шаг[в процессе]: комментарий
tester.SendMessageDebug("действия", "action", Tester.PROCESS, "комментарий", "comment", Tester.IMAGE_STATUS_PROCESS);
// Шаг[успешно]: комментарий
tester.SendMessageDebug("действия", "action", Tester.PASSED, "комментарий", "comment", Tester.IMAGE_STATUS_PASSED);
// Шаг[выполнено]: комментарий
tester.SendMessageDebug("действия", "action", Tester.COMPLETED, "комментарий", "comment", Tester.IMAGE_STATUS_MESSAGE);
// Шаг[остановлено]: комментарий
tester.SendMessageDebug("действия", "action", Tester.STOPPED, "комментарий", "comment", Tester.IMAGE_STATUS_WARNING);
// Шаг[неудача]: действие комментарий
tester.SendMessageDebug("действия", "action", Tester.FAILED, "комментарий", "comment", Tester.IMAGE_STATUS_FAILED);
// Шаг[предупреждение]: действие комментарий
tester.SendMessageDebug("действия", "action", Tester.WARNING, "комментарий", "comment", Tester.IMAGE_STATUS_WARNING);
```

---

Created with the Personal Edition of HelpNDoc: [Free CHM Help documentation generator](#)

---

## Методы для подготовки и завершения тестирования

---

Created with the Personal Edition of HelpNDoc: [Easily create HTML Help documents](#)

---

### TestBeginAsync

#### TestBeginAsync

**Описание:** метод подготовки к тесту (каждый тест обязательно должен начинаться этим методом)

**Синтаксис:** TestBeginAsync()

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.TestBeginAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

---

### TestEndAsync

#### TestEndAsync

**Описание:** метод завершения теста (каждый тест обязательно должен заканчиваться этим методом), именно этот метод выполняет сохранение отчета и отправку его по почте если такая опция будет настроена и включена.

**Синтаксис:** TestEndAsync()

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.TestEndAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

---

### TestStopAsync

#### TestStopAsync

**Описание:** метод принудительно останавливает процесс тестирования

**Синтаксис:** TestStopAsync()

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.TestStopAsync();
```

---

Created with the Personal Edition of HelpNDoc: [iPhone web sites made easy](#)

---

### GetTestResult

#### GetTestResult

**Описание:** метод возвращает статус выполняемого теста

**Синтаксис:** GetTestResult()

**Возвращаемое значение:** строка (string) значение которой проверяется константами:

Tester.PASSED, Tester.FAILED, Tester.PROCESS.

**Пример:**

```
Tester tester = new Tester(browserForm);
await tester.TestBeginAsync();
...
if(tester.GetTestResult() == Tester.PROCESS) { }
...
await tester.TestEndAsync();
if(tester.GetTestResult() == Tester.FAILED) { }
if(tester.GetTestResult() == Tester.PASSED) { }
```

---

Created with the Personal Edition of HelpNDoc: [Write EPub books for the iPad](#)

---

## DefineTestStop

### DefineTestStop

**Описание:** метод проверяет статус процесса (остановлен или нет)

**Синтаксис:** DefineTestStop()

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
if (tester.DefineTestStop() == true) return; // значит тест остановлен
```

---

Created with the Personal Edition of HelpNDoc: [Free HTML Help documentation generator](#)

---

## Методы для выполнения действий

---

Created with the Personal Edition of HelpNDoc: [Full-featured EBook editor](#)

---

## Атрибуты

---

Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

---

## GetAttributeFromElementAsync

### GetAttributeFromElementAsync

**Описание:** метод возвращает строчное значение из указанного атрибута в выбранном элементе

**Синтаксис:**

GetAttributeFromElementAsync(string by, string locator, string attribute)

GetAttributeFromElementAsync(Locator locator, string attribute)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string value = await tester.GetAttributeFromElementAsync(Tester.BY_CSS, "input", "name");
```

```
string value = await tester.GetAttributeFromElementAsync(Tester.BY_XPATH, "//input", "name");
```

```
string value = await tester.GetAttributeFromElementAsync(tester.GetLocator("locatorName"),  
"name");
```

## GetAttributeFromElementByClassAsync

### GetAttributeFromElementByClassAsync

**Описание:** метод возвращает строчное значение из указанного атрибута в выбранном элементе

**Синтаксис:** `GetAttributeFromElementByClassAsync(string _class, int index, string attribute)`

**Возвращаемое значение:** строка (string)

#### Пример:

```
string value = await tester.GetAttributeFromElementByClassAsync("my-element", 0, "href");
```

---

Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

---

GetAttributeFromElementByIdAsync

GetAttributeFromElementByIdAsync

**Описание:** метод возвращает строчное значение из указанного атрибута в выбранном элементе

**Синтаксис:** GetAttributeFromElementByIdAsync(string id, string attribute)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string value = await tester.GetAttributeFromElementByIdAsync("MyElement", "href");
```

---

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

---

## GetAttributeFromElementByNameAsync

### GetAttributeFromElementByNameAsync

**Описание:** метод возвращает строчное значение из указанного атрибута в выбранном элементе

**Синтаксис:** GetAttributeFromElementByNameAsync(string name, int index, string attribute)

**Возвращаемое значение:** строка (string)

#### Пример:

```
string value = await tester.GetAttributeFromElementByNameAsync("MyElement", 0, "href");
```

## GetAttributeFromElementByTagAsync

### GetAttributeFromElementByTagAsync

**Описание:** метод возвращает строчное значение из указанного атрибута в выбранном элементе

**Синтаксис:** GetAttributeFromElementByTagAsync(string tag, int index, string attribute)

**Возвращаемое значение:** строка (string)

#### Пример:

```
string value = await tester.GetAttributeFromElementByTagAsync("a", 0, "href");
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

---

## GetAttributeFromElementsAsync

### GetAttributeFromElementsAsync

**Описание:** метод возвращает список значений указанного атрибута из множества элементов

**Синтаксис:**

```
GetAttributeFromElementsAsync(string by, string locator, string attribute)
```

```
GetAttributeFromElementsAsync(Locator locator, string attribute)
```

**Возвращаемое значение:** список (List)

**Пример:**

```
List<string> values = await tester.GetAttributeFromElementsAsync(Tester.BY_CSS, "input",
"name");
if(values != null)
{
    foreach (string attr in values)
        tester.ConsoleMsg(attr);
}
```

```
List<string> values = await tester.GetAttributeFromElementsAsync(Tester.BY_XPATH, "//input",
"name");
if(values != null)
{
    foreach (string attr in values)
        tester.ConsoleMsg(attr);
}
```

```
List<string> values = await
tester.GetAttributeFromElementsAsync(tester.GetLocator("locatorName"), "name");
if(values != null)
{
    foreach (string attr in values)
        tester.ConsoleMsg(attr);
}
```

GetAttributeFromElementsByClassAsync

GetAttributeFromElementsByClassAsync

**Описание:** метод возвращает список значений указанного атрибута из множества элементов

**Синтаксис:** GetAttributeFromElementsByClassAsync(string \_class, string attribute)

**Возвращаемое значение:** список (List)

**Пример:**

```
List<string> values = await tester.GetAttributeFromElementsByClassAsync("text-field", "name");
foreach (string value in values)
{
    tester.ConsoleMsg(value);
}
```

## GetAttributeFromElementsByNameAsync

### GetAttributeFromElementsByNameAsync

**Описание:** метод возвращает список значений указанного атрибута из множества элементов

**Синтаксис:** `GetAttributeFromElementsByNameAsync(string name, string attribute)`

**Возвращаемое значение:** список (List)

#### Пример:

```
List<string> values = await tester.GetAttributeFromElementsByNameAsync("link", "href");
foreach (string value in values)
{
    tester.ConsoleMsg(value);
}
```

## GetAttributeFromElementsByTagAsync

### GetAttributeFromElementsByTagAsync

**Описание:** метод возвращает список значений указанного атрибута из множества элементов

**Синтаксис:** `GetAttributeFromElementsByTagAsync(string tag, string attribute)`

**Возвращаемое значение:** список (List)

#### Пример:

```
List<string> values = await tester.GetAttributeFromElementsByTagAsync("a", "href");
foreach (string value in values)
{
    tester.ConsoleMsg(value);
}
```

## SetAttributeInElementAsync

### SetAttributeInElementAsync

**Описание:** метод вставляет атрибут со значением в указанный элемент

**Синтаксис:**

SetAttributeInElementAsync(string by, string locator, string attribute, string value)

SetAttributeInElementAsync(Locator locator, string attribute, string value)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetAttributeInElementAsync(Tester.BY_CSS, "#auth > h2", "name", "test");  
await tester.SetAttributeInElementAsync(Tester.BY_XPATH, "//div[@id='auth']//h2", "name",  
"test");
```

```
await tester.SetAttributeInElementAsync(tester.GetLocator("locatorName"), "name", "test");
```

SetAttributeInElementByClassAsync

SetAttributeInElementByClassAsync

**Описание:** метод вставляет атрибут со значением в указанный элемент

**Синтаксис:** SetAttributeInElementByClassAsync(string \_class, int index, string attribute, string value)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetAttributeInElementByClassAsync("my-element", 0, "value", "значение");
```

---

Created with the Personal Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

---

## SetAttributeInElementByIdAsync

### SetAttributeInElementByIdAsync

**Описание:** метод вставляет атрибут со значением в указанный элемент

**Синтаксис:** SetAttributeInElementByIdAsync(string id, string attribute, string value)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetAttributeInElementByIdAsync("MyElement", "value", "значение");
```

---

Created with the Personal Edition of HelpNDoc: [Free help authoring environment](#)

---

## SetAttributeInElementByNameAsync

### SetAttributeInElementByNameAsync

**Описание:** метод вставляет атрибут со значением в указанный элемент

**Синтаксис:** SetAttributeInElementByNameAsync(string name, int index, string attribute, string value)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetAttributeInElementByNameAsync("MyElement", 0, "value", "значение");
```

---

Created with the Personal Edition of HelpNDoc: [Free EPub and documentation generator](#)

---

## SetAttributeInElementByTagAsync

### SetAttributeInElementByTagAsync

**Описание:** метод вставляет атрибут со значением в указанный элемент

**Синтаксис:** SetAttributeInElementByTagAsync(string tag, int index, string attribute, string value)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetAttributeInElementByTagAsync("input", 0, "value", "значение");
```

## SetAttributeInElementsAsync

### SetAttributeInElementsAsync

**Описание:** метод вставить атрибут с указанным значением в множество элементов и в результате возвращает список

**Синтаксис:**

```
SetAttributeInElementsAsync(string by, string locator, string attribute, string value)
```

```
SetAttributeInElementsAsync(Locator locator, string attribute, string value)
```

**Возвращаемое значение:** список (List)

**Пример:**

```
List<string> values = await tester.SetAttributeInElementsAsync(Tester.BY_CSS, "input", "class",
"test-class");
foreach (string value in values)
{
    tester.ConsoleMsg(value);
}
```

```
List<string> values = await tester.SetAttributeInElementsAsync(Tester.BY_XPATH, "//input",
"class", "test-class");
foreach (string value in values)
{
    tester.ConsoleMsg(value);
}
```

```
List<string> values = await
tester.SetAttributeInElementsAsync(tester.GetLocator("locatorName"), "class", "test-class");
foreach (string value in values)
{
    tester.ConsoleMsg(value);
}
```

## SetAttributeInElementsByClassAsync

### SetAttributeInElementsByClassAsync

**Описание:** метод вставить атрибут с указанным значением в множество элементов и в результате возвращает список

**Синтаксис:** `SetAttributeInElementsByClassAsync(string _class, string attribute, string value)`

**Возвращаемое значение:** список (List)

#### Пример:

```
List<string> values = await tester.SetAttributeInElementsByClassAsync("text-field", "value",
"test");
foreach (string value in values)
{
    tester.ConsoleMsg(value);
}
```

## SetAttributeInElementsByNameAsync

### SetAttributeInElementsByNameAsync

**Описание:** метод вставить атрибут с указанным значением в множество элементов и в результате возвращает список

**Синтаксис:** SetAttributeInElementsByNameAsync(string name, string attribute, string value)

**Возвращаемое значение:** список (List)

#### Пример:

```
List<string> values = await tester.SetAttributeInElementsByNameAsync("link", "href",  
"www.test.ru");  
foreach (string value in values)  
{  
    tester.ConsoleMsg(value);  
}
```

## SetAttributeInElementsByTagAsync

### SetAttributeInElementsByTagAsync

**Описание:** метод вставить атрибут с указанным значением в множество элементов и в результате возвращает список

**Синтаксис:** SetAttributeInElementsByTagAsync(string tag, string attribute, string value)

**Возвращаемое значение:** список (List)

#### Пример:

```
List<string> values = await tester.SetAttributeInElementsByTagAsync("a", "href",  
"www.test.ru");  
foreach (string value in values)  
{  
    tester.ConsoleMsg(value);  
}
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

---

#### Значение

---

Created with the Personal Edition of HelpNDoc: [Write EPub books for the iPad](#)

---

## GetValueFromElementAsync

### GetValueFromElementAsync

**Описание:** метод возвращает значение из указанного элемента

**Синтаксис:**

GetValueFromElementAsync(string by, string locator)

GetValueFromElementAsync(Locator locator)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string value = await tester.GetValueFromElementAsync(Tester.BY_CSS, "input[id='login']");
```

```
string value = await tester.GetValueFromElementAsync(Tester.BY_XPATH,  
"//input[@id='login']");
```

```
string value = await tester.GetValueFromElementAsync(tester.GetLocator("locatorName"));
```

GetValueFromElementByClassAsync

GetValueFromElementByClassAsync

**Описание:** метод возвращает значение из указанного элемента

**Синтаксис:** GetValueFromElementByClassAsync(string \_class, int index)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string value = await tester.GetValueFromElementByClassAsync("my-element", 0);
```

---

Created with the Personal Edition of HelpNDoc: [Free Qt Help documentation generator](#)

---

## GetValueFromElementByIdAsync

### GetValueFromElementByIdAsync

**Описание:** метод возвращает значение из указанного элемента

**Синтаксис:** GetValueFromElementByIdAsync(string id)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string value = await tester.GetValueFromElementByIdAsync("MyElement");
```

## GetValueFromElementByNameAsync

### GetValueFromElementByNameAsync

**Описание:** метод возвращает значение из указанного элемента

**Синтаксис:** GetValueFromElementByNameAsync(string name, int index)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string value = await tester.GetValueFromElementByNameAsync("MyElement", 0);
```

---

Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

---

GetValueFromElementByTagAsync

GetValueFromElementByTagAsync

**Описание:** метод возвращает значение из указанного элемента

**Синтаксис:** GetValueFromElementByTagAsync(string tag, int index)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string value = await tester.GetValueFromElementByTagAsync("input", 0);
```

---

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

---

## SetValueInElementAsync

### SetValueInElementAsync

**Описание:** метод вставляет значение в указанный элемент

**Синтаксис:**

```
SetValueInElementAsync(string by, string locator, string value)
```

```
SetValueInElementAsync(Locator locator, string value)
```

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetValueInElementAsync(Tester.BY_CSS, "input[id='login']", "admin");
```

```
await tester.SetValueInElementAsync(Tester.BY_CSS, "input[id='pass']", "0000");
```

```
await tester.SetValueInElementAsync(Tester.BY_XPATH, "//input[@id='login']", "admin");
```

```
await tester.SetValueInElementAsync(Tester.BY_XPATH, "//input[@id='pass']", "0000");
```

```
await tester.SetValueInElementAsync(tester.GetLocator("locatorName"), "admin");
```

```
await tester.SetValueInElementAsync(tester.GetLocator("locatorName"), "0000");
```

SetValueInElementByClassAsync

SetValueInElementByClassAsync

**Описание:** метод вставляет значение в указанный элемент

**Синтаксис:** SetValueInElementByClassAsync(string \_class, int index, string value)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetValueInElementByClassAsync("my-element", 0, "значение");
```

## SetValueInElementByIdAsync

### SetValueInElementByIdAsync

**Описание:** метод вставляет значение в указанный элемент

**Синтаксис:** SetValueInElementByIdAsync(string id, string value)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetValueInElementByIdAsync("MyElement", "значение");
```

## SetValueInElementByNameAsync

### SetValueInElementByNameAsync

**Описание:** метод вставляет значение в указанный элемент

**Синтаксис:** SetValueInElementByNameAsync(string name, int index, string value)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetValueInElementByNameAsync("MyElement", 0, "значение");
```

---

Created with the Personal Edition of HelpNDoc: [News and information about help authoring tools and software](#)

---

## SetValueInElementByTagAsync

### SetValueInElementByTagAsync

**Описание:** метод вставляет значение в указанный элемент

**Синтаксис:** SetValueInElementByTagAsync(string tag, int index, string value)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetValueInElementByTagAsync("input", 0, "значение");
```

---

Created with the Personal Edition of HelpNDoc: [Qt Help documentation made easy](#)

---

Нажатие

---

Created with the Personal Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

---

## ClickElementAsync

### ClickElementAsync

**Описание:** метод выполняет нажатие на элемент

**Синтаксис:**

ClickElementAsync(string by, string locator)

ClickElementAsync(Locator locator)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.ClickElementAsync(Tester.BY_CSS, "#auth #buttonLogin");
```

```
await tester.ClickElementAsync(Tester.BY_XPATH,  
"//div[@id='auth']//input[@id='buttonLogin']");
```

```
await tester.ClickElementAsync(tester.GetLocator("locatorName"));
```

---

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

---

ClickElementByClassAsync

ClickElementByClassAsync

**Описание:** метод выполняет нажатие на элемент

**Синтаксис:** ClickElementByClassAsync(string \_class, int index);

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.ClickElementByClassAsync("my-element", 0)
```

ClickElementByIdAsync

ClickElementByIdAsync

**Описание:** метод выполняет нажатие на элемент

**Синтаксис:** ClickElementByIdAsync(string id)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.ClickElementByIdAsync("MyElement");
```

---

Created with the Personal Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

---

ClickElementByNameAsync

ClickElementByNameAsync

**Описание:** метод выполняет нажатие на элемент

**Синтаксис:** ClickElementByNameAsync(string name, int index)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.ClickElementByNameAsync("MyElement", 0);
```

ClickElementByTagAsync

ClickElementByTagAsync

**Описание:** метод выполняет нажатие на элемент

**Синтаксис:** ClickElementByTagAsync(string tag, int index)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.ClickElementByTagAsync("a", 0);
```

## FocusElementAsync

### FocusElementAsync

**Описание:** метод переключает фокус на указанный элемент

**Синтаксис:**

```
FocusElementAsync(string by, string locator)
```

```
FocusElementAsync(Locator locator)
```

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.FocusElementAsync(Tester.BY_CSS, "input[id='login']");
```

```
await tester.SetValueInElementAsync(Tester.BY_CSS, "input[id='login']", "admin");
```

```
await tester.FocusElementAsync(Tester.BY_XPATH, "//input[@id='login']");
```

```
await tester.SetValueInElementAsync(Tester.BY_XPATH, "//input[@id='login']", "admin");
```

```
await tester.FocusElementAsync(tester.GetLocatorValue("locatorName"));
```

```
await tester.SetValueInElementAsync(tester.GetLocator("locatorName"), "admin");
```

IsClickableElementAsync

IsClickableElementAsync

**Описание:** метод определяет кликабельность элемента и возвращает true или false

**Синтаксис:**

IsClickableElementAsync(string by, string locator)

IsClickableElementAsync(Locator locator)

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
Tester tester = new Tester(browserForm);
```

```
await tester.TestBeginAsync();
```

```
await tester.GoToUrlAsync("https://somovstudio.github.io/test.html", 5);
```

```
bool clickable = await tester.IsClickableElementAsync(Tester.BY_XPATH,
```

```
"//*[@id='buttonLogin']");
```

```
await tester.AssertTrueAsync(clickable);
```

```
await tester.TestEndAsync();
```

```
bool clickable = await tester.IsClickableElementAsync(tester.GetLocator("locatorName"));
```

```
await tester.AssertTrueAsync(clickable);
```

---

Created with the Personal Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

## ScrollToElementAsync

### ScrollToElementAsync

**Описание:** метод выполняет прокрутку к указанному элементу (параметр behaviorSmooth определяет плавность прокрутки)

**Синтаксис:**

```
ScrollToElementAsync(string by, string locator, bool behaviorSmooth = false)
```

```
ScrollToElementAsync(Locator locator, bool behaviorSmooth = false)
```

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.ScrollToElementAsync(Tester.BY_CSS, "body > footer", true);
```

```
await tester.ScrollToElementAsync(Tester.BY_XPATH, "/html/body/footer", true);
```

```
await tester.ScrollToElementAsync(tester.GetLocator("locatorName"), true);
```

---

Created with the Personal Edition of HelpNDoc: [Easily create HTML Help documents](#)

---

## Объекты

---

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

---

## GetElementAsync

### GetElementAsync

**Описание:** метод возвращает элемент в виде объект класс которого [HTMLElement](#)

**Синтаксис:**

```
GetElementAsync(string by, string locator)
```

```
GetElementAsync(Locator locator)
```

**Возвращаемое значение:** объект (HTMLElement)

**Примеры:**

```
HTMLElement element = await tester.GetElementAsync(tester.GetLocator("locatorName"));
await element.ClickAsync();
```

```
HTMLElement element = await tester.GetElementAsync(Tester.BY_CSS, "#auth #buttonLogin");
await element.ClickAsync();
```

```
HTMLElement element = await tester.GetElementAsync(Tester.BY_XPATH,
"//div[@id='auth']//input[@id='buttonLogin']");
await element.ClickAsync();
```

```
HTMLElement element = await tester.GetElementAsync(Tester.BY_XPATH, "//*[@id='MyFile']");
tester.ConsoleMsg("ID: " + element.Id);
```

```
HTMLElement element = await tester.GetElementAsync(Tester.BY_XPATH, "//h1");
string text = await element.GetTextAsync();
tester.ConsoleMsg(text);
await element.SetTextAsync("TEST");
```

```
HTMLElement element = await tester.GetElementAsync(Tester.BY_XPATH,
"//*[@id='MyInput']");
await element.SetValueAsync("Тестирование");
string value = await element.GetValueAsync();
tester.ConsoleMsg(value);
```

```
HTMLElement element = await tester.GetElementAsync(Tester.BY_XPATH, "//h1");
await element.SetAttributeAsync("class", "my-class");
string attrClass = await element.GetAttributeAsync("class");
tester.ConsoleMsg(attrClass);
```

```
HTMLElement element = await tester.GetElementAsync(Tester.BY_XPATH, "//h1");
string html = await element.GetHtmlAsync();
tester.ConsoleMsg(html);
```

```
HTMLElement element = await tester.GetElementAsync(Tester.BY_XPATH, "//h1");
await element.SetHtmlAsync("<div>Это тест</div>");
```

```
HTMLElement element = await tester.GetElementAsync(Tester.BY_XPATH, "/html/body/footer");
await element.ScrollToAsync();
```

```
HTMLElement element = await tester.GetElementAsync(Tester.BY_XPATH, "//h1");  
await element.WaitVisibleAsync(2);  
await element.WaitNotVisibleAsync(2);
```

## GetFrameAsync

## GetFrameAsync

**Описание:** метод возвращает элемент в виде объект класс которого [FRAMEElement](#)

**Синтаксис:** GetFrameAsync(int index)

**Возвращаемое значение:** объект (FRAMEElement)

**Пример:**

```
Tester tester = new Tester(browserForm);
```

```
await tester.TestBeginAsync();
```

```
await tester.GoToUrlAsync("https://somovstudio.github.io/test2.html", 5);
```

```
FRAMEElement frame = await tester.GetFrameAsync(0);
```

```
tester.ConsoleMsg("Index: " + frame.Index);
```

```
tester.ConsoleMsg("Name: " + frame.Name);
```

```
string name = await frame.GetAttributeFromElementAsync(Tester.BY_XPATH,
"//input[@id='login']", "name");
```

```
List<string> values = await frame.GetAttributeFromElementsAsync(Tester.BY_XPATH, "//input",
"name");
```

```
if (values != null)
```

```
{
```

```
    foreach (string attr in values)
```

```
        tester.ConsoleMsg(attr);
```

```
}
```

```
await frame.SetAttributeInElementAsync(Tester.BY_XPATH, "//input[@id='buttonLogin']",
"name", "NameButtonLogin");
```

```
await frame.SetAttributeInElementsAsync(Tester.BY_XPATH, "//input", "name", "test");
```

```
await frame.SetValueInElementAsync(Tester.BY_XPATH, "//input[@id='login']", "Тестировщик");
```

```
string value = await frame.GetValueFromElementAsync(Tester.BY_XPATH,
"//input[@id='login']");
```

```
await frame.ClickElementAsync(Tester.BY_XPATH, "//*[@id='buttonLogin']");
```

```
bool result = await frame.IsClickableElementAsync(Tester.BY_XPATH, "//*[@id='buttonLogin']");
```

```
await frame.ScrollToElementAsync(Tester.BY_XPATH, "//*[@id='buttonLogin']", true);
```

```
int result = await frame.GetCountElementsAsync(Tester.BY_XPATH, "//input");
```

```
string html = await frame.GetHtmlFromElementAsync(Tester.BY_XPATH,
"//*[@id='buttonLogin']");

await frame.SetHtmlInElementAsync(Tester.BY_XPATH, "//*[@id='auth']/h2", "<h2>Тестовый
заголовок</h2>");

await frame.WaitNotVisibleElementAsync(Tester.BY_XPATH, "//*[@id='result']", 5);
await frame.ClickElementAsync(Tester.BY_XPATH, "//*[@id='buttonLogin']");
await frame.WaitVisibleElementAsync(Tester.BY_XPATH, "//*[@id='result']", 5);

bool result = await frame.FindElementAsync(Tester.BY_XPATH, "//*[@id='result']", 5);

bool result = await frame.FindVisibleElementAsync(Tester.BY_XPATH, "//*[@id='result']", 5);

string title = await frame.GetTitleAsync();

string url = await frame.GetUrlAsync();

await frame.SetTextInElementAsync(Tester.BY_XPATH, "//*[@id='auth']/h2", "Это тест");
string text = await frame.GetTextFromElementAsync(Tester.BY_XPATH, "//*[@id='auth']/h2");

await frame.SelectOptionAsync(Tester.BY_XPATH, "//*[@id='MySelect']",
FRAMEElement.BY_INDEX, "2");
await frame.SelectOptionAsync(Tester.BY_XPATH, "//*[@id='MySelect']",
FRAMEElement.BY_VALUE, "Mobile");
await frame.SelectOptionAsync(Tester.BY_XPATH, "//*[@id='MySelect']",
FRAMEElement.BY_TEXT, "Other");

string index = await frame.GetOptionAsync(Tester.BY_XPATH, "//*[@id='MySelect']",
FRAMEElement.BY_INDEX);
string value = await frame.GetOptionAsync(Tester.BY_XPATH, "//*[@id='MySelect']",
FRAMEElement.BY_VALUE);
string text = await frame.GetOptionAsync(Tester.BY_XPATH, "//*[@id='MySelect']",
FRAMEElement.BY_TEXT);
```

GetCountElementsAsync

GetCountElementsAsync

**Описание:** метод возвращает количество найденных элементов

**Синтаксис:**

GetCountElementsAsync(string by, string locator)

GetCountElementsAsync(Locator locator)

**Возвращаемое значение:** целое число (int)

**Пример:**

```
int count = await tester.GetCountElementsAsync(Tester.BY_CSS, "input");
```

```
int count = await tester.GetCountElementsAsync(Tester.BY_XPATH, "//input");
```

```
int count = await tester.GetCountElementsAsync(tester.GetLocator("locatorName"));
```

GetCountElementsByClassAsync

GetCountElementsByClassAsync

**Описание:** метод возвращает количество найденных элементов

**Синтаксис:** GetCountElementsByClassAsync(string \_class)

**Возвращаемое значение:** целое число (int)

**Пример:**

```
int count = await tester.GetCountElementsByClassAsync("my-element");
```

---

Created with the Personal Edition of HelpNDoc: [iPhone web sites made easy](#)

---

GetCountElementsByNameAsync

GetCountElementsByNameAsync

**Описание:** метод возвращает количество найденных элементов

**Синтаксис:** GetCountElementsByNameAsync(string name)

**Возвращаемое значение:** целое число (int)

**Пример:**

```
int count = await tester.GetCountElementsByNameAsync("MyElement");
```

GetCountElementsByTagAsync

GetCountElementsByTagAsync

**Описание:** метод возвращает количество найденных элементов

**Синтаксис:** GetCountElementsByTagAsync(string tag)

**Возвращаемое значение:** целое число (int)

**Пример:**

```
int count = await tester.GetCountElementsByTagAsync("h2");
```

---

Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

---

GetHtmlFromElementAsync

GetHtmlFromElementAsync

**Описание:** метод возвращает html представление объекта в строчном выражении

**Синтаксис:**

GetHtmlFromElementAsync(string by, string locator)

GetHtmlFromElementAsync(Locator locator)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string html = await tester.GetHtmlFromElementAsync(Tester.BY_CSS, "#auth > h2");
```

```
string html = await tester.GetHtmlFromElementAsync(Tester.BY_XPATH,  
"//div[@id='auth']//h2");
```

```
string html = await tester.GetHtmlFromElementAsync(tester.GetLocator("locatorName"));
```

GetHtmlFromElementByClassAsync

GetHtmlFromElementByClassAsync

**Описание:** метод возвращает html представление объекта в строчном выражении

**Синтаксис:** GetHtmlFromElementByClassAsync(string \_class, int index)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string html = await tester.GetHtmlFromElementByClassAsync("text-field", 0);
```

---

Created with the Personal Edition of HelpNDoc: [Produce online help for Qt applications](#)

---

## GetHtmlFromElementByIdAsync

### GetHtmlFromElementByIdAsync

**Описание:** метод возвращает html представление объекта в строчном выражении

**Синтаксис:** GetHtmlFromElementByIdAsync(string id)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string html = await tester.GetHtmlFromElementByIdAsync("login");
```

---

Created with the Personal Edition of HelpNDoc: [Free iPhone documentation generator](#)

---

GetHtmlFromElementByNameAsync

GetHtmlFromElementByNameAsync

**Описание:** метод возвращает html представление объекта в строчном выражении

**Синтаксис:** GetHtmlFromElementByNameAsync(string name, int index)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string html = await tester.GetHtmlFromElementByNameAsync("field", 0);
```

## GetHtmlFromElementByTagAsync

### GetHtmlFromElementByTagAsync

**Описание:** метод возвращает html представление объекта в строчном выражении

**Синтаксис:** GetHtmlFromElementByTagAsync(string tag, int index)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string html = await tester.GetHtmlFromElementByTagAsync("h1", 0);
```

## IsVisibleElementAsync

### IsVisibleElementAsync

**Описание:** метод определяет видимость элемента и возвращает значение true или false

**Синтаксис:**

```
IsVisibleElementAsync(string by, string locator)
```

```
IsVisibleElementAsync(Locator locator)
```

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
bool result = await tester.IsVisibleElementAsync(Tester.BY_XPATH, "//*[@id='login']");
```

```
bool result = await tester.IsVisibleElementAsync(Tester.BY_CSS, "#login");
```

```
await tester.AssertTrueAsync(result);
```

```
bool result = await tester.IsVisibleElementAsync(tester.GetLocator("locatorName"));
```

```
await tester.AssertTrueAsync(result);
```

---

Created with the Personal Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

---

MakeElementVisibleAsync

MakeElementVisibleAsync

**Описание:** метод делает элемент видимым путем добавления стилей

**Синтаксис:**

```
MakeElementVisibleAsync(string by, string locator, string visibility = "visible", int opacity = 1, int index = 1000)
```

```
MakeElementVisibleAsync(Locator locator, string visibility = "visible", int opacity = 1, int index = 1000)
```

**Пример:**

```
await tester.MakeElementVisibleAsync(Tester.BY_XPATH, "//*[@id='button']");  
await tester.MakeElementVisibleAsync(Tester.BY_BY_CSS, "//*[@id='button']");
```

```
await tester.MakeElementVisibleAsync(tester.GetLocator("locatorName"));
```

---

Created with the Personal Edition of HelpNDoc: [Qt Help documentation made easy](#)

---

## SetHtmlInElementAsync

### SetHtmlInElementAsync

**Описание:** метод вставляет html представление объекта в указанный элемент

**Синтаксис:**

```
SetHtmlInElementAsync(string by, string locator, string html)
```

```
SetHtmlInElementAsync(Locator locator, string html)
```

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetHtmlInElementAsync(Tester.BY_CSS, "#auth > h2", "<div>Тестовый  
блок</div>");
```

```
await tester.SetHtmlInElementAsync(Tester.BY_XPATH, "//div[@id='auth']/h2", "<div>Тестовый  
блок</div>");
```

```
await tester.SetHtmlInElementAsync(tester.GetLocator("locatorName"), "<div>Тестовый  
блок</div>");
```

## SetHtmlInElementByClassAsync

### SetHtmlInElementByClassAsync

**Описание:** метод вставляет html представление объекта в указанный элемент

**Синтаксис:** SetHtmlInElementByClassAsync(string \_class, int index, string html)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetHtmlInElementByClassAsync("text-field", 0, "<h1>это тест</h1>");
```

---

Created with the Personal Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

---

## SetHtmlInElementByIdAsync

### SetHtmlInElementByIdAsync

**Описание:** метод вставляет html представление объекта в указанный элемент

**Синтаксис:** SetHtmlInElementByIdAsync(string id, string html)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetHtmlInElementByIdAsync("auth", "<h1>это тест</h1>");
```

## SetHtmlInElementByNameAsync

### SetHtmlInElementByNameAsync

**Описание:** метод вставляет html представление объекта в указанный элемент

**Синтаксис:** SetHtmlInElementByNameAsync(string name, int index, string html)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetHtmlInElementByNameAsync("block", 0, "<h1>это тест</h1>");
```

## SetHtmlInElementByTagAsync

### SetHtmlInElementByTagAsync

**Описание:** метод вставляет html представление объекта в указанный элемент

**Синтаксис:** SetHtmlInElementByTagAsync(string tag, int index, string html)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetHtmlInElementByTagAsync("div", 0, "<h1>это тест</h1>");
```

---

Created with the Personal Edition of HelpNDoc: [Write EPub books for the iPad](#)

---

## Ожидание

---

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

---

## WaitAsync

### WaitAsync

**Описание:** метод выполняет временную остановку выполнения теста на указанное количество секунд

**Синтаксис:** WaitAsync(int sec)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.WaitAsync(5);
```

## WaitElementInDomAsync

### WaitElementInDomAsync

**Описание:** метод выполняет временную остановку выполнения теста на указанное количество секунд и ждет когда запрашиваемый элемент появится в DOM

**Синтаксис:**

```
WaitElementInDomAsync(string by, string locator, int sec)
```

```
WaitElementInDomAsync(Locator locator, int sec)
```

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.WaitElementInDomAsync(Tester.BY_XPATH, "//div[@id='result']", 5);
```

```
await tester.WaitElementInDomAsync(Tester.BY_CSS, "#result", 5);
```

```
await tester.WaitElementInDomAsync(tester.GetLocator("locatorName"), 5);
```

## WaitElementNotDomAsync

### WaitElementNotDomAsync

**Описание:** метод выполняет временную остановку выполнения теста на указанное количество секунд и ждет когда запрашиваемый элемент перестанет присутствовать в DOM

**Синтаксис:**

```
WaitElementNotDomAsync(string by, string locator, int sec)
```

```
WaitElementNotDomAsync(Locator locator, int sec)
```

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.WaitElementNotDomAsync(Tester.BY_XPATH, "//div[@id='element']", 5);
```

```
await tester.WaitElementNotDomAsync(Tester.BY_CSS, "#element", 5);
```

```
await tester.WaitElementNotDomAsync(tester.GetLocator("locatorName"), 5);
```

## WaitNotVisibleElementAsync

### WaitNotVisibleElementAsync

**Описание:** метод выполняет временную остановку выполнения теста на указанное количество секунд и ждет когда запрашиваемый элемент перестанет отображаться

**Синтаксис:**

```
WaitNotVisibleElementAsync(string by, string locator, int sec)
```

```
WaitNotVisibleElementAsync(Locator locator, int sec)
```

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.WaitNotVisibleElementAsync(Tester.BY_CSS, "div[id='result']", 2);
```

```
await tester.WaitNotVisibleElementAsync(Tester.BY_XPATH, "//div[@id='result']", 2);
```

```
await tester.WaitNotVisibleElementAsync(tester.GetLocator("locatorName"), 2);
```

## WaitNotVisibleElementByClassAsync

### WaitNotVisibleElementByClassAsync

**Описание:** метод выполняет временную остановку выполнения теста на указанное количество секунд и ждет когда запрашиваемый элемент перестанет отображаться

**Синтаксис:** `WaitNotVisibleElementByClassAsync(string _class, int index, int sec)`

**Возвращаемое значение:** отсутствует

#### Пример:

```
await tester.WaitNotVisibleElementByClassAsync("my-element", 0, 25);
```

## WaitNotVisibleElementByIdAsync

### WaitNotVisibleElementByIdAsync

**Описание:** метод выполняет временную остановку выполнения теста на указанное количество секунд и ждет когда запрашиваемый элемент перестанет отображаться

**Синтаксис:** `WaitNotVisibleElementByIdAsync(string id, int sec)`

**Возвращаемое значение:** отсутствует

#### Пример:

```
await tester.WaitNotVisibleElementByIdAsync("MyElement", 25);
```

## WaitNotVisibleElementByNameAsync

### WaitNotVisibleElementByNameAsync

**Описание:** метод выполняет временную остановку выполнения теста на указанное количество секунд и ждет когда запрашиваемый элемент перестанет отображаться

**Синтаксис:** `WaitNotVisibleElementByNameAsync(string name, int index, int sec)`

**Возвращаемое значение:** отсутствует

#### Пример:

```
await tester.WaitNotVisibleElementByNameAsync("MyElement", 0, 25);
```

---

Created with the Personal Edition of HelpNDoc: [News and information about help authoring tools and software](#)

---

## WaitNotVisibleElementByTagAsync

### WaitNotVisibleElementByTagAsync

**Описание:** метод выполняет временную остановку выполнения теста на указанное количество секунд и ждет когда запрашиваемый элемент перестанет отображаться

**Синтаксис:** `WaitNotVisibleElementByTagAsync(string tag, int index, int sec)`

**Возвращаемое значение:** отсутствует

#### Пример:

```
await tester.WaitNotVisibleElementByTagAsync("h1", 0, 25);
```

## WaitVisibleElementAsync

### WaitVisibleElementAsync

**Описание:** метод выполняет временную остановку выполнения теста на указанное количество секунд и ждет когда запрашиваемый элемент отобразится

**Синтаксис:**

```
WaitVisibleElementAsync(string by, string locator, int sec)
```

```
WaitVisibleElementAsync(Locator locator, int sec)
```

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.WaitVisibleElementAsync(Tester.BY_CSS, "div[id='result']", 2);
```

```
await tester.WaitVisibleElementAsync(Tester.BY_XPATH, "//div[@id='result']", 2);
```

```
await tester.WaitVisibleElementAsync(tester.GetLocator("locatorName"), 2);
```

## WaitVisibleElementByClassAsync

### WaitVisibleElementByClassAsync

**Описание:** метод выполняет временную остановку выполнения теста на указанное количество секунд и ждет когда запрашиваемый элемент отобразится

**Синтаксис:** WaitVisibleElementByClassAsync(string \_class, int index, int sec)

**Возвращаемое значение:** отсутствует

#### Пример:

```
await tester.WaitVisibleElementByClassAsync("my-element", 0, 25);
```

---

Created with the Personal Edition of HelpNDoc: [Free iPhone documentation generator](#)

---

WaitVisibleElementByIdAsync

WaitVisibleElementByIdAsync

**Описание:** метод выполняет временную остановку выполнения теста на указанное количество секунд и ждет когда запрашиваемый элемент отобразится

**Синтаксис:** WaitVisibleElementByIdAsync(string id, int sec)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.WaitVisibleElementByIdAsync("MyElement", 25);
```

WaitVisibleElementByNameAsync

WaitVisibleElementByNameAsync

**Описание:** метод выполняет временную остановку выполнения теста на указанное количество секунд и ждет когда запрашиваемый элемент отобразится

**Синтаксис:** WaitVisibleElementByNameAsync(string name, int index, int sec)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.WaitVisibleElementByNameAsync("MyElement", 25);
```

---

Created with the Personal Edition of HelpNDoc: [Produce online help for Qt applications](#)

---

WaitVisibleElementByTagAsync

WaitVisibleElementByTagAsync

**Описание:** метод выполняет временную остановку выполнения теста на указанное количество секунд и ждет когда запрашиваемый элемент отобразится

**Синтаксис:** WaitVisibleElementByTagAsync(string tag, int index, int sec)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.WaitVisibleElementByTagAsync("h1", 25);
```

---

Created with the Personal Edition of HelpNDoc: [Easy EPub and documentation editor](#)

---

Поиск

---

Created with the Personal Edition of HelpNDoc: [Free EBook and documentation generator](#)

---

## FindElementAsync

### FindElementAsync

**Описание:** метод выполняет поиск элемент в DOM с ожиданием в секундах и возвращает логический результат true или false

**Синтаксис:**

FindElementAsync(string by, string locator, int sec)

FindElementAsync(Locator locator, int sec)

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
bool result = await tester.FindElementAsync(Tester.BY_CSS, "div[id='result']", 2);
```

```
bool result = await tester.FindElementAsync(Tester.BY_XPATH, "//div[@id='result']", 2);
```

```
bool result = await tester.FindElementAsync(tester.GetLocator("locatorName"), 2);
```

## FindElementByClassAsync

### FindElementByClassAsync

**Описание:** метод выполняет поиск элемент в DOM с ожиданием в секундах и возвращает логический результат true или false

**Синтаксис:** FindElementByClassAsync(string \_class, int index, int sec)

**Возвращаемое значение:** логическое (true или false)

#### Пример:

```
bool result = await tester.FindElementByClassAsync("my-element", 0, 5);
```

---

Created with the Personal Edition of HelpNDoc: [Produce online help for Qt applications](#)

---

## FindElementByIdAsync

### FindElementByIdAsync

**Описание:** метод выполняет поиск элемент в DOM с ожиданием в секундах и возвращает логический результат true или false

**Синтаксис:** FindElementByIdAsync(string id, int sec)

**Возвращаемое значение:** логическое (true или false)

#### Пример:

```
bool result = await tester.FindElementByIdAsync("MyElement", 5);
```

## FindElementByNameAsync

### FindElementByNameAsync

**Описание:** метод выполняет поиск элемент в DOM с ожиданием в секундах и возвращает логический результат true или false

**Синтаксис:** FindElementByNameAsync(string name, int index, int sec)

**Возвращаемое значение:** логическое (true или false)

#### Пример:

```
bool result = await tester.FindElementByNameAsync("MyElement", 0, 5);
```

---

Created with the Personal Edition of HelpNDoc: [Free Kindle producer](#)

---

## FindElementByTagAsync

### FindElementByTagAsync

**Описание:** метод выполняет поиск элемент в DOM с ожиданием в секундах и возвращает логический результат true или false

**Синтаксис:** FindElementByTagAsync(string tag, int index, int sec)

**Возвращаемое значение:** логическое (true или false)

#### Пример:

```
bool result = await tester.FindElementByTagAsync("h1", 0, 5);
```

## FindVisibleElementAsync

### FindVisibleElementAsync

**Описание:** метод выполняет поиск визуально отображаемого элемента с ожиданием в секундах и возвращает логический результат true или false

**Синтаксис:**

```
FindVisibleElementAsync(string by, string locator, int sec)
```

```
FindVisibleElementAsync(Locator locator, int sec)
```

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
bool result = await tester.FindVisibleElementAsync(Tester.BY_CSS, "#auth #buttonLogin", 2);
```

```
bool result = await tester.FindVisibleElementAsync(Tester.BY_XPATH,
```

```
"//div[@id='auth']//input[@id='buttonLogin']", 2);
```

```
bool result = await tester.FindVisibleElementAsync(tester.GetLocator("locatorName"), 2);
```

---

Created with the Personal Edition of HelpNDoc: [Free iPhone documentation generator](#)

---

## FindVisibleElementByClassAsync

### FindVisibleElementByClassAsync

**Описание:** метод выполняет поиск визуально отображаемого элемента с ожиданием в секундах и возвращает логический результат true или false

**Синтаксис:** FindVisibleElementByClassAsync(string \_class, int index, int sec)

**Возвращаемое значение:** логическое (true или false)

#### Пример:

```
bool result = await tester.FindVisibleElementByClassAsync("my-element", 0, 5);
```

---

Created with the Personal Edition of HelpNDoc: [News and information about help authoring tools and software](#)

---

FindVisibleElementByIdAsync

FindVisibleElementByIdAsync

**Описание:** метод выполняет поиск визуально отображаемого элемента с ожиданием в секундах и возвращает логический результат true или false

**Синтаксис:** FindVisibleElementByIdAsync(string id, int sec)

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
bool result = await tester.FindVisibleElementByIdAsync("MyElement", 5);
```

---

Created with the Personal Edition of HelpNDoc: [Free EBook and documentation generator](#)

---

FindVisibleElementByNameAsync

FindVisibleElementByNameAsync

**Описание:** метод выполняет поиск визуально отображаемого элемента с ожиданием в секундах и возвращает логический результат true или false

**Синтаксис:** FindVisibleElementByNameAsync(string name, int index, int sec)

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
bool result = await tester.FindVisibleElementByNameAsync("MyElement", 0, 5);
```

---

Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

---

## FindVisibleElementByTagAsync

### FindVisibleElementByTagAsync

**Описание:** метод выполняет поиск визуально отображаемого элемента с ожиданием в секундах и возвращает логический результат true или false

**Синтаксис:** FindVisibleElementByTagAsync(string tag, int index, int sec)

**Возвращаемое значение:** логическое (true или false)

#### Пример:

```
bool result = await tester.FindVisibleElementByTagAsync("h1", 0, 5);
```

---

Created with the Personal Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

---

## Стили

---

Created with the Personal Edition of HelpNDoc: [News and information about help authoring tools and software](#)

---

GetStyleFromElementAsync

GetStyleFromElementAsync

**Описание:** метод возвращает стиль элемента из указанного свойства

**Синтаксис:**

GetStyleFromElementAsync(string by, string locator, string property)

GetStyleFromElementAsync(Locator locator, string property)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string style = await tester.GetStyleFromElementAsync(Tester.BY_CSS, "#auth", "padding");
```

```
string style = await tester.GetStyleFromElementAsync(Tester.BY_XPATH, "//div[@id='auth']",  
"position");
```

```
string style = await tester.GetStyleFromElementAsync(tester.GetLocator("locatorName"),  
"padding");
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

---

## GetStyleFromElementByClassAsync

### GetStyleFromElementByClassAsync

**Описание:** метод возвращает стиль элемента из указанного свойства

**Синтаксис:** GetStyleFromElementByClassAsync(string \_class, int index, string property)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string style = await tester.GetStyleFromElementByClassAsync("text-field", 0, "border");
```

---

Created with the Personal Edition of HelpNDoc: [Free help authoring environment](#)

---

## GetStyleFromElementByIdAsync

### GetStyleFromElementByIdAsync

**Описание:** метод возвращает стиль элемента из указанного свойства

**Синтаксис:** GetStyleFromElementByIdAsync(string id, string property)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string style = await tester.GetStyleFromElementByIdAsync("buttonLogin", "background-color");
```

GetStyleFromElementByNameAsync

GetStyleFromElementByNameAsync

**Описание:** метод возвращает стиль элемента из указанного свойства

**Синтаксис:** GetStyleFromElementByNameAsync(string name, int index, string property)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string style = await tester.GetStyleFromElementByNameAsync("pass", 0, "height");
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

---

## GetStyleFromElementByTagAsync

### GetStyleFromElementByTagAsync

**Описание:** метод возвращает стиль элемента из указанного свойства

**Синтаксис:** GetStyleFromElementByTagAsync(string tag, int index, string property)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string style = await tester.GetStyleFromElementByTagAsync("h2", 0, "width");
```

## SetStyleInElementAsync

### SetStyleInElementAsync

**Описание:** метод устанавливает стиль элемента

**Синтаксис:**

```
SetStyleInElementAsync(string by, string locator, string cssText)
```

```
SetStyleInElementAsync(Locator locator, string cssText)
```

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetStyleInElementAsync(Tester.BY_XPATH, "//div[@id='auth']", "width: 250px; color: white; background-color: #000000;");
```

```
await tester.SetStyleInElementAsync(Tester.BY_CSS, "#auth", "width: 250px; color: white; background-color: #000000;");
```

```
await tester.SetStyleInElementAsync(tester.GetLocator("locatorName"), "width: 250px; color: white; background-color: #000000;");
```

## SetStyleInElementByClassAsync

### SetStyleInElementByClassAsync

**Описание:** метод устанавливает стиль элемента

**Синтаксис:** SetStyleInElementByClassAsync(string \_class, int index, string cssText)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetStyleInElementByClassAsync("text-field", 0, "background-color: #123456;");
```

## SetStyleInElementByIdAsync

### SetStyleInElementByIdAsync

**Описание:** метод устанавливает стиль элемента

**Синтаксис:** SetStyleInElementByIdAsync(string id, string cssText)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetStyleInElementByIdAsync("buttonLogin", "background-color: #123456;");
```

## SetStyleInElementByNameAsync

### SetStyleInElementByNameAsync

**Описание:** метод устанавливает стиль элемента

**Синтаксис:** SetStyleInElementByNameAsync(string name, int index, string cssText)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetStyleInElementByNameAsync("pass", 0, "background-color: #123456;");
```

---

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

---

## SetStyleInElementByTagAsync

### SetStyleInElementByTagAsync

**Описание:** метод устанавливает стиль элемента

**Синтаксис:** SetStyleInElementByTagAsync(string tag, int index, string cssText)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetStyleInElementByTagAsync("h2", 0, "background-color: #123456;");
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

---

Страница

---

Created with the Personal Edition of HelpNDoc: [Free Kindle producer](#)

---

## GetCookiesAsync

### GetCookiesAsync

**Описание:** метод возвращает список куки (document.cookie).

**Синтаксис:** GetCookiesAsync()

#### Пример:

```
List<string> cookies = await tester.GetCookiesAsync();
if (cookies.Count() > 0)
{
    foreach (string cookie in cookies)
    {
        tester.ConsoleMsg("COOKIE: " + cookie);
    }
}
```

## GetListRedirectUrlAsync

### GetListRedirectUrlAsync

**Описание:** метод возвращает список редиректов произошедших при загрузке страницы

**Синтаксис:** GetListRedirectUrlAsync()

**Возвращаемое значение:** список (List)

**Пример:**

```
Tester tester = new Tester(browserForm);  
await tester.TestBeginAsync();  
await tester.GoToUrlAsync("https://yandex.ru/", 5);
```

```
List<string> redirects = await tester.GetListRedirectUrlAsync();  
foreach (string url in redirects)  
{  
    tester.ConsoleMsg(url);  
}
```

GetTitleAsync

GetTitleAsync

**Описание:** метод возвращает заголовок страницы

**Синтаксис:** GetTitleAsync()

**Возвращаемое значение:** строка (string)

**Пример:**

```
string text = await tester.GetTitleAsync();
```

## GetUrlAsync

### GetUrlAsync

**Описание:** метод возвращает текущий URL

**Синтаксис:** GetUrlAsync()

**Возвращаемое значение:** строка (string)

**Пример:**

```
string text = await tester.GetUrlAsync();
```

## GetUrlResponseAsync

### GetUrlResponseAsync

**Описание:** метод возвращает HTTP ответ указанного URL

**Синтаксис:** GetUrlResponseAsync(string url)

**Возвращаемое значение:** целое число (int)

**Пример:**

```
int response = await tester.GetUrlResponseAsync("https://somovstudio.github.io/test.html");  
await tester.AssertEqualsAsync(200, response);
```

## GoToUrlAsync

### GoToUrlAsync

**Описание:** метод выполняет загрузку веб сайта по указанному URL с указанным ожиданием в секундах

Флаг abortLoadAfterTime принудительно завершает загрузку страницы после завершения таймера

**Синтаксис:** GoToUrlAsync(string url, int sec, bool abortLoadAfterTime = false)

**Возвращаемое значение:** отсутствует

### Пример:

```
await tester.GoToUrlAsync(@"https://www.google.com/", 25);
```

## GoToUrlBaseAuthAsync

### GoToUrlBaseAuthAsync

**Описание:** метод выполняет загрузку веб сайта по указанному URL при базовой авторизации с указанным ожиданием в секундах  
Флаг abortLoadAfterTime принудительно завершает загрузку страницы после завершения таймера

**Синтаксис:** GoToUrlBaseAuthAsync(string url, string login, string pass, int sec, bool abortLoadAfterTime = false)

**Возвращаемое значение:** отсутствует

#### Пример:

```
await tester.GoToUrlBaseAuthAsync("https://dev.site.com", "login", "pass", 25);
```

## LoadPageAsync

### LoadPageAsync

**Описание:** метод выполняет загрузку веб-страницы по указанному URL с указанным ожиданием в секундах. Флаг `abortLoadAfterTime` принудительно завершает загрузку страницы после завершения таймера. Ошибки при загрузке страницы игнорируются и не приводят к падению автотеста.

**Синтаксис:** `LoadPageAsync(string url, int sec, bool abortLoadAfterTime = false)`

### Пример:

```
await tester.LoadPageAsync (@"https://www.google.com/", 25, true);
```

---

Created with the Personal Edition of HelpNDoc: [Create iPhone web-based documentation](#)

---

## Текст

---

Created with the Personal Edition of HelpNDoc: [Free EPub producer](#)

---

GetTextFromElementAsync

GetTextFromElementAsync

**Описание:** метод возвращает текст из указанного элемента

**Синтаксис:**

GetTextFromElementAsync(string by, string locator)

GetTextFromElementAsync(Locator locator)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string text = await tester.GetTextFromElementAsync(Tester.BY_CSS, "#auth > h2");
```

```
string text = await tester.GetTextFromElementAsync(Tester.BY_XPATH, "//div[@id='auth']/h2");
```

```
string text = await tester.GetTextFromElementAsync(tester.GetLocator("locatorName"));
```

GetTextFromElementByClassAsync

GetTextFromElementByClassAsync

**Описание:** метод возвращает текст из указанного элемента

**Синтаксис:** GetTextFromElementByClassAsync(string \_class, int index)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string text = await tester.GetTextFromElementByClassAsync("my-element", 0);
```

---

Created with the Personal Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

---

## GetTextFromElementByIdAsync

### GetTextFromElementByIdAsync

**Описание:** метод возвращает текст из указанного элемента

**Синтаксис:** GetTextFromElementByIdAsync(string id)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string text = await tester.GetTextFromElementByIdAsync("MyElement");
```

GetTextFromElementByNameAsync

GetTextFromElementByNameAsync

**Описание:** метод возвращает текст из указанного элемента

**Синтаксис:** GetTextFromElementByNameAsync(string name, int index)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string text = await tester.GetTextFromElementByNameAsync("MyElement", 0);
```

---

Created with the Personal Edition of HelpNDoc: [Produce online help for Qt applications](#)

---

## GetTextFromElementByTagAsync

### GetTextFromElementByTagAsync

**Описание:** метод возвращает текст из указанного элемента

**Синтаксис:** GetTextFromElementByTagAsync(string tag, int index)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string text = await tester.GetTextFromElementByTagAsync("h1", 0);
```

## SetTextInElementAsync

### SetTextInElementAsync

**Описание:** метод вставляет текст в указанный элемент

**Синтаксис:**

```
SetTextInElementAsync(string by, string locator, string text)
```

```
SetTextInElementAsync(Locator locator, string text)
```

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetTextInElementAsync(Tester.BY_CSS, "#auth > h2", "Тестовый заголовок");  
await tester.SetTextInElementAsync(Tester.BY_XPATH, "//div[@id='auth']/h2", "Тестовый  
заголовок");
```

```
await tester.SetTextInElementAsync(tester.GetLocator("locatorName"), "Тестовый заголовок");
```

---

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

---

## SetTextInElementByClassAsync

### SetTextInElementByClassAsync

**Описание:** метод вставляет текст в указанный элемент

**Синтаксис:** `SetTextInElementByClassAsync(string _class, int index, string text)`

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetTextInElementByClassAsync("my-element", 0, "текст");
```

---

Created with the Personal Edition of HelpNDoc: [Produce online help for Qt applications](#)

---

## SetTextInElementByIdAsync

### SetTextInElementByIdAsync

**Описание:** метод вставляет текст в указанный элемент

**Синтаксис:** `SetTextInElementByIdAsync(string id, string text)`

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetTextInElementByIdAsync("MyElement", "текст");
```

## SetTextInElementByNameAsync

### SetTextInElementByNameAsync

**Описание:** метод вставляет текст в указанный элемент

**Синтаксис:** `SetTextInElementByNameAsync(string name, int index, string text)`

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetTextInElementByNameAsync("MyElement", 0, "текст");
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

---

## SetTextInElementByTagAsync

### SetTextInElementByTagAsync

**Описание:** метод вставляет текст в указанный элемент

**Синтаксис:** SetTextInElementByTagAsync(string tag, int index, string text)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.SetTextInElementByTagAsync("h1", 0, "текст");
```

---

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

---

## Методы для выполнения JavaScript

---

Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

---

## ExecuteJavaScriptAsync

### ExecuteJavaScriptAsync

**Описание:** метод выполняет JavaScript код и возвращает строку с результатом выполнения

**Синтаксис:** ExecuteJavaScriptAsync(string script)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string script = "(function(){ var element = document.getElementById('MyElement'); return element.innerText; });";
string result = await tester.ExecuteJavaScriptAsync(script);
```

---

Created with the Personal Edition of HelpNDoc: [Easily create Help documents](#)

---

## Методы для выполнения Rest запросов

---

Created with the Personal Edition of HelpNDoc: [Free CHM Help documentation generator](#)

---

## RestGetAsync

### RestGetAsync

**Описание:** метод выполняет Get Rest запрос и получает результат в формате json

**Синтаксис:** RestGetAsync(string url, TimeSpan timeout, string charset = "UTF-8")

**Возвращаемое значение:** строка (string)

**Пример:**

```
string result = await tester.RestGetAsync("https://jsonplaceholder.typicode.com/posts/1/",
TimeSpan.FromDays(1), "UTF-8");
tester.ConsoleMsg(result);
```

**Данный метод использует стандартный подход:**

```

using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;

Uri uri = new Uri(url);
HttpClient client = new HttpClient();
client.Timeout = TimeSpan.FromDays(1);
client.BaseAddress = uri;
client.DefaultRequestHeaders.Clear();
client.DefaultRequestHeaders.Accept.Clear();
client.DefaultRequestHeaders.Accept.Add(new
MediaTypeWithQualityHeaderValue("application/json"));
client.DefaultRequestHeaders.Add("charset", "UTF-8");
client.DefaultRequestHeaders.Add("User-Agent", userAgent);
HttpResponseMessage response = await client.GetAsync(url);
if (response.IsSuccessStatusCode)
{
    return await response.Content.ReadAsStringAsync();
}

```

---

Created with the Personal Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

---

## RestGetBasicAuthAsync

### RestGetBasicAuthAsync

**Описание:** метод выполняет Get Rest запрос и получает результат в формате json

**Синтаксис:** RestGetBasicAuthAsync(string login, string pass, string url, string charset = "UTF-8")

**Возвращаемое значение:** строка (string)

#### Пример:

```

string result = await tester.RestGetBasicAuthAsync("admin", "0000",
"https://jsonplaceholder.typicode.com/posts/1/", TimeSpan timeout, "UTF-8");
tester.ConsoleMsg(result);

```

**Данный метод использует стандартный подход:**

```

using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;

byte[] authToken = Encoding.ASCII.GetBytes($"{{login}}:{{pass}}");
Uri uri = new Uri(url);
HttpClient client = new HttpClient();
client.Timeout = TimeSpan.FromDays(1);
client.BaseAddress = uri;

```

```

client.DefaultRequestHeaders.Clear();
client.DefaultRequestHeaders.Accept.Clear();
client.DefaultRequestHeaders.Accept.Add(new
MediaTypeWithQualityHeaderValue("application/json"));
client.DefaultRequestHeaders.Add("charset", "UTF-8");
client.DefaultRequestHeaders.Add("User-Agent", userAgent);
client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Basic",
Convert.ToBase64String(authToken));
HttpResponseMessage response = await client.GetAsync(url);
if (response.IsSuccessStatusCode)
{
    return await response.Content.ReadAsStringAsync();
}

```

---

Created with the Personal Edition of HelpNDoc: [Free iPhone documentation generator](#)

---

## RestGetStatusCodeAsync

### RestGetStatusCodeAsync

**Описание:** метод выполняет Get Rest запрос и в результате получает код статуса

**Синтаксис:**

```

RestGetStatusCodeAsync(string url)
RestGetStatusCodeAsync(string login, string pass, string url)

```

**Возвращаемое значение:** целое число (int)

**Пример:**

```

int statusCode = await tester.RestGetAsync("https://jsonplaceholder.typicode.com");
tester.ConsoleMsg(statusCode.ToString());

```

---

Created with the Personal Edition of HelpNDoc: [Full-featured EBook editor](#)

---

## RestPostAsync

### RestPostAsync

**Описание:** метод выполняет Post Rest запрос с отправкой данных в формате json и получает результат так же в формате json

**Синтаксис:** RestPostAsync(string url, string json, TimeSpan timeout, string charset = "UTF-8")

**Возвращаемое значение:** строка (string)

**Пример:**

```

string result = await tester.RestPostAsync("https://jsonplaceholder.typicode.com/posts/1/",
"{}", TimeSpan.FromDays(1), "UTF-8");
tester.ConsoleMsg(result);

```

**Данный метод использует стандартный подход:**

```

using System.Net;
using System.Net.Http;

```

```

using System.Net.Http.Headers;

Uri uri = new Uri(url);
HttpClient client = new HttpClient();
client.Timeout = TimeSpan.FromDays(1);
client.DefaultRequestHeaders.Add("charset", "UTF-8");
client.DefaultRequestHeaders.Add("User-Agent", userAgent);
HttpContent content = new StringContent("{} ", Encoding.UTF8, "application/json");
HttpResponseMessage response = await client.PostAsync(uri, content);
if (response.IsSuccessStatusCode)
{
    return await response.Content.ReadAsStringAsync();
}

```

---

Created with the Personal Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

---

## Методы для замера затраченного времени

---

Created with the Personal Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

---

### TimerStart

#### TimerStart

**Описание:** метод запускает отсчет времени и возвращает значение DateTime

**Синтаксис:** TimerStart()

**Возвращаемое значение:** дата и время (DateTime)

#### Пример:

```

Tester tester = new Tester(browserForm);
DateTime start = await tester.TimerStart();
await tester.TestBeginAsync();
...
await tester.TestEndAsync();
TimeSpan result = await tester.TimerStop(start);
tester.ConsoleMsg("Time " + result.TotalSeconds);

```

---

Created with the Personal Edition of HelpNDoc: [Easy EBook and documentation generator](#)

---

### TimerStop

#### TimerStop

**Описание:** метод завершает отсчет времени и возвращает значение TimeSpan с результатом (например 7,132157 что значит 7 секунд и 132157 миллисекунд)

**Синтаксис:** TimerStop(DateTime start)

**Возвращаемое значение:** время (TimeSpan)

#### Пример:

```

Tester tester = new Tester(browserForm);
DateTime start = await tester.TimerStart();
await tester.TestBeginAsync();
...
await tester.TestEndAsync();
TimeSpan result = await tester.TimerStop(start);
tester.ConsoleMsg("Time " + result.TotalSeconds);

```

---

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

---

## Методы для отправки email и message

---

Created with the Personal Edition of HelpNDoc: [Free EBook and documentation generator](#)

---

### SendMsgToMailAsync

#### SendMsgToMailAsync

**Описание:** метод отправляет письмо на почту

**Синтаксис:** SendMsgToMailAsync(string subject, string body, string filename = "")

**Возвращаемое значение:** отсутствует

#### Пример:

```

Tester tester = new Tester(browserForm);
await tester.TestBeginAsync();
...
await tester.TestEndAsync();
if(tester.GetTestResult() == Tester.PASSED)
{
    await tester.SendMsgToMailAsync("Тестовое сообщение", "Тест завершился успешно");
}

```

---

Created with the Personal Edition of HelpNDoc: [Free EPub and documentation generator](#)

---

### SendMsgToTelegramAsync

#### SendMsgToTelegramAsync

**Описание:** метод отправляет сообщение в Телеграм, для этого нужно создать бота добавить его в чат и в функцию передать токен бота (botToken), идентификатор чата (chatId), текст сообщения (text) и при необходимости указать кодировку (charset)

**Синтаксис:** SendMsgToTelegramAsync(string botToken, string chatId, string text, string charset = "UTF-8", int timeHourFrom = 0, int timeHourBefore = 0)

**Возвращаемое значение:** отсутствует

#### Пример:

```

Tester tester = new Tester(browserForm);
await tester.TestBeginAsync();

```

```

...
await tester.TestEndAsync();
if(tester.GetTestResult() == Tester.FAILED)
{
    // сообщение будет отправлено только если текущее время отправки попадает в
    // диапазон от 9:00 до 21:00
    // если указать 0, 0, в таком случае сообщение будет отправлено без учета
    // текущего времени
    await
tester.SendMsgToTelegramAsync("0000000001:ABCDabcd123ABCDabcd123ABCDabcd123ZX",
"-123456789", "Тестовое сообщение", "UTF-8", 9, 21);
}

```

---

Created with the Personal Edition of HelpNDoc: [Qt Help documentation made easy](#)

---

## Методы для проверки результата

---

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

---

### AssertEqualsAsync

#### AssertEqualsAsync

**Описание:** метод выполняет проверку между фактическим и ожидаемым значениями, в случае несовпадения проверка будет считаться провальной

**Синтаксис:** AssertEqualsAsync(dynamic expected, dynamic actual)

**Возвращаемое значение:** логическое (true или false)

#### Пример:

```

string expected = "xyz";
string actual = "xyz";
bool result = await tester.AssertEqualsAsync(expected, actual);

```

---

Created with the Personal Edition of HelpNDoc: [Full-featured Help generator](#)

---

### AssertNotEqualsAsync

#### AssertNotEqualsAsync

**Описание:** метод выполняет проверку между фактическим и ожидаемым значениями, в случае совпадения проверка будет считаться провальной

**Синтаксис:** AssertNotEqualsAsync(dynamic expected, dynamic actual)

**Возвращаемое значение:** логическое (true или false)

#### Пример:

```

string expected = "abc";
string actual = "xyz";
bool result = await tester.AssertNotEqualsAsync(expected, actual);

```

---

Created with the Personal Edition of HelpNDoc: [Free Kindle producer](#)

---

## AssertTrueAsync

### AssertTrueAsync

**Описание:** метод выполняет проверку значения true или false, и в случае если значение равно false проверка будет считаться провальной

**Синтаксис:** AssertTrueAsync(bool condition)

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
bool flag = true;  
bool result = await tester.AssertTrueAsync(flag);
```

---

Created with the Personal Edition of HelpNDoc: [News and information about help authoring tools and software](#)

---

## AssertFalseAsync

### AssertFalseAsync

**Описание:** метод выполняет проверку значения true или false, и в случае если значение равно true проверка будет считаться провальной

**Синтаксис:** AssertFalseAsync(bool condition)

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
bool flag = false;  
bool result = await tester.AssertFalseAsync(flag);
```

---

Created with the Personal Edition of HelpNDoc: [Easy EPub and documentation editor](#)

---

## AssertNotNullAsync

### AssertNotNullAsync

**Описание:** метод выполняет проверку значения которое не должно быть null, и в случае если значение null проверка будет считаться провальной

**Синтаксис:** AssertNotNullAsync(dynamic obj)

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
string text = "текст";  
await tester.AssertNotNullAsync(text);
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

---

## AssertNullAsync

### AssertNullAsync

**Описание:** метод выполняет проверку значения которое должно быть null, и в случае если значение не null проверка будет считаться провальной

**Синтаксис:** AssertNullAsync(dynamic obj)

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
string text = null;
await tester.AssertNullAsync(text);
```

---

Created with the Personal Edition of HelpNDoc: [Easy EPub and documentation editor](#)

---

## AssertNoErrorsAsync

### AssertNoErrorsAsync

**Описание:** метод выполняет проверку отсутствия ошибок на странице и если ошибки присутствуют проверка будет считаться провальной

**Синтаксис:** AssertNoErrorsAsync(bool showListErrors = false, string[] listIgnored = null)

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
Tester tester = new Tester(browserForm);
...
await tester.TestBeginAsync();
await tester.GoToUrlAsync("https://somovstudio.github.io/test_error.html", 25);
await tester.AssertNoErrorsAsync(true, new string[1] { "stats.g.doubleclick.net" });
await tester.TestEndAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured EBook editor](#)

---

## AssertNetworkEventsAsync

### AssertNetworkEventsAsync

**Описание:** метод выполняет проверку присутствия (presence = true) или отсутствие (presence = false) указанных событий (events), данный метод хорошо использовать при проверке событий Google Analytics и Яндекс Метрика

**Синтаксис:** AssertNetworkEventsAsync(bool presence, string[] events)

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
await tester.AssertNetworkEventsAsync(true, new string[] {
    "ec=zayavka", "ea=b2c_new_main", "el=some_shpd_nm", "zayavka_shpd"
});
```

---

Created with the Personal Edition of HelpNDoc: [Free EBook and documentation generator](#)

---

## Методы для работы с файлами

---

Created with the Personal Edition of HelpNDoc: [Free Web Help generator](#)

---

## FileDownloadAsync

### FileDownloadAsync

**Описание:** метод скачивает файла по указанному URL

**Синтаксис:** FileDownloadAsync(string fileURL, string filename, int waitingSec = 60)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await tester.FileDownloadAsync("https://somovstudio.github.io/img/logo.png", "C:\\download\\logo.png", 60);
```

---

Created with the Personal Edition of HelpNDoc: [Easy CHM and documentation editor](#)

---

FileGetHashMD5Async

[FileGetHashMD5Async](#)

**Описание:** метод определяет HashMD5 код указанного файла

**Синтаксис:** FileGetHashMD5Async(string filename)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string hash = await tester.FileGetHashMD5Async("C:\\download\\logo.png");
```

---

Created with the Personal Edition of HelpNDoc: [Free EBook and documentation generator](#)

---

FileReadAsync

[FileReadAsync](#)

**Описание:** метод выполняет чтение указанного файла

**Синтаксис:** FileReadAsync(string encoding, string filename)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string text = await tester.FileReadAsync(Tester.UTF8, "C:\\Hat\\file.txt");
```

---

Created with the Personal Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

---

FileWriteAsync

[FileWriteAsync](#)

**Описание:** метод выполняет запись текста в указанный файла

**Синтаксис:** FileWriteAsync(string content, string encoding, string filename)

**Возвращаемое значение:** отсутствует

**Пример:**

```
string text = "my text";  
await tester.FileWriteAsync(text, Tester.UTF8, "C:\\Hat\\file.txt");
```

---

Created with the Personal Edition of HelpNDoc: [Free PDF documentation generator](#)

---

## Методы для разных задач

---

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

---

### CreateHashMD5FromTextAsync

#### CreateHashMD5FromTextAsync

**Описание:** метод создает HashDM5 код из указанного текста

**Синтаксис:** CreateHashMD5FromTextAsync(string text)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string text = "Hello World";
string hash = await tester.CreateHashMD5FromTextAsync(text);
tester.ConsoleMsg(hash);
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured EPub generator](#)

---

### GetProjectPath

#### GetProjectPath

**Описание:** метод возвращает путь к папке проекта

**Синтаксис:** GetProjectPath()

**Возвращаемое значение:** строка (string)

**Пример:**

```
string path = tester.GetProjectPath();
tester.ConsoleMsg(path);
```

---

Created with the Personal Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

---

## Класс: HTMLElement

---

Created with the Personal Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

---

### Конструктор

---

Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

---

### HTMLElement

#### HTMLElement

**Описание:** вспомогательный класс, объект html элемента

**Синтаксис:** HTMLElement(Tester tester, string by, string locator)

**Пример получения объекта:**

```
HTMLElement element = await tester.GetElementAsync(Tester.BY_CSS, "#auth #buttonLogin");
await element.ClickAsync();
```

```
HTMLElement element = await tester.GetElementAsync(Tester.BY_XPATH,  
"//div[@id='auth']//input[@id='buttonLogin']");  
await element.ClickAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

---

## Константы

---

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

---

BY\_INDEX

[BY\\_INDEX](#)

**Описание:** константа обозначает тип обрабатываемого значения

**Синтаксис:** BY\_INDEX = "BY\_INDEX"

**Пример:**

HTMLElement.BY\_INDEX

---

Created with the Personal Edition of HelpNDoc: [News and information about help authoring tools and software](#)

---

BY\_TEXT

[BY\\_TEXT](#)

**Описание:** константа обозначает тип обрабатываемого значения

**Синтаксис:** BY\_TEXT = "BY\_TEXT"

**Пример:**

HTMLElement.BY\_TEXT

---

Created with the Personal Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

---

BY\_VALUE

[BY\\_VALUE](#)

**Описание:** константа обозначает тип обрабатываемого значения

**Синтаксис:** BY\_VALUE = "BY\_VALUE"

**Пример:**

HTMLElement.BY\_VALUE

---

Created with the Personal Edition of HelpNDoc: [Produce online help for Qt applications](#)

---

## Переменные

---

Created with the Personal Edition of HelpNDoc: [Generate Kindle eBooks with ease](#)

---

Id

Id

**Описание:** переменная возвращает или получает данные ID

**Синтаксис:** string Id { get; set; }

**Пример:**

```
HTMLInputElement element = await tester.GetElementAsync(Tester.BY_XPATH,
"//*[@id='MyInput']");
tester.ConsoleMsg("ID: " + element.Id);
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured Help generator](#)

---

Name

Name

**Описание:** переменная возвращает или получает данные Name

**Синтаксис:** string Name { get; set; }

**Пример:**

```
HTMLInputElement element = await tester.GetElementAsync(Tester.BY_XPATH,
"//*[@id='MyInput']");
tester.ConsoleMsg("NAME: " + element.Name);
```

---

Created with the Personal Edition of HelpNDoc: [Free Qt Help documentation generator](#)

---

Class

Class

**Описание:** переменная возвращает или получает данные Class

**Синтаксис:** string Class { get; set; }

**Пример:**

```
HTMLInputElement element = await tester.GetElementAsync(Tester.BY_XPATH,
"//*[@id='MyInput']");
tester.ConsoleMsg("CLASS: " + element.Class);
```

---

Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

---

Type

Type

**Описание:** переменная возвращает или получает данные Type

**Синтаксис:** string Type { get; set; }

**Пример:**

```
HTMLInputElement element = await tester.GetElementAsync(Tester.BY_XPATH,
"//*[@id='MyInput']");
tester.ConsoleMsg("TYPE: " + element.Type);
```

---

Created with the Personal Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

---

## Методы

---

Created with the Personal Edition of HelpNDoc: [Free CHM Help documentation generator](#)

---

### ClickAsync

#### ClickAsync

**Описание:** метод выполняет нажатие на элемент

**Синтаксис:** ClickAsync();

**Возвращаемое значение:** отсутствует

**Пример:**

```
await element.ClickAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Free HTML Help documentation generator](#)

---

### ClickMouseAsync

#### ClickMouseAsync

**Описание:** метод выполняет нажатие на элемент эмулируя мышку

**Синтаксис:** ClickMouseAsync()

**Возвращаемое значение:** отсутствует

**Пример:**

```
await element.ClickMouseAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Easy CHM and documentation editor](#)

---

### FocusAsync

#### FocusAsync

**Описание:** метод переключает фокус на элемент

**Синтаксис:** FocusAsync();

**Возвращаемое значение:** отсутствует

**Пример:**

```
await element.FocusAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

---

### GetAttributeAsync

#### GetAttributeAsync

**Описание:** метод возвращает значение указанного атрибута из элементе

**Синтаксис:** GetAttributeAsync(string name);

**Возвращаемое значение:** строка (string)

**Пример:**

```
string attrClass = await element.GetAttributeAsync("class");
```

## GetHtmlAsync

### GetHtmlAsync

**Описание:** метод возвращает html представление элемента в строчном выражении

**Синтаксис:** GetHtmlAsync();

**Возвращаемое значение:** строка (string)

**Пример:**

```
string html = await element.GetHtmlAsync();
```

## GetLocatorAsync

### GetLocatorAsync

**Описание:** метод возвращает локатор элемента

**Синтаксис:** GetLocatorAsync()

**Возвращаемое значение:** строка (string)

**Пример:**

```
string locator = await element.GetLocatorAsync();
tester.ConsoleMsg(locator);
```

## GetOptionAsync

### GetOptionAsync

**Описание:** метод возвращает индекс, текст или значение выбоанной опции выбранной option из элемента select

**Синтаксис:** GetOptionAsync(string by)

**Возвращаемое значение:** строка (string)

**Пример:**

```
Tester tester = new Tester(browserForm);
await tester.TestBeginAsync();
await tester.GoToUrlAsync("https://somovstudio.github.io/test2.html", 5);
HTMLElement element = await tester.GetElementAsync(Tester.BY_XPATH,
"//*[@id='MySelect']");
await element.SelectOptionAsync(HTMLElement.BY_INDEX, "2");
await element.SelectOptionAsync(HTMLElement.BY_VALUE, "Mobile");
await element.SelectOptionAsync(HTMLElement.BY_TEXT, "Other");
string index = await element.GetOptionAsync(HTMLElement.BY_INDEX);
string text = await element.GetOptionAsync(HTMLElement.BY_TEXT);
string value = await element.GetOptionAsync(HTMLElement.BY_VALUE);
```

...

```
await tester.TestEndAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

---

## GetStyleAsync

### GetStyleAsync

**Описание:** метод возвращает стиль элемента из указанного свойства

**Синтаксис:** GetStyleAsync(string property)

**Возвращаемое значение:** строка (string)

**Пример:**

```
string style = await element.GetStyleAsync("width");
```

---

Created with the Personal Edition of HelpNDoc: [Easily create HTML Help documents](#)

---

## GetTextAsync

### GetTextAsync

**Описание:** метод возвращает текст из элемента

**Синтаксис:** GetTextAsync();

**Возвращаемое значение:** строка (string)

**Пример:**

```
string text = await element.GetTextAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

---

## GetValueAsync

### GetValueAsync

**Описание:** метод возвращает значение из элемента

**Синтаксис:** GetValueAsync();

**Возвращаемое значение:** строка (string)

**Пример:**

```
string value = await element.GetValueAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Easily create EBooks](#)

---

## IsClickableAsync

### IsClickableAsync

**Описание:** метод определяет кликабельность элемента и возвращает true или false

**Синтаксис:** IsClickableAsync()

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
Tester tester = new Tester(browserForm);
```

```
await tester.TestBeginAsync();
await tester.GoToUrlAsync("https://somovstudio.github.io/test2.html", 5);
HTMLElement element = await tester.GetElementAsync(Tester.BY_XPATH, "//button");
bool clickable = await element.IsClickableAsync();
await tester.AssertTrueAsync(clickable);
await tester.TestEndAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

---

## MakeVisibleAsync

### MakeVisibleAsync

**Описание:** метод делает элемент видимым путем добавления стилей

**Синтаксис:** MakeVisibleAsync(string by, string locator, string visibility = "visible", int opacity = 1, int index = 1000)

**Пример:**

```
await tester.MakeVisibleAsync(Tester.BY_XPATH, "//*[@id='button']");
```

```
await tester.MakeVisibleAsync(Tester.BY_BY_CSS, "//*[@id='button']");
```

---

Created with the Personal Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

---

## ScrollToAsync

### ScrollToAsync

**Описание:** метод выполняет прокрутку к элементу (параметр behaviorSmooth определяет плавность прокрутки)

**Синтаксис:** ScrollToAsync(bool behaviorSmooth = false);

**Возвращаемое значение:** отсутствует

**Пример:**

```
await element.ScrollToAsync();
```

```
await element.ScrollToAsync(true);
```

```
await element.ScrollToAsync(false);
```

---

Created with the Personal Edition of HelpNDoc: [Free CHM Help documentation generator](#)

---

## SelectOptionAsync

### SelectOptionAsync

**Описание:** метод выбирает option из элемента select по указанному индексу, тексту или значению.

**Синтаксис:** SelectOptionAsync(string by, string value)

**Возвращаемое значение:** отсутствует

**Пример:**

```
Tester tester = new Tester(browserForm);
```

```
await tester.TestBeginAsync();
await tester.GoToUrlAsync("https://somovstudio.github.io/test2.html", 5);
HTMLInputElement element = await tester.GetElementAsync(Tester.BY_XPATH,
"//*[@id='MySelect']");
await element.SelectOptionAsync(HTMLInputElement.BY_INDEX, "2");
await element.SelectOptionAsync(HTMLInputElement.BY_VALUE, "Mobile");
await element.SelectOptionAsync(HTMLInputElement.BY_TEXT, "Other");
string index = await element.GetOptionAsync(HTMLInputElement.BY_INDEX);
string text = await element.GetOptionAsync(HTMLInputElement.BY_TEXT);
string value = await element.GetOptionAsync(HTMLInputElement.BY_VALUE);
...
await tester.TestEndAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Easy CHM and documentation editor](#)

---

## SetAttributeAsync

### SetAttributeAsync

**Описание:** метод вставляет значение указанного атрибута в элемент

**Синтаксис:** SetAttributeAsync(string name, string value);

**Возвращаемое значение:** отсутствует

**Пример:**

```
await element.SetAttributeAsync("class", "my-class");
```

---

Created with the Personal Edition of HelpNDoc: [Free EPub and documentation generator](#)

---

## SetHtmlAsync

### SetHtmlAsync

**Описание:** метод вставляет html в элемент

**Синтаксис:** SetHtmlAsync(string html);

**Возвращаемое значение:** отсутствует

**Пример:**

```
await element.SetHtmlAsync("<div>Это тест</div>");
```

---

Created with the Personal Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

---

## SetStyleAsync

### SetStyleAsync

**Описание:** метод устанавливает стиль элемента

**Синтаксис:** SetStyleAsync(string cssText)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await element.SetStyleAsync("width: 250px; background-color: #000000;");
```

## SetTextAsync

### SetTextAsync

**Описание:** метод вставляет текст в элемент

**Синтаксис:** `SetTextAsync(string text);`

**Возвращаемое значение:** отсутствует

**Пример:**

```
await element.SetTextAsync("это тестовый текст");
```

## SetValueAsync

### SetValueAsync

**Описание:** метод вставляет значение в элемент

**Синтаксис:** `SetValueAsync(string value);`

**Возвращаемое значение:** отсутствует

**Пример:**

```
await element.SetValueAsync("это тестовое значение");
```

## WaitNotVisibleAsync

### WaitNotVisible

**Описание:** метод ждет указанное количество секунд пока элемент перестанет отображаться

**Синтаксис:** `WaitNotVisibleAsync(int sec)`

**Возвращаемое значение:** отсутствует

**Пример:**

```
await element.WaitNotVisibleAsync(2);
```

## WaitVisibleAsync

### WaitVisibleAsync

**Описание:** метод ждет отображение элемента указанное количество секунд

**Синтаксис:** `WaitVisibleAsync(int sec)`

**Возвращаемое значение:** отсутствует

**Пример:**

```
await element.WaitVisibleAsync(2);
```

---

Created with the Personal Edition of HelpNDoc: [Free HTML Help documentation generator](#)

---

## Класс: FRAMEElement

---

Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

---

### Конструктор

---

Created with the Personal Edition of HelpNDoc: [Free help authoring tool](#)

---

FRAMEElement

**FRAMEElement**

**Описание:** вспомогательный класс для работы с фреймом указанного индекса

**Синтаксис:** FRAMEElement(Tester tester, int index)

**Пример получения объекта:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);  
tester.ConsoleMsg("Index: " + frame.Index);  
tester.ConsoleMsg("Name: " + frame.Name);
```

---

Created with the Personal Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

---

### Константы

---

Created with the Personal Edition of HelpNDoc: [Produce electronic books easily](#)

---

BY\_INDEX

**BY\_INDEX**

**Описание:** константа обозначает тип обрабатываемого значения

**Синтаксис:** BY\_INDEX = "BY\_INDEX"

**Пример:**

```
FRAMEElement.BY_INDEX
```

---

Created with the Personal Edition of HelpNDoc: [Easily create HTML Help documents](#)

---

BY\_TEXT

**BY\_TEXT**

**Описание:** константа обозначает тип обрабатываемого значения

**Синтаксис:** BY\_TEXT = "BY\_TEXT"

**Пример:**

```
FRAMEElement.BY_TEXT
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

---

BY\_VALUE

BY\_VALUE

**Описание:** константа обозначает тип обрабатываемого значения

**Синтаксис:** BY\_VALUE = "BY\_VALUE"

**Пример:**

FRAMEElement.BY\_VALUE

---

Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

---

Переменные

---

Created with the Personal Edition of HelpNDoc: [Full-featured eBook editor](#)

---

Name

Name

**Описание:** переменная возвращает или получает имя (Name) фрейма

**Синтаксис:** string Name { get; set; }

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
tester.ConsoleMsg("Name: " + frame.Name);
```

---

Created with the Personal Edition of HelpNDoc: [Free CHM Help documentation generator](#)

---

Index

Index

**Описание:** переменная возвращает или получает индекс (Index) фрейма

**Синтаксис:** int Index { get; set; }

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
tester.ConsoleMsg("Index: " + frame.Index);
```

---

Created with the Personal Edition of HelpNDoc: [iPhone web sites made easy](#)

---

Index

Index

**Описание:** переменная возвращает или получает индекс (Index) фрейма

**Синтаксис:** int Index { get; set; }

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
tester.ConsoleMsg("Index: " + frame.Index);
```

## Методы

### ClickElementAsync

#### ClickElementAsync

**Описание:** метод выполняет нажатие на элемент

**Синтаксис:** ClickElementAsync(string by, string locator)

**Возвращаемое значение:** отсутствует

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
await frame.ClickElementAsync(Tester.BY_CSS, "#auth #buttonLogin");
```

```
await frame.ClickElementAsync(Tester.BY_XPATH,
"//div[@id='auth']//input[@id='buttonLogin']");
```

### FindElementAsync

#### FindElementAsync

**Описание:** метод выполняет поиск элемент в DOM с ожиданием в секундах и возвращает логический результат true или false

**Синтаксис:** FindElementAsync(string by, string locator, int sec)

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
bool result = await frame.FindElementAsync(Tester.BY_CSS, "div[id='result']", 2);
```

```
bool result = await frame.FindElementAsync(Tester.BY_XPATH, "//div[@id='result']", 2);
```

### FindVisibleElementAsync

#### FindVisibleElementAsync

**Описание:** метод выполняет поиск визуально отображаемого элемента с ожиданием в секундах и возвращает логический результат true или false

**Синтаксис:** FindVisibleElementAsync(string by, string locator, int sec)

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
```

```
bool result = await frame.FindVisibleElementAsync(Tester.BY_CSS, "#auth #buttonLogin", 2);
```

```
bool result = await frame.FindVisibleElementAsync(Tester.BY_XPATH,
"//div[@id='auth']//input[@id='buttonLogin']", 2);
```

---

Created with the Personal Edition of HelpNDoc: [Produce electronic books easily](#)

---

## GetAttributeFromElementAsync

### GetAttributeFromElementAsync

**Описание:** метод возвращает строчное значение из указанного атрибута в выбранном элементе

**Синтаксис:** GetAttributeFromElementAsync(string by, string locator, string attribute)

**Возвращаемое значение:** строка (string)

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
```

```
string value = await frame.GetAttributeFromElementAsync(Tester.BY_CSS, "input", "name");
```

```
string value = await frame.GetAttributeFromElementAsync(Tester.BY_XPATH, "//input", "name");
```

---

Created with the Personal Edition of HelpNDoc: [Free Qt Help documentation generator](#)

---

## GetAttributeFromElementsAsync

### GetAttributeFromElementsAsync

**Описание:** метод возвращает список значений указанного атрибута из множества элементов

**Синтаксис:** GetAttributeFromElementsAsync(string by, string locator, string attribute)

**Возвращаемое значение:** список (List)

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
```

```
List<string> values = await frame.GetAttributeFromElementsAsync(Tester.BY_CSS, "input",
"name");
```

```
if(values != null)
```

```
{
    foreach (string attr in values)
        tester.ConsoleMsg(attr);
}
```

```
List<string> values = await frame.GetAttributeFromElementsAsync(Tester.BY_XPATH, "//input",
"name");
```

```
if(values != null)
```

```
{
    foreach (string attr in values)
```

```

    tester.ConsoleMsg(attr);
}

```

---

Created with the Personal Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

---

## GetCountElementsAsync

### GetCountElementsAsync

**Описание:** метод возвращает количество найденных элементов

**Синтаксис:** GetCountElementsAsync(string by, string locator)

**Возвращаемое значение:** целое число (int)

**Пример:**

```

FRAMEElement frame = await tester.GetFrameAsync(0);
int count = await frame.GetCountElementsAsync(Tester.BY_CSS, "input");

```

```

int count = await frame.GetCountElementsAsync(Tester.BY_XPATH, "//input");

```

---

Created with the Personal Edition of HelpNDoc: [Easily create Qt Help files](#)

---

## GetHtmlFromElementAsync

### GetHtmlFromElementAsync

**Описание:** метод возвращает html представление объекта в строчном выражении

**Синтаксис:** GetHtmlFromElementAsync(string by, string locator)

**Возвращаемое значение:** строка (string)

**Пример:**

```

FRAMEElement frame = await tester.GetFrameAsync(0);
string html = await frame.GetHtmlFromElementAsync(Tester.BY_CSS, "#auth > h2");

```

```

string html = await frame.GetHtmlFromElementAsync(Tester.BY_XPATH,
"//div[@id='auth']//h2");

```

---

Created with the Personal Edition of HelpNDoc: [Qt Help documentation made easy](#)

---

## GetOptionAsync

### GetOptionAsync

**Описание:** метод возвращает индекс, текст или значение выбоанной опции выбранной option из элемента select

**Синтаксис:** GetOptionAsync(string by, string locator, string type)

**Возвращаемое значение:** строка (string)

**Пример:**

```

FRAMEElement frame = await tester.GetFrameAsync(0);

```

```

string index = await frame.GetOptionAsync(Tester.BY_CSS, "#MySelect",

```

```
FRAMEElement.BY_INDEX);
string value = await frame.GetOptionAsync(Tester.BY_CSS, "#MySelect",
FRAMEElement.BY_VALUE);
string text = await frame.GetOptionAsync(Tester.BY_CSS, "#MySelect",
FRAMEElement.BY_TEXT);

string index = await frame.GetOptionAsync(Tester.BY_XPATH, "//*[@id='MySelect']",
FRAMEElement.BY_INDEX);
string value = await frame.GetOptionAsync(Tester.BY_XPATH, "//*[@id='MySelect']",
FRAMEElement.BY_VALUE);
string text = await frame.GetOptionAsync(Tester.BY_XPATH, "//*[@id='MySelect']",
FRAMEElement.BY_TEXT);
```

---

Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

---

## GetStyleFromElementAsync

### GetStyleFromElementAsync

**Описание:** метод возвращает стиль элемента из указанного свойства

**Синтаксис:** GetStyleFromElementAsync(string by, string locator, string property)

**Возвращаемое значение:** строка (string)

#### Пример:

```
FRAMEElement frame = await tester.GetFrameAsync(0);
string style = await frame.GetStyleFromElementAsync(Tester.BY_CSS, "#auth > h2", "width");

string style = await frame.GetStyleFromElementAsync(Tester.BY_XPATH, "//div[@id='auth']",
"width");
```

---

Created with the Personal Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

---

## GetTextFromElementAsync

### GetTextFromElementAsync

**Описание:** метод возвращает текст из указанного элемента

**Синтаксис:** GetTextFromElementAsync(string by, string locator)

**Возвращаемое значение:** строка (string)

#### Пример:

```
FRAMEElement frame = await tester.GetFrameAsync(0);
string text = await frame.GetTextFromElementAsync(Tester.BY_CSS, "#auth > h2");

string text = await frame.GetTextFromElementAsync(Tester.BY_XPATH, "//div[@id='auth']/h2");
```

---

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

---

## GetTitleAsync

### GetTitleAsync

**Описание:** метод возвращает заголовок страницы

**Синтаксис:** GetTitleAsync()

**Возвращаемое значение:** строка (string)

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);  
string title = await frame.GetTitleAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

---

## GetUrlAsync

### GetUrlAsync

**Описание:** метод возвращает текущий URL

**Синтаксис:** GetUrlAsync()

**Возвращаемое значение:** строка (string)

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);  
string url = await frame.GetUrlAsync();
```

---

Created with the Personal Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

---

## GetValueFromElementAsync

### GetValueFromElementAsync

**Описание:** метод возвращает значение из указанного элемента

**Синтаксис:** GetValueFromElementAsync(string by, string locator)

**Возвращаемое значение:** строка (string)

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);  
string value = await frame.GetValueFromElementAsync(Tester.BY_CSS, "input[id='login']");
```

```
string value = await frame.GetValueFromElementAsync(Tester.BY_XPATH,  
"//input[@id='login']");
```

---

Created with the Personal Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

---

## IsClickableElementAsync

### IsClickableElementAsync

**Описание:** метод определяет кликабельность элемента и возвращает true или false

**Синтаксис:** IsClickableElementAsync(string by, string locator)

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
bool clickable = await frame.IsClickableElementAsync(Tester.BY_CSS, "#buttonLogin");
await tester.AssertTrueAsync(clickable);
```

```
bool clickable = await frame.IsClickableElementAsync(Tester.BY_XPATH,
"//*[@id='buttonLogin']");
await frame.AssertTrueAsync(clickable);
```

---

Created with the Personal Edition of HelpNDoc: [Easily create EBooks](#)

---

**IsVisibleElementAsync****IsVisibleElementAsync**

**Описание:** метод определяет видимость элемента и возвращает значение true или false

**Синтаксис:** IsVisibleElementAsync(string by, string locator)

**Возвращаемое значение:** логическое (true или false)

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
```

```
bool result = await frame.IsVisibleElementAsync(Tester.BY_XPATH, "//*[@id='login']");
```

```
bool result = await frame.IsVisibleElementAsync(Tester.BY_CSS, "#login");
```

```
await frame.AssertTrueAsync(result);
```

---

Created with the Personal Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

---

**MakeVisibleAsync****MakeVisibleAsync**

**Описание:** метод делает элемент видимым путем добавления стилей

**Синтаксис:** MakeVisibleAsync(string by, string locator, string visibility = "visible", int opacity = 1, int index = 1000)

**Пример:**

```
await tester.MakeVisibleAsync(Tester.BY_XPATH, "//*[@id='button']");
```

```
await tester.MakeVisibleAsync(Tester.BY_BY_CSS, "//*[@id='button']");
```

---

Created with the Personal Edition of HelpNDoc: [Easily create Web Help sites](#)

---

**ScrollToElementAsync****ScrollToElementAsync**

**Описание:** метод выполняет прокрутку к указанному элементу (параметр behaviorSmooth

определяет плавность прокрутки)

**Синтаксис:** ScrollToElementAsync(string by, string locator, bool behaviorSmooth = false)

**Возвращаемое значение:** отсутствует

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
await frame.ScrollToElementAsync(Tester.BY_CSS, "body > footer", true);
```

```
await frame.ScrollToElementAsync(Tester.BY_XPATH, "/html/body/footer", true);
```

---

Created with the Personal Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

---

## SelectOptionAsync

### SelectOptionAsync

**Описание:** метод выбирает option из элемента select по указанному индексу, тексту или значению

**Синтаксис:** SelectOptionAsync(string by, string locator, string type, string value)

**Возвращаемое значение:** отсутствует

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
await frame.SelectOptionAsync(Tester.BY_CSS, "#MySelect", FRAMEElement.BY_INDEX, "2");
await frame.SelectOptionAsync(Tester.BY_CSS, "#MySelect", FRAMEElement.BY_VALUE,
"Mobile");
await frame.SelectOptionAsync(Tester.BY_CSS, "#MySelect", FRAMEElement.BY_TEXT, "Other");
```

```
await frame.SelectOptionAsync(Tester.BY_XPATH, "//*[@id='MySelect']",
FRAMEElement.BY_INDEX, "2");
await frame.SelectOptionAsync(Tester.BY_XPATH, "//*[@id='MySelect']",
FRAMEElement.BY_VALUE, "Mobile");
await frame.SelectOptionAsync(Tester.BY_XPATH, "//*[@id='MySelect']",
FRAMEElement.BY_TEXT, "Other");
```

---

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

---

## SetAttributeInElementAsync

### SetAttributeInElementAsync

**Описание:** метод вставляет атрибут со значением в указанный элемент

**Синтаксис:** SetAttributeInElementAsync(string by, string locator, string attribute, string value)

**Возвращаемое значение:** отсутствует

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
await frame.SetAttributeInElementAsync(Tester.BY_CSS, "#auth > h2", "name", "test");
```

```
await frame.SetAttributeInElementAsync(Tester.BY_XPATH, "//div[@id='auth']/h2", "name", "test");
```

---

Created with the Personal Edition of HelpNDoc: [Free EPub and documentation generator](#)

---

## SetAttributeInElementsAsync

### SetAttributeInElementsAsync

**Описание:** метод вставить атрибут с указанным значением в множество элементов и в результате возвращает список

**Синтаксис:** SetAttributeInElementsAsync(string by, string locator, string attribute, string value)

**Возвращаемое значение:** список (List)

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
```

```
List<string> values = await frame.SetAttributeInElementsAsync(Tester.BY_CSS, "input", "class", "test-class");
foreach (string value in values)
{
    tester.ConsoleMsg(value);
}
```

```
List<string> values = await frame.SetAttributeInElementsAsync(Tester.BY_XPATH, "//input", "class", "test-class");
foreach (string value in values)
{
    tester.ConsoleMsg(value);
}
```

---

Created with the Personal Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

---

## SetHtmlInElementAsync

### SetHtmlInElementAsync

**Описание:** метод вставляет html представление объекта в указанный элемент

**Синтаксис:** SetHtmlInElementAsync(string by, string locator, string html)

**Возвращаемое значение:** отсутствует

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
```

```
await frame.SetHtmlInElementAsync(Tester.BY_CSS, "#auth > h2", "<div>Тестовый блок</div>");
```

```
await frame.SetHtmlInElementAsync(Tester.BY_XPATH, "//div[@id='auth']/h2", "<div>Тестовый
```

```
блок</div>");
```

---

Created with the Personal Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

---

## SetStyleInElementAsync

### SetStyleInElementAsync

**Описание:** метод устанавливает стиль элемента

**Синтаксис:** SetStyleInElementAsync(string by, string locator, string cssText)

**Возвращаемое значение:** отсутствует

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
await frame.SetStyleInElementAsync(Tester.BY_XPATH, "//div[@id='auth']", "background-color:
#000000;");
```

```
await frame.SetStyleInElementAsync(Tester.BY_CSS, "#auth", "background-color: #000000;");
```

---

Created with the Personal Edition of HelpNDoc: [Qt Help documentation made easy](#)

---

## SetTextInElementAsync

### SetTextInElementAsync

**Описание:** метод вставляет текст в указанный элемент

**Синтаксис:** SetTextInElementAsync(string by, string locator, string text)

**Возвращаемое значение:** отсутствует

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
await frame.SetTextInElementAsync(Tester.BY_CSS, "#auth > h2", "Тестовый заголовок");
```

```
await frame.SetTextInElementAsync(Tester.BY_XPATH, "//div[@id='auth']//h2", "Тестовый
заголовок");
```

---

Created with the Personal Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

---

## SetValueInElementAsync

### SetValueInElementAsync

**Описание:** метод вставляет значение в указанный элемент

**Синтаксис:** SetValueInElementAsync(string by, string locator, string value)

**Возвращаемое значение:** отсутствует

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
await frame.SetValueInElementAsync(Tester.BY_CSS, "input[id='login']", "admin");
await frame.SetValueInElementAsync(Tester.BY_CSS, "input[id='pass']", "0000");
```

```
await frame.SetValueInElementAsync(Tester.BY_XPATH, "//input[@id='login']", "admin");
await frame.SetValueInElementAsync(Tester.BY_XPATH, "//input[@id='pass']", "0000");
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured EBook editor](#)

---

## WaitNotVisibleElementAsync

### WaitNotVisibleElementAsync

**Описание:** метод выполняет временную остановку выполнения теста на указанное количество секунд и ждет когда запрашиваемый элемент перестанет отображаться

**Синтаксис:** WaitNotVisibleElementAsync(string by, string locator, int sec)

**Возвращаемое значение:** отсутствует

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
await frame.WaitNotVisibleElementAsync(Tester.BY_CSS, "div[id='result']", 2);
```

```
await frame.WaitNotVisibleElementAsync(Tester.BY_XPATH, "//div[@id='result']", 2);
```

---

Created with the Personal Edition of HelpNDoc: [Create iPhone web-based documentation](#)

---

## WaitVisibleElementAsync

### WaitVisibleElementAsync

**Описание:** метод выполняет временную остановку выполнения теста на указанное количество секунд и ждет когда запрашиваемый элемент отобразится

**Синтаксис:** WaitVisibleElementAsync(string by, string locator, int sec)

**Возвращаемое значение:** отсутствует

**Пример:**

```
FRAMEElement frame = await tester.GetFrameAsync(0);
await frame.WaitVisibleElementAsync(Tester.BY_CSS, "div[id='result']", 2);
```

```
await frame.WaitVisibleElementAsync(Tester.BY_XPATH, "//div[@id='result']", 2);
```

---

Created with the Personal Edition of HelpNDoc: [News and information about help authoring tools and software](#)

---

## Плагины

---

Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

---

### Плагин HatPluginMySql

---

Created with the Personal Edition of HelpNDoc: [Easily create EPub books](#)

---

#### Установка плагина

### HatPluginMySql

1. **Hat**  
: <https://github.com/SomovStudio/Hat/releases>

2. **HatPluginMySql**  
: <https://github.com/SomovStudio/HatPluginMySql/releases>

3. **HatPluginMySql-1.0.0.zip**

Имя	Дата изменения	Тип	Размер
plugin	26.07.2023 10:45	Папка с файлами	
LICENSE	18.07.2023 11:34	Файл	18 КБ
Readme.txt	26.07.2023 10:46	Файл "ТХТ"	1 КБ

4. **plugin** **Hat**

The screenshot shows a Windows File Explorer window with the following structure:

- Left sidebar (Folders):** runtimes, Hat.exe, HatFramework.dll, help.chm, ICSharpCode.AvalonEdit.dll, LICENSE, Microsoft.Web.WebView2.Core.dll, Microsoft.Web.WebView2.WinForms.dll, Microsoft.Web.WebView2.Wpf.dll, Newtonsoft.Json.dll, readme.html.
- Navigation pane:** Быстрый доступ, OneDrive - Personal, Этот компьютер, Сеть.
- Main pane (Files):**
  - BouncyCastle.Crypto.dll (21.07.2023 9:54)
  - Google.Protobuf.dll (21.07.2023 17:59)
  - HatFramework.dll (21.07.2023 9:54)
  - HatPluginMySql.dll (26.07.2023 10:35)** - highlighted with a red box
  - K4os.Compression.LZ4.dll (21.07.2023 9:54)
  - K4os.Compression.LZ4.Streams.dll (21.07.2023 9:54)
  - K4os.Hash.xxHash.dll (21.07.2023 9:54)
  - Microsoft.Bcl.AsyncInterfaces.dll (21.07.2023 17:59)
  - MySQL.Data.dll (21.07.2023 9:54)
  - Newtonsoft.Json.dll (17.03.2021 20:03)
  - System Buffers.dll (21.07.2023 9:54)
  - System.Configuration.ConfigurationMan... (21.07.2023 17:59)
  - System.Diagnostics.DiagnosticSource.dll (21.07.2023 9:54)
  - System.IO.Pipelines.dll (21.07.2023 17:59)
  - System.Memory.dll (21.07.2023 9:54)
  - System.Numerics.Vectors.dll (21.07.2023 9:54)
  - System.Runtime.CompilerServices.Unsaf... (21.07.2023 9:54)
  - System.Security.AccessControl.dll (21.07.2023 17:59)
  - System.Security.Permissions.dll (21.07.2023 17:59)
  - System.Security.Principal.Windows.dll (21.07.2023 17:59)
  - System.Threading.Tasks.Extensions.dll (21.07.2023 9:54)
  - ZstdSharp.dll (21.07.2023 17:59)

Имя	Дата изменения	Тип	Размер
runtimes	27.07.2023 11:26	Папка с файлами	
BouncyCastle.Crypto.dll	21.07.2023 9:54	Расширение при...	3 240 КБ
Google.Protobuf.dll	21.07.2023 17:59	Расширение при...	434 КБ
Hat.exe	26.06.2023 11:08	Приложение	1 689 КБ
HatFramework.dll			
HatPluginMySQL.dll			
help.chm			
ICSharpCode.AvalonEdit.dll			
K4os.Compression.LZ4.dll			
K4os.Compression.LZ4.Strea...			
K4os.Hash.xxHash.dll			
LICENSE			
Microsoft.Bcl.AsyncInterfac...			
Microsoft.Web.WebView2.Co...			
Microsoft.Web.WebView2.Wi...			
Microsoft.Web.WebView2.Wy...			
MySQL.Data.dll			
Newtonsoft.Json.dll			
readme.html			
System Buffers.dll	21.07.2023 9:54	Расширение при...	21 КБ
System.Configuration.Configu...	21.07.2023 17:59	Расширение при...	90 КБ
System.Diagnostics.DiagnosticSource.dll	21.07.2023 9:54	Расширение при...	170 КБ
System.IO.Pipelines.dll	21.07.2023 17:59	Расширение при...	84 КБ
System.Memory.dll	21.07.2023 9:54	Расширение при...	139 КБ
System.Numerics.Vectors.dll	21.07.2023 9:54	Расширение при...	114 КБ
System.Runtime.CompilerServices.Unsaf...	21.07.2023 9:54	Расширение при...	18 КБ

Замена или пропуск файлов

Перемещение элементов (22) из plugin в Hat

В папке назначения есть файлы (2) с такими же именами

- ✓ Заменить файлы в папке назначения
- ↶ Пропустить эти файлы ←
- 📄 Запрашивать для каждого файла

Меньше сведений

Имя	Дата изменения	Тип	Размер
runtimes	27.07.2023 11:26	Папка с файлами	
BouncyCastle.Crypto.dll	21.07.2023 9:54	Расширение при...	3 240 КБ
Google.Protobuf.dll	21.07.2023 17:59	Расширение при...	434 КБ
Hat.exe	26.06.2023 11:08	Приложение	1 689 КБ
HatFramework.dll	26.06.2023 11:06	Расширение при...	325 КБ
HatPluginMySql.dll	26.07.2023 10:35	Расширение при...	26 КБ
help.chm	26.06.2023 11:28	Скомпилирован...	3 308 КБ
ICSharpCode.AvalonEdit.dll	03.04.2023 9:54	Расширение при...	606 КБ
K4os.Compression.LZ4.dll	21.07.2023 9:54	Расширение при...	66 КБ
K4os.Compression.LZ4.Streams.dll	21.07.2023 9:54	Расширение при...	73 КБ
K4os.Hash.xxHash.dll	21.07.2023 9:54	Расширение при...	13 КБ
LICENSE	28.02.2023 14:56	Файл	2 КБ
Microsoft.Bcl.AsyncInterfaces.dll	21.07.2023 17:59	Расширение при...	27 КБ
Microsoft.Web.WebView2.Core.dll	28.05.2023 11:30	Расширение при...	462 КБ
Microsoft.Web.WebView2.WinForms.dll	28.05.2023 11:30	Расширение при...	38 КБ
Microsoft.Web.WebView2.Wpf.dll	28.05.2023 11:30	Расширение при...	44 КБ
MySql.Data.dll	21.07.2023 9:54	Расширение при...	1 144 КБ
Newtonsoft.Json.dll	16.03.2023 13:00	Расширение при...	696 КБ
readme.html	26.06.2023 11:08	Chrome HTML Do...	25 КБ
System Buffers.dll	21.07.2023 9:54	Расширение при...	21 КБ
System.Configuration.ConfigurationMan...	21.07.2023 17:59	Расширение при...	90 КБ
System.Diagnostics.DiagnosticSource.dll	21.07.2023 9:54	Расширение при...	170 КБ
System.IO.Pipelines.dll	21.07.2023 17:59	Расширение при...	84 КБ
System.Memory.dll	21.07.2023 9:54	Расширение при...	139 КБ
System.Numerics.Vectors.dll	21.07.2023 9:54	Расширение при...	114 КБ
System.Runtime.CompilerServices.Unsaf...	21.07.2023 9:54	Расширение при...	18 КБ
System.Security.AccessControl.dll	21.07.2023 17:59	Расширение при...	36 КБ
System.Security.Permissions.dll	21.07.2023 17:59	Расширение при...	30 КБ
System.Security.Principal.Windows.dll	21.07.2023 17:59	Расширение при...	18 КБ
System.Threading.Tasks.Extensions.dll	21.07.2023 9:54	Расширение при...	26 КБ
ZstdSharp.dll	21.07.2023 17:59	Расширение при...	427 КБ

!

---

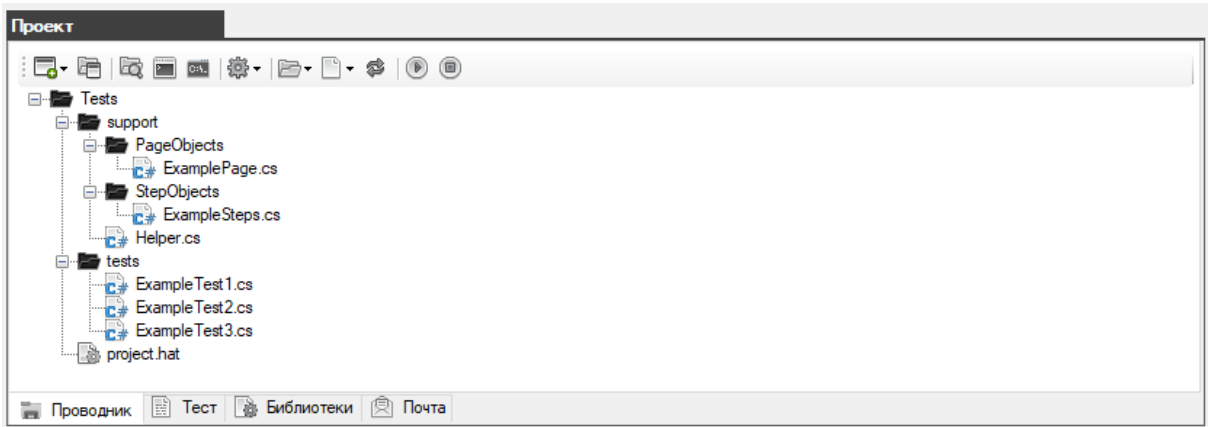
Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

---

## Подключение плагина к проекту

### HatPluginMySql

1. **Hat**
- 2.



3. " " HatPluginMySql.dll



4. " "

---

Created with the Personal Edition of HelpNDoc: [Free iPhone documentation generator](#)

---

Пример автотеста

MySQL  
test\_table

test\_db  
: id, name, age, post

The screenshot shows the phpMyAdmin interface. On the left, the server is identified as 'MySQL'. The database 'test\_db' is selected, and the table 'test\_table' is visible in the tree view. The main panel displays a successful query: 'SELECT \* FROM `test\_table`'. The result shows 3 rows of data:

id	name	age	post
1	John	30	Driver
2	Dave	35	Manager
3	Paul	45	Director

Each row has interactive options: 'Изменить' (Change), 'Копировать' (Copy), and 'Удалить' (Delete).

: ExampleTest3.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
using System.IO;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Reflection;
using Newtonsoft.Json;
using HatFramework;
using HatPluginMySQL;

namespace Hat
{
    public class ExampleTest3
    {
        Tester tester;

        public async void Main(Form browserWindow)
        {
            tester = new Tester(browserWindow);
        }
    }
}

```

```

        await setUp();
        await test();
        await tearDown();
    }

    public async Task setUp()
    {
        tester.Description("    #3
");
        await tester.BrowserFullScreenAsync();
    }

    public async Task test()
    {
        DataTable dataTable;
        List<List<string>> entries = new
List<List<string>>();
        TesterMySQL testerMySQL =
TesterMySQL(tester);

        await tester.TestBeginAsync();

        await
testerMySQL.ConnectionOpenAsync
("server=127.0.0.1;uid=root;pwd=;database=test_db");

        int count = await
testerMySQL.GetCountEntriesAsync("SELECT * FROM test_table");
        if (count > 0)
        {
            await =
testerMySQL.GetEntriesAsync("SELECT * FROM test_table");
            dataTable =
testerMySQL.GetDataTableAsync("SELECT * FROM test_table");
            foreach (DataRow row in dataTable.Rows)
                foreach (DataColumn col in
dataTable.Columns)
                    tester.ConsoleMsg(row[col].ToString()
);
        }

        await testerMySQL.SetEntryAsync("INSERT INTO
test_table VALUES(NULL, 'I am Tester', 100, 'My post QA')");
        bool result = await
testerMySQL.FindEntryAsync("test_table", "name", "'I am
Tester'");
        await
testerMySQL.AssertHaveInTableAsync("test_table", "name", "'I
am Tester'");
    }

```

```

        entries = await
testerMySQL.GetEntriesFromTableAsync("test_table");
        if (entries != null){
            foreach(List<string> entry in entries)
                foreach(string value in entry)
                    tester.ConsoleMsg(value);
        }

        await testerMySQL.EditEntryAsync("UPDATE
test_table SET age = 111 WHERE name = 'I am Tester'");
        result = await
testerMySQL.FindEntryAsync("test_table", "age", "111");
        await
testerMySQL.AssertHaveInTableAsync("test_table", "age",
"111");

        await testerMySQL.RemoveEntryAsync("DELETE FROM
test_table WHERE name = 'I am Tester'");
        result = await
testerMySQL.FindEntryAsync("test_table", "name", "'I am
Tester'");
        await
testerMySQL.AssertDontHaveInTableAsync("test_table", "name",
"'I am Tester'");

        await testerMySQL.ConnectionCloseAsync();

        await tester.TestEndAsync();
    }

    public async Task tearDown()
    {
        // await tester.BrowserCloseAsync();
    }
}
}

```

```
using HatPluginMySQL;
```

```

TesterMySQL testerMySQL = new TesterMySQL(tester);
await
testerMySQL.ConnectionOpenAsync
("server=127.0.0.1;uid=root;pwd=;database=test_db");

```

```

await testerMySQL.GetCountEntriesAsync("SELECT * FROM
test_table");
await testerMySQL.GetEntriesAsync("SELECT * FROM
test_table");
await testerMySQL.GetEntriesFromTableAsync("test_table");
await testerMySQL.GetDataTableAsync("SELECT * FROM
test_table");
await testerMySQL.SetEntryAsync("INSERT INTO test_table
VALUES(NULL, 'I am Tester', 100, 'My post QA')");
await testerMySQL.FindEntryAsync("test_table", "name", "'I
am Tester'");
await testerMySQL.EditEntryAsync("UPDATE test_table SET age
= 111 WHERE name = 'I am Tester'");
await testerMySQL.RemoveEntryAsync("DELETE FROM test_table
WHERE name = 'I am Tester'");

```

```

await testerMySQL.AssertHaveInTableAsync("test_table",
"name", "'I am Tester'");
await testerMySQL.AssertDontHaveInTableAsync("test_table",
"name", "'I am Tester'")

```

```

await testerMySQL.ConnectionCloseAsync();

```

Действие	Статус	Комментарий
Сообщение	В процессе	Запуск автотеста
Сообщение	В процессе	Запущен автотест из файла: ExampleTest3.cs
BrowserFullScreenAsync()	Выполнено	Размер браузера изменён
Тестирование началось	Выполнено	Инициализация теста
Инициализация теста	Выполнено	Выполнена инициализация теста
ConnectionOpenAsync("server=127.0.0.1;uid=root;pwd=test_db")	В процессе	Подключение к базе данных и открытие соединения
ConnectionOpenAsync("server=127.0.0.1;uid=root;pwd=test_db")	Успешно	Подключение к базе данных открыто
GetCountEntriesAsync("SELECT * FROM test_table")	Успешно	В таблице 3 записей
GetEntriesAsync("SELECT * FROM test_table")	Успешно	Получены записи из таблицы
GetDataTableAsync("SELECT * FROM test_table")	Успешно	Получена таблица записей
SetEntryAsync("INSERT INTO test_table VALUES(NULL, 'I am Tester', 100, 'My post QA')")	Успешно	Данные успешно добавлены в базу данных
FindEntryAsync("test_table", "name", "'I am Tester'")	Выполнено	В таблице test_table присутствует запись со значением 'I am Tester' в колонке name
AssertHaveInTableAsync("test_table", "name", "'I am Tester'")	Успешно	В таблице test_table присутствует запись со значением 'I am Tester' в колонке name
GetEntriesFromTableAsync("test_table")	Успешно	Получены все записи из таблицы "test_table"
EditEntryAsync("UPDATE test_table SET age = 111 WHERE name = 'I am Tester'")	Успешно	Данные успешно обновлены в базу данных
FindEntryAsync("test_table", "age", "111")	Выполнено	В таблице test_table присутствует запись со значением 111 в колонке age
AssertHaveInTableAsync("test_table", "age", "111")	Успешно	В таблице test_table присутствует запись со значением 111 в колонке age
RemoveEntryAsync("DELETE FROM test_table WHERE name = 'I am Tester'")	Успешно	Данные успешно удалены из базы данных
FindEntryAsync("test_table", "name", "'I am Tester'")	Выполнено	В таблице test_table в колонке name нет записи со значением 'I am Tester'
AssertDontHaveInTableAsync("test_table", "name", "'I am Tester'")	Успешно	В таблице test_table в колонке name нет записи со значением 'I am Tester'
ConnectionCloseAsync()	Успешно	Подключение к базе данных закрыто
Тестирование завершено	Успешно	Тест завершен - все шаги выполнены успешно

---

Created with the Personal Edition of HelpNDoc: [Create iPhone web-based documentation](#)

---

**Класс:** [TesterMySQL](#)

---

Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

---

**Конструктор**

---

Created with the Personal Edition of HelpNDoc: [Easy EBook and documentation generator](#)

---

TesterMySql

TesterMySql

**Описание:** основной класс работы с базой данных MySql

**Синтаксис:** TesterMySql(Tester tester)

**Пример:**

```
Tester tester = new Tester(browserWindow);
```

```
TesterMySql testerMySql = new TesterMySql(tester);
```

---

Created with the Personal Edition of HelpNDoc: [Easily create EPub books](#)

---

Методы

---

Created with the Personal Edition of HelpNDoc: [Full-featured EPub generator](#)

---

ConnectionOpenAsync

ConnectionOpenAsync

**Описание:** метод открывает соединение с базой данных MySQL

**Синтаксис:** ConnectionOpenAsync(string connectionString)

**Возвращаемое значение:** отсутствует

**Пример:**

```
await testerMySQL.ConnectionOpenAsync("server=127.0.0.1;uid=root;pwd=;database=test_db");
```

---

Created with the Personal Edition of HelpNDoc: [Free help authoring environment](#)

---

ConnectionCloseAsync

ConnectionCloseAsync

**Описание:** метод закрывает соединение с базой данных MySQL

**Синтаксис:** ConnectionCloseAsync()

**Возвращаемое значение:** отсутствует

**Пример:**

```
await testerMySQL.ConnectionCloseAsync();
```

## GetCountEntriesAsync

### GetCountEntriesAsync

**Описание:** метод возвращает количество записей в таблице после выполнения запроса

**Синтаксис:** GetCountEntriesAsync(string sqlQuerySelect)

**Возвращаемое значение:** целое число (int)

**Пример:**

```
int count = await testerMySQL.GetCountEntriesAsync("SELECT * FROM test_table");
```

## GetEntriesAsync

### GetEntriesAsync

**Описание:** метод возвращает список записей из таблицы после выполнения запроса

**Синтаксис:** GetEntriesAsync(string sqlQuerySelect)

**Возвращаемое значение:** список (List)

**Пример:**

```
List<List<string>> entries = new List<List<string>>();  
entries = await testerMySQL.GetEntriesAsync("SELECT * FROM test_table");
```

```
foreach(List<string> entry in entries)  
    foreach(string value in entry)  
        tester.ConsoleMsg(value);
```

---

Created with the Personal Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

---

## GetEntriesFromTableAsync

### GetEntriesFromTableAsync

**Описание:** метод возвращает список записей из указанной таблицы базы данных

**Синтаксис:** GetEntriesFromTableAsync(string tableName)

**Возвращаемое значение:** список (List)

**Пример:**

```
List<List<string>> entries = new List<List<string>>();  
entries = await testerMySQL.GetEntriesFromTableAsync("test_table");
```

```
foreach(List<string> entry in entries)  
    foreach(string value in entry)  
        tester.ConsoleMsg(value);
```

## GetDataTableAsync

### GetDataTableAsync

**Описание:** метод возвращает таблицу записей из таблицы базы данных после выполнения запроса

**Синтаксис:** GetDataTableAsync(string sqlQuerySelect)

**Возвращаемое значение:** таблица (DataTable)

**Пример:**

```
DataTable dataTable = null;
```

```
dataTable = await testerMySql.GetDataTableAsync("SELECT * FROM test_table");
```

```
foreach (DataRow row in dataTable.Rows)
```

```
    foreach (DataColumn col in dataTable.Columns)
```

```
        tester.ConsoleMsg(row[col].ToString());
```

## SetEntryAsync

### SetEntryAsync

**Описание:** метод выполняет запрос который добавляет данные в таблицу базы данных и возвращает номер записи

**Синтаксис:** SetEntryAsync(string sqlQueryInsert)

**Возвращаемое значение:** целое число (int)

**Пример:**

```
int result = await testerMySQL.SetEntryAsync("INSERT INTO test_table VALUES(NULL, 'I am Tester', 100, 'My post QA');");
```

## EditEntryAsync

### EditEntryAsync

**Описание:** метод выполняет запрос который изменяет данные в таблице базы данных и возвращает номер записи

**Синтаксис:** EditEntryAsync(string sqlQueryUpdate)

**Возвращаемое значение:** целое число (int)

### Пример:

```
int result = await testerMySQL.EditEntryAsync("UPDATE test_table SET age = 111 WHERE name = 'I am Tester'");
```

---

Created with the Personal Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

## RemoveEntryAsync

### RemoveEntryAsync

**Описание:** метод выполняет запрос который удаляет данные в таблице базы данных и возвращает номер записи

**Синтаксис:** RemoveEntryAsync(string sqlQueryDelete)

**Возвращаемое значение:** целое число (int)

#### Пример:

```
int result = await testerMySQL.RemoveEntryAsync("DELETE FROM test_table WHERE name = 'I am Tester'");
```

## FindEntryAsync

### FindEntryAsync

**Описание:** метод выполняет поиск данные в указанной таблице базы данных и возвращает логическое значение результата поиска

**Синтаксис:** FindEntryAsync(string tableName, string columnName, string value)

**Возвращаемое значение:** логическое (true или false)

#### Пример:

```
bool result = await testerMySql.FindEntryAsync("test_table", "name", "I am Tester");
```

---

Created with the Personal Edition of HelpNDoc: [Free help authoring environment](#)

---

## AssertHaveInTableAsync

### AssertHaveInTableAsync

**Описание:** метод выполняет проверку данные в указанной таблице базы данных и возвращает логическое значение результата поиска, в случае отрицательного результата проверка будет считаться провальной

**Синтаксис:** AssertHaveInTableAsync(string tableName, string columnName, string value)

**Возвращаемое значение:** логическое (true или false)

### Пример:

```
bool result = await testerMySql.AssertHaveInTableAsync("test_table", "name", "I am Tester");
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

---

## AssertDontHaveInTableAsync

### AssertDontHaveInTableAsync

**Описание:** метод выполняет проверку данные в указанной таблице базы данных и возвращает логическое значение результата поиска, в случае положительного результата проверка будет считаться провальной

**Синтаксис:** AssertDontHaveInTableAsync(string tableName, string columnName, string value)

**Возвращаемое значение:** логическое (true или false)

#### Пример:

```
bool result = await testerMySQL.AssertDontHaveInTableAsync("test_table", "name", "I am Tester");
```

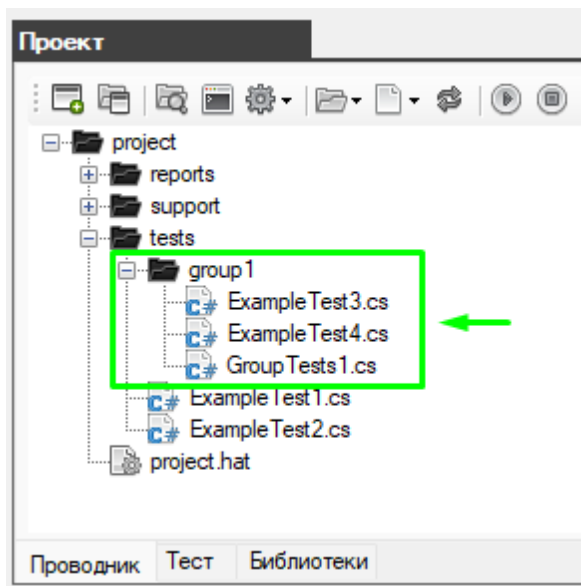
Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

## Практические примеры

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

### Группа автотестов

( group1)



group1.

:

Main

Task

ExampleTest3.cs

: ExampleTest3.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
using System.IO;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Reflection;
using Newtonsoft.Json;
using HatFramework;

namespace Hat
{
    // JSON
    public class TestJson
    {
        public int userId { get; set; }
        public int id { get; set; }
        public string title { get; set; }
        public string body { get; set; }
    }

    public class ExampleTest3
    {
        //
        Tester tester;

        // Task ! (
        public async Task Main(Form browserWindow)
        {
            //
            tester = new Tester(browserWindow);

            await setUp();
            await test();
            await tearDown();
        }

        //
        public async Task setUp()
        {
            await tester.BrowserFullScreenAsync(); //
        }

        //
        public async Task test()
        {
            await tester.TestBeginAsync();
        }
    }
}

```

```

        await
tester.GoToUrlAsync("https://jsonplaceholder.typicode.com", 5);
        string result = await
tester.RestGetAsync(@"https://jsonplaceholder.typicode.com/posts/1/",
    TimeSpan.FromDays(1), "UTF-8");
        TestJson dataJson =
JsonConvert.DeserializeObject<TestJson>(result);
        tester.AssertEqualsAsync("1", dataJson.userId.ToString());
        await tester.TestEndAsync();
    }

    //
    public async Task tearDown()
    {
        // await tester.BrowserCloseAsync();
    }
}
}

```

### ExampleTest4.cs

```

: ExampleTest4.cs

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
using System.IO;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Reflection;
using Newtonsoft.Json;
using HatFramework;

namespace Hat
{
    public class ExampleTest4
    {
        //
        Tester tester;

        //
        Task Main() {
        }

        public async Task Main(Form browserWindow)
        {
            //
            tester = new Tester(browserWindow);

            await setUp();
            await test();
        }
    }
}

```

```

        await tearDown();
    }

    //
    public async Task setUp()
    {
        await tester.BrowserFullScreenAsync(); //

    }

    //
    public async Task test()
    {
        await tester.TestBeginAsync();
        await
tester.GoToUrlAsync("https://somovstudio.github.io/test.html", 5);
        string script = @"(function(){ var element =
document.getElementsByTagName('h2')[0]; return element.outerText; }());";
        string actual = await tester.ExecuteJavaScriptAsync(script);
        string expected = "          ";
        await tester.AssertEqualsAsync(expected, actual);
        await tester.TestEndAsync();
    }

    //
    public async Task tearDown()
    {
        // await tester.BrowserCloseAsync();
    }
}

```

"ExampleTest3.cs" "ExampleTest4.cs"  
"group1".

GroupTests1.cs

: GroupTests1.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
using System.IO;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Reflection;
using Newtonsoft.Json;
using HatFramework;

```

```

namespace Hat
{
    public class GroupTests1
    {
        //
        Tester tester;

        //
        (
        )
        public async void Main(Form browserWindow)
        {
            tester = new Tester(browserWindow); //

            await setUp(); //
            await test1(browserWindow); //
            ExampleTest3.cs
            await test2(browserWindow); //
            ExampleTest4.cs
            await tearDown(); //
        }

        public async Task setUp()
        {
            await tester.BrowserFullScreenAsync();
        }

        //
        ExampleTest3.cs
        public async Task test1(Form browserWindow)
        {
            tester.SendMessage("
                ExampleTest3", "", "
                :
            ExampleTest3.cs", Tester.IMAGE_STATUS_MESSAGE);
            ExampleTest3 exampleTest3 = new ExampleTest3(); //

            await exampleTest3.Main(browserWindow); //
        }

        //
        ExampleTest4.cs
        public async Task test2(Form browserWindow)
        {
            tester.SendMessage("
                ExampleTest4", "", "
                :
            ExampleTest4.cs", Tester.IMAGE_STATUS_MESSAGE);
            ExampleTest4 exampleTest4 = new ExampleTest4(); //

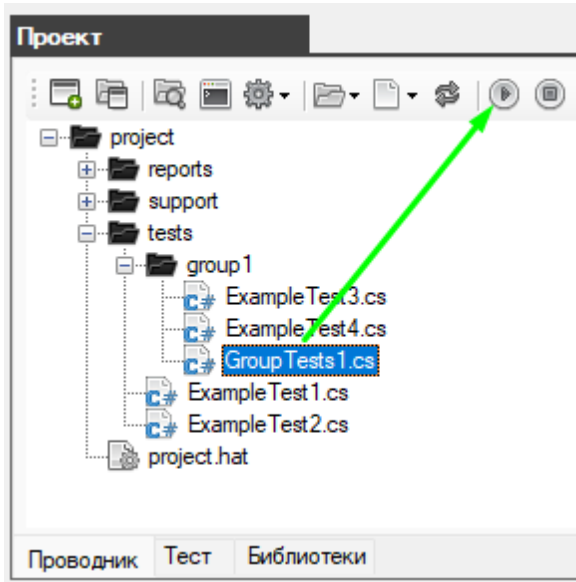
            await exampleTest4.Main(browserWindow); //
        }

        //
        public async Task tearDown()
        {
            await tester.BrowserCloseAsync(); //
        }
    }
}

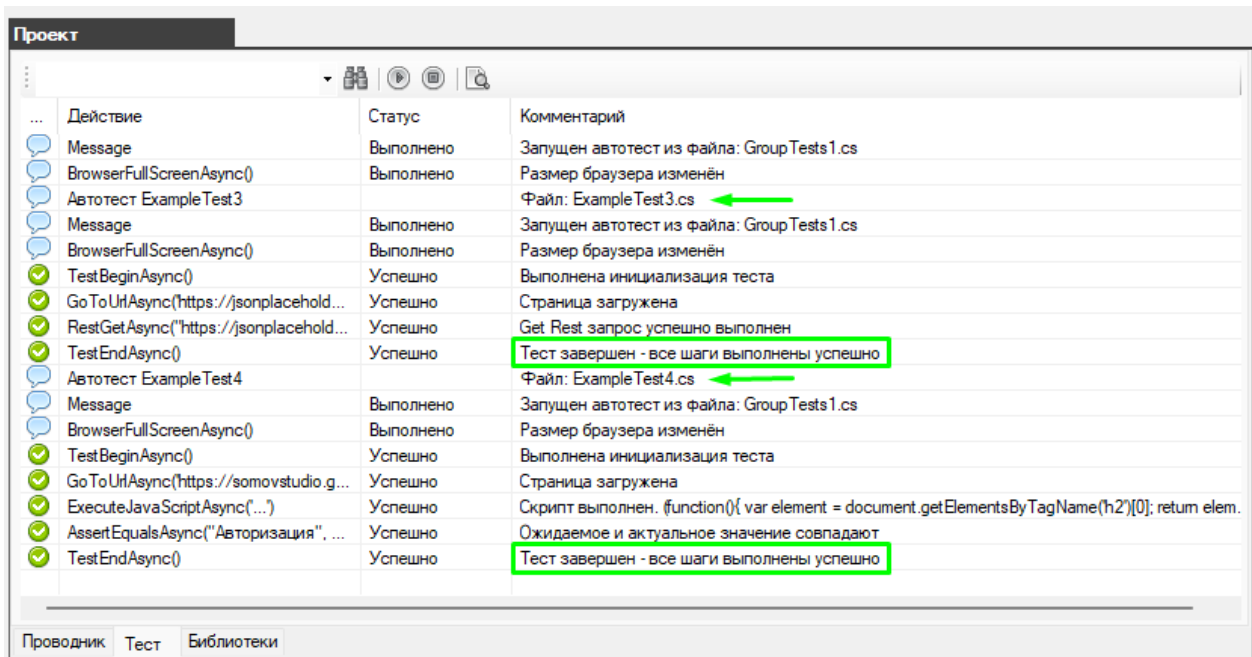
```

ExampleTest3.cs ExampleTest4.cs  
test1 test2.

GroupTests1.cs



" "



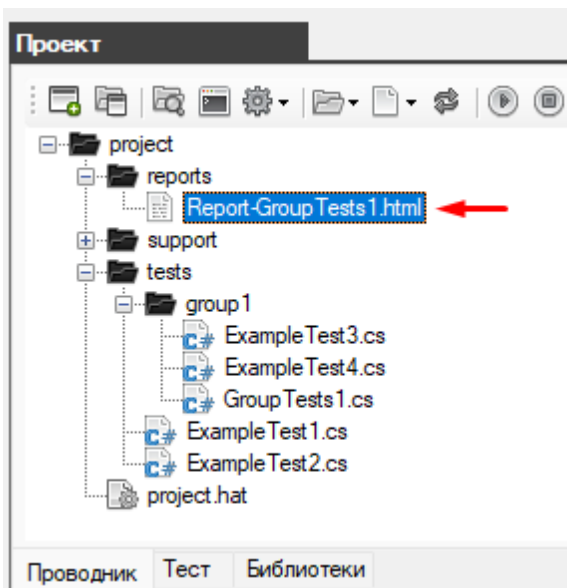
Действие	Статус	Комментарий
BrowserFullScreenAsync()	Выполнено	Размер браузера изменён
Автотест ExampleTest3		Файл: ExampleTest3.cs
Message	Выполнено	Запущен автотест из файла: GroupTests1.cs
BrowserFullScreenAsync()	Выполнено	Размер браузера изменён
TestBeginAsync()	Успешно	Выполнена инициализация теста
GoToUrlAsync(https://jsonplacehold...	Успешно	Страница загружена
RestGetAsync("https://jsonplacehold...	Успешно	Get Rest запрос успешно выполнен
AssertEqualsAsync(2, 1)	Провально	Ожидаемое и фактическое значение не совпадают
TestEndAsync()	Провально	Тест завершен - шаги теста выполнены неуспешно
Автотест ExampleTest4		Файл: ExampleTest4.cs
Message	Выполнено	Запущен автотест из файла: GroupTests1.cs
BrowserFullScreenAsync()	Выполнено	Размер браузера изменён
TestBeginAsync()	Успешно	Выполнена инициализация теста
GoToUrlAsync(https://somovstudio.g...	Успешно	Страница загружена
ExecuteJavaScriptAsync(...)	Успешно	Скрипт выполнен. (function(){ var element = document.getElementsByTagName(h2)[0]; return el
AssertEqualsAsync("Авторизация", ...)	Успешно	Ожидаемое и фактическое значение совпадают
TestEndAsync()	Успешно	Тест завершен - все шаги выполнены успешно

GroupTests1.cs

```

Действие: BrowserCloseAsync()
Статус: Выполняется
Комментарий: Браузер закрывается

=====
Tests ended. Finished: FAILURE
    
```



## Отчет о работе автотеста

Файл: GroupTests1.cs

Результат: Провально

Статус	Действие	Комментарий
Выполняется	AssertEqualsAsync(2, 1)	Проверка совпадения ожидаемого и фактического значения
Провально	AssertEqualsAsync(2, 1)	Ожидаемое и фактическое значение не совпадают
Выполняется	TestEndAsync()	Завершение теста
Провально	TestEndAsync()	Тест завершен - шаги теста выполнены неуспешно
Автотест ExampleTest4		Файл: ExampleTest4.cs
Выполняется	Message	Запуск автотеста
Выполнено	Message	Запущен автотест из файла: GroupTests1.cs
Выполняется	BrowserFullScreenAsync()	Изменяется размер браузера
Выполнено	BrowserFullScreenAsync()	Размер браузера изменен
Выполняется	TestBeginAsync()	Инициализация теста
Успешно	TestBeginAsync()	Выполнена инициализация теста
Выполняется	GoToUriAsync("https://somovstudio.github.io/test.html", 5)	Загрузка страницы
Успешно	GoToUriAsync("https://somovstudio.github.io/test.html", 5)	Страница загружена
Выполняется	ExecuteJavaScriptAsync(...)	Выполнение скрипта. (function(){ var element = document.getElementsByTagName("h2")[0]; return element.outerText; });
Успешно	ExecuteJavaScriptAsync(...)	Скрипт выполнен. (function(){ var element = document.getElementsByTagName("h2")[0]; return element.outerText; });
Выполняется	AssertEqualsAsync("Авторизация", "Авторизация")	Проверка совпадения ожидаемого и фактического значения
Успешно	AssertEqualsAsync("Авторизация", "Авторизация")	Ожидаемое и фактическое значение совпадают
Выполняется	TestEndAsync()	Завершение теста
Успешно	TestEndAsync()	Тест завершен - все шаги выполнены успешно
Ошибок:	2	

Created with the Personal Edition of HelpNDoc: [Free help authoring environment](#)

## Выполнение JavaScript кода

## JavaScript

JavaScript  
ExecuteJavaScriptAsync.

HatFramework

```
string script = @"(function(){
var element = document.getElementsByTagName('h2')[0];
return element.outerText;
})();";
```

## ExecuteJavaScriptAsync

```
string actual = await tester.ExecuteJavaScriptAsync(script);
```

H2

: ExampleTest4.cs

using System;

```

using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
using System.IO;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Reflection;
using Newtonsoft.Json;
using HatFramework;

namespace Hat
{
    public class ExampleTest4
    {
        Tester tester; //

        //          (
        //          )
        public async void Main(Form browserWindow)
        {
            tester = new Tester(browserWindow); //

            await setUp(); //
            await test(); //
            await tearDown(); //
        }

        public async Task setUp()
        {
            await tester.BrowserFullScreenAsync(); //
        }

        public async Task test()
        {
            await tester.TestBeginAsync(); //
            await
tester.GoToUrlAsync("https://somovstudio.github.io/test.html", 5); //

            //          JavaScript
            string script = @"(function(){
                var element = document.getElementsByTagName('h2')[0];
                return element.outerText;
            });";

            //
            string actual = await tester.ExecuteJavaScriptAsync(script);

            //

```

```

        string expected = "          ";
        await tester.AssertEqualsAsync(expected, actual); //

        await tester.TestEndAsync(); //
    }

    public async Task tearDown()
    {
        await tester.BrowserCloseAsync(); //
    }
}

```

## JavaScript JQuery

## ExecuteJavaScriptAsync :

## - input

```

var element = document.evaluate("//input[@id='phone']", document, null,
XPathResult.FIRST_ORDERED_NODE_TYPE, null).singleNodeValue;
element.focus();
element.value = '9999999999';
element.setAttribute("value", "9999999999");

```

## - input

```

document.getElementById("phone").focus();
document.execCommand("insertHTML", false, "9999999999");

```

## - input

```

var element = document.evaluate("//input[@id='phone']", document, null,
XPathResult.FIRST_ORDERED_NODE_TYPE, null).singleNodeValue;
element.value = '9999999999';
element.dispatchEvent(new KeyboardEvent('keydown', { bubbles: true }));
element.dispatchEvent(new KeyboardEvent('keypress', { bubbles: true }));
element.dispatchEvent(new KeyboardEvent('keyup', { bubbles: true }));
element.dispatchEvent(new Event('input', { bubbles: true }));
element.dispatchEvent(new Event('change', { bubbles: true }));

```

## - input

```

var element = document.evaluate("//input[@id='phone']", document, null,
XPathResult.FIRST_ORDERED_NODE_TYPE, null).singleNodeValue;
element.select();

```

```

var clickEvent = new MouseEvent("click", { "view": window, "bubbles": true,
"cancelable": false });
var element = document.evaluate("//input[@id='phone']", document, null,
XPathResult.FIRST_ORDERED_NODE_TYPE, null).singleNodeValue;
element.dispatchEvent(clickEvent);

```

```

var theEvent = document.createEvent("MouseEvent");
theEvent.initMouseEvent("click", true, true, window, 0, 0, 0, 0, 0, false,
false, false, false, 0, null);

```

```
element.dispatchEvent(theEvent);
```

```
-      JQuery      (
      )
```

```
string script = @"$(function() {
console.log('autotest JQUERY');
$('#btn-send > div.btn > a').click();
});";
```

---

Created with the Personal Edition of HelpNDoc: [Easy CHM and documentation editor](#)

---

## Обработка Json данных с помощью Newtonsoft

Json

Newtonsoft

Json

Newtonsoft.Json

: <https://www.newtonsoft.com/json>

Rest

<https://jsonplaceholder.typicode.com/posts/1/>

Json

:

```
{ "userId": 1, "id": 1, "title": "sunt aut facere repellat provident occaecati
excepturi optio reprehenderit", "body": "quia et suscipit\nsuscipit recusandae
consequuntur expedita et cum\nreprehenderit molestiae ut ut quas
totam\nnostrum rerum est autem sunt rem eveniet architecto" }
```

Newtonsoft.Json

Json-

```
public class TestJson
{
    public int userId { get; set; }
    public int id { get; set; }
    public string title { get; set; }
    public string body { get; set; }
}
```

Rest

Json

```
string result = await
tester.RestGetAsync(@"https://jsonplaceholder.typicode.com/posts/1/",
    TimeSpan.FromDays(1), "UTF-8");
```

```
TestJson dataJson = JsonConvert.DeserializeObject<TestJson>(result);
tester.ConsoleMsg("UserID: " + dataJson.userId.ToString());
```

:

: ExampleTest3.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
using System.IO;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Reflection;
using Newtonsoft.Json;
using HatFramework;

namespace Hat
{
    //                JSON
    public class TestJson
    {
        public int userId { get; set; }
        public int id { get; set; }
        public string title { get; set; }
        public string body { get; set; }
    }

    public class ExampleTest3
    {
        Tester tester; //

        //                (
        //                )
        public async void Main(Form browserWindow)
        {
            tester = new Tester(browserWindow); //

            await setUp(); //
            await test(); //
            await tearDown(); //
        }

        public async Task setUp()
        {
            await tester.BrowserFullScreenAsync(); //
        }

        public async Task test()
        {
            await tester.TestBeginAsync(); //
            await
tester.GoToUrlAsync("https://jsonplaceholder.typicode.com", 5); //

```

```

//          Rest          API          URL
//          JSON
string result = await
tester.RestGetAsync(@"https://jsonplaceholder.typicode.com/posts/1/",
TimeSpan.FromDays(1), "UTF-8");
tester.ConsoleMsg(result); //

//          JSON
TestJson dataJson =
JsonConvert.DeserializeObject<TestJson>(result);

//
tester.ConsoleMsg("UserID: " + dataJson.userId.ToString());
tester.ConsoleMsg("ID: " + dataJson.id.ToString());
tester.ConsoleMsg("Title: " + dataJson.title.ToString());
tester.ConsoleMsg("Body: " + dataJson.body.ToString());

await tester.TestEndAsync(); //
}

public async Task tearDown()
{
await tester.BrowserCloseAsync(); //
}
}
}

```

Created with the Personal Edition of HelpNDoc: [News and information about help authoring tools and software](#)

## Обработка XML данных для проверки карты сайта Sitemap

**XML**

**Sitemap**

XML

```

using System.Collections;
using System.Xml;

```

ReadSitemapXML

sitemap.xml

Steps

Helper.

ReadSitemapXML

sitemap.xml

: Helper.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
using System.IO;

```

```

using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Reflection;
using Newtonsoft.Json;
using HatFramework;
using System.Collections;
using System.Xml;

namespace Hat
{
    public static class Helper
    {
        /*      :      XML      */
        public static async Task<List<string>> ReadSitemapXML(Tester tester,
string filename, bool useragent)
        {
            //
            int step = tester.SendMessage("ReadSitemapXML(\"" + filename +
"\")", Tester.PROCESS, "      xml      ",
Tester.IMAGE_STATUS_PROCESS);

            //      (

        )
            List<string> list = new List<string>();

            //      (

        )
            bool errors = false;

            try
            {
                //

                ServicePointManager.Expect100Continue = true;
                ServicePointManager.SecurityProtocol =
SecurityProtocolType.Tls12;
                ServicePointManager.ServerCertificateValidationCallback =
delegate { return true; };

                //      xml
                XmlDocument xDoc;
                if (useragent == false)
                {
                    xDoc = new XmlDocument();
                    xDoc.Load(filename);
                }
                else
                {
                    WebClient client = new WebClient();
                    client.Headers["User-Agent"] = await
tester.BrowserGetUserAgentAsync();
                    client.Headers["Accept"] =
"text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8";

```

```

        string data = client.DownloadString(filename);

        xDoc = new XmlDocument();
        xDoc.LoadXml(data);
    }

    //                                     xml
    XmlElement xRoot = xDoc.DocumentElement;
    foreach (XmlNode xnode in xRoot)
    {
        for (int j = 0; j <= xnode.ChildNodes.Count; j++)
        {
            if (xnode.ChildNodes[j].Name == "loc")
            {
                //
                list.Add(xnode.ChildNodes[j].InnerText);
                break;
            }
        }
    }

    //
    tester.EditMessage(step, null, Tester.PASSED, "                                     xml
    ", Tester.IMAGE_STATUS_PASSED);
}
catch (Exception ex)
{
    //
    tester.EditMessage(step, null, Tester.FAILED, "
    xml                                     : " + ex.Message,
    Tester.IMAGE_STATUS_FAILED);
    errors = true; //
}

//                                     (                                     true

await tester.AssertFalseAsync(errors);

return list; //
}
}
}

```

: Sitemap\_Test.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
using System.IO;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

```

```

using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Reflection;
using Newtonsoft.Json;
using HatFramework;

namespace Hat
{
    public class Sitemap_Test
    {
        //
        Tester tester;

        //
        (
        )
        public async void Main(Form browserWindow)
        {
            //
            tester = new Tester(browserWindow);
            await setUp();
            await test();
            await tearDown();
        }

        //
        public async Task setUp()
        {
            await tester.BrowserFullScreenAsync(); //
        }

        //
        public async Task test()
        {
            await tester.TestBeginAsync(); //

            //
            tester.SendMessage("Message", Tester.COMPLETED, "
                ", Tester.IMAGE_STATUS_MESSAGE);

            //
            await
            tester.GoToUrlAsync("https://somovstudio.github.io/sitemap.xml", 25);

            //
            sitemap.xml
            List<string> list = await Helper.ReadSitemapXML(tester,
            "https://somovstudio.github.io/sitemap.xml", false);

            //
            int status = 0;
            foreach (string url in list)
            {
                //
                (200 -
                , 404 -
                , 502 -
                . .)
                status = await tester.RestGetStatusCodeAsync(url);
                //
                tester.ConsoleMsg("
                : " + status.ToString() + " |
                : " + url);
            }
        }
    }
}

```

```

        //
        await tester.AssertEqualsAsync(200, status);
    }

    await tester.TestEndAsync(); //
}

//
public async Task tearDown()
{
    await tester.BrowserCloseAsync(); //
}
}
}

```

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

## Диалоговые окна alert, prompt и confirm

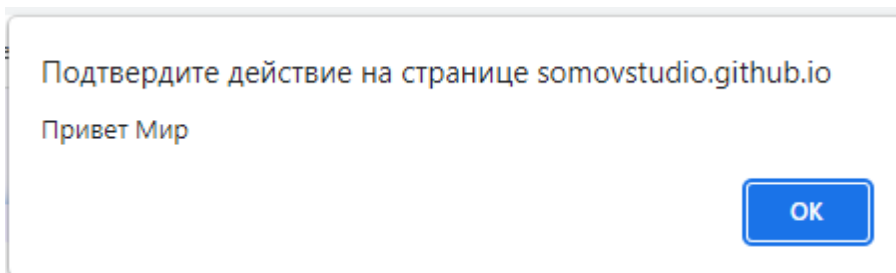
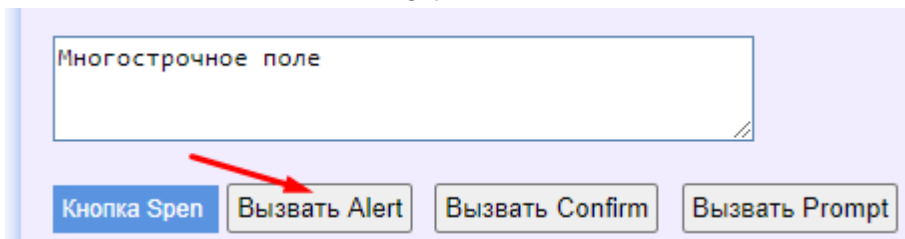
**alert, prompt confirm**

alert, prompt confirm.

?


<https://somovstudio.github.io/test2.html>

" Alert"



"

"

Привет Мир 

Кнопка Spen Вызвать Alert Вызвать Confirm Вызвать Prompt

"  
Alert."  
" "  
" (" Confirm"  
" ")

OK

Подтвердите действие на странице somovstudio.github.io  
Вопрос: вы согласны?

OK Отмена

Да (спасибо за ответ)

Кнопка Spen Вызвать Alert Вызвать Confirm Вызвать Prompt

" Confirm"  
" "

Подтвердите действие на странице somovstudio.github.io  
Какой сейчас год?

2022

OK Отмена

Сейчас 2022 год

Кнопка Spen Вызвать Alert Вызвать Confirm Вызвать Prompt

: ExampleTest5.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
using System.IO;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Reflection;
using Newtonsoft.Json;
using HatFramework;

namespace Hat
{
    public class ExampleTest5
    {
        Tester tester; //

        //
        (
        )
        public async void Main(Form browserWindow)
        {
            tester = new Tester(browserWindow); //

            await setUp(); //
            await test(); //
            await tearDown(); //
        }

        public async Task setUp()
        {
            await tester.BrowserFullScreenAsync(); //
        }

        public async Task test()
        {
            await tester.TestBeginAsync(); //
            await
tester.GoToUrlAsync("https://somovstudio.github.io/test2.html", 5); //

            await tester.WaitAsync(2); // 2
            await tester.ClickElementByIdAsync("btnAlert"); //

            await tester.WaitAsync(2); // 2
            await tester.ClickElementByIdAsync("btnConfirm"); //

            await tester.WaitAsync(2); // 2
            await tester.ClickElementByIdAsync("btnPrompt"); //

```

```

        await tester.TestEndAsync(); //
    }

    public async Task tearDown()
    {
        await tester.BrowserCloseAsync(); //
    }
}

```

Многострочное поле

Кнопка Spen Вызвать Alert Вызвать Confirm Вызвать Prompt

" Alert", \_\_\_\_\_

Привет Мир

Кнопка Spen Вызвать Alert Вызвать Confirm Вызвать Prompt

" Confirm", \_\_\_\_\_

Нет (спасибо за ответ)

Кнопка Spen Вызвать Alert Вызвать Confirm Вызвать Prompt

" Prompt", \_\_\_\_\_

Сейчас null год

Кнопка Spen Вызвать Alert Вызвать Confirm Вызвать Prompt

## Базовая авторизация

**Войдите в систему, чтобы получить доступ к этому сайту**

Требуется авторизация для <http://test.site.com>  
Подключение к этому сайту не защищено.

Имя пользователя

Пароль

- [BrowserBasicAuthenticationAsync](#) -
- [GoToUrlBaseAuthAsync](#) -

URL

### BrowserBasicAuthenticationAsync

```
: ExampleTest6.cs
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
using System.IO;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Reflection;
using Newtonsoft.Json;
using HatFramework;

namespace Hat
{
```

```

public class ExampleTest5
{
    Tester tester; //

    //          (
    )
    public async void Main(Form browserWindow)
    {
        tester = new Tester(browserWindow); //

        await setUp(); //
        await test(); //
        await tearDown(); //
    }

    public async Task setUp()
    {
        await tester.BrowserFullScreenAsync(); //
    }

    public async Task test()
    {
        await tester.TestBeginAsync(); //
        await tester.BrowserBasicAuthenticationAsync("login",
"pass"); //
        await tester.GoToUrlAsync("https://test.site.com", 5); //

        await tester.TestEndAsync(); //
    }

    public async Task tearDown()
    {
        await tester.BrowserCloseAsync(); //
    }
}
}

```

### GoToUrlBaseAuthAsync

: ExampleTest6.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
using System.IO;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Reflection;

```

```

using Newtonsoft.Json;
using HatFramework;

namespace Hat
{
    public class ExampleTest5
    {
        Tester tester; //

        //          (
        //          )
        public async void Main(Form browserWindow)
        {
            tester = new Tester(browserWindow); //

            await setUp(); //
            await test(); //
            await tearDown(); //
        }

        public async Task setUp()
        {
            await tester.BrowserFullScreenAsync(); //
        }

        public async Task test()
        {
            await tester.TestBeginAsync(); //

            //
            await tester.GoToUrlBaseAuthAsync("https://test.site.com",
"login", "pass", 5);

            await tester.TestEndAsync(); //
        }

        public async Task tearDown()
        {
            await tester.BrowserCloseAsync(); //
        }
    }
}

```

## Перехват событий google analytics и yandex metrika

google analytics yandex metrika

GA YM

```

await tester.AssertNetworkEventsAsync(true, new string[] {
    "ec=ga_category", "ea=ga_action", "el=ga_label", "yandex_event"
});
await Helper.ClearBrowserNetworkLogs(tester); //

```

: ExampleTest7.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
using System.IO;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Reflection;
using Newtonsoft.Json;
using HatFramework;

namespace Hat
{
    public class ExampleTest5
    {
        Tester tester; //
        List<string> events; //

        //          (
        //          )
        public async void Main(Form browserWindow)
        {
            tester = new Tester(browserWindow); //

            await setUp(); //
            await test(); //
            await tearDown(); //
        }

        //
        private void browserConsoleEvents(object sender,
Microsoft.Web.WebView2.Core.CoreWebView2DevToolsProtocolEventReceivedEventAr
gs e)
        {
            events.Add(e.ParameterObjectAsJson);
            tester.ConsoleMsg(e.ParameterObjectAsJson);
        }

        public async Task setUp()
        {
            await tester.BrowserFullScreenAsync(); //

```

```

        /*
        *
        *           : https://chromedevtools.github.io/devtools-
protocol/tot/Network/
        */
        tester.BrowserView.CoreWebView2.GetDevToolsProtocolEventReceiver
("Log.entryAdded").DevToolsProtocolEventReceived += browserConsoleEvents;
        await
tester.BrowserView.CoreWebView2.CallDevToolsProtocolMethodAsync
("Log.enable", "{}");
        tester.BrowserView.CoreWebView2.GetDevToolsProtocolEventReceiver
("Console.messageAdded").DevToolsProtocolEventReceived +=
browserConsoleEvents;
        await
tester.BrowserView.CoreWebView2.CallDevToolsProtocolMethodAsync
("Console.enable", "{}");
        tester.BrowserView.CoreWebView2.GetDevToolsProtocolEventReceiver
("Runtime.consoleAPICalled").DevToolsProtocolEventReceived +=
browserConsoleEvents;
        tester.BrowserView.CoreWebView2.GetDevToolsProtocolEventReceiver
("Runtime.exceptionThrown").DevToolsProtocolEventReceived +=
browserConsoleEvents;
        await
tester.BrowserView.CoreWebView2.CallDevToolsProtocolMethodAsync
("Runtime.enable", "{}");
tester.BrowserView.CoreWebView2.GetDevToolsProtocolEventReceiver
("Network.dataReceived").DevToolsProtocolEventReceived +=
browserConsoleEvents;
        await
tester.BrowserView.CoreWebView2.CallDevToolsProtocolMethodAsync
("Network.enable", "{}");
        tester.BrowserView.CoreWebView2.GetDevToolsProtocolEventReceiver
("Debugger.scriptParsed").DevToolsProtocolEventReceived +=
browserConsoleEvents;
        await
tester.BrowserView.CoreWebView2.CallDevToolsProtocolMethodAsync
("Debugger.enable", "{}");
    }

    public async Task test()
    {
        await tester.TestBeginAsync(); //
        await tester.GoToUrlAsync("https://test.site.com/?_ym_debug=1",
5); //
                                YM

        await tester.SetValueInElementAsync(Tester.BY_XPATH,
"//input[@id='SECOND_NAME']", Helper.TestName);
        await tester.WaitAsync(1);
        await tester.SetValueInElementAsync(Tester.BY_XPATH,
"//input[@id='NAME']", Helper.TestName);
        await tester.WaitAsync(1);
        await tester.SetValueInElementAsync(Tester.BY_XPATH,
"//input[@id='PHONE']", Helper.TestPhone);
        await tester.WaitAsync(1);
        await tester.ClickElementAsync(Tester.BY_XPATH,
"//input[@id='fb_close']");
        await tester.WaitAsync(1);
        await tester.WaitVisibleElementAsync(Tester.BY_XPATH,

```

```

"//p[text() = '      !      .']", 25);
    await tester.WaitAsync(5);

    //      YM
    tester.SendMessage("Message", Tester.COMPLETED, "      :
YM (Goal id: yandex_event)", Tester.IMAGE_STATUS_MESSAGE);
    bool checkYM = false;
    foreach(string value in events)
    {
        checkYM = value.Contains("Goal id: yandex_event");
        if (checkYM == true)
        {
            tester.SendMessage("Message", Tester.COMPLETED,
"      YM: " + value, Tester.IMAGE_STATUS_MESSAGE);
            break;
        }
    }
    await tester.AssertTrueAsync(checkYM); //

    await tester.TestEndAsync(); //
}

public async Task tearDown()
{
    await tester.BrowserCloseAsync(); //
}

}
}

```

YM

YM.

---

Created with the Personal Edition of HelpNDoc: [Easily create Help documents](#)

---

## Версия 1.3 (не актуальна)

---

Created with the Personal Edition of HelpNDoc: [Easily create Qt Help files](#)

---

### Описание

1.3

:

- SendMessage -
- SendMessageDebug -
- EditMessage -
- EditMessageDebug -
- DefineTestStop -

## SendMessage [изменён начиная с версии 1.3.0]

**SendMessage** (старая версия метода до обновления 1.3.0)

**Описание:** метод выводит сообщения в таблице процесса выполнения теста  
При выполнении в системную консоль выводится action если status равен FAILED, WARNING, "", null,  
в противном случае выводится только status и comment (без action).

**Синтаксис:** SendMessage(string action, string status, string comment, int image)

**Пример:**

```
int step = tester.SendMessage("текст действия", Tester.PROCESS, "текст комментария",  
Tester.IMAGE_STATUS_PROCESS);
```

```
/* Примеры вывода в консоль */
```

```
// действие комментариев
```

```
int step = tester.SendMessage("действия", null, "комментарий",  
Tester.IMAGE_STATUS_MESSAGE);
```

```
int step = tester.SendMessage("действия", "", "комментарий",  
Tester.IMAGE_STATUS_MESSAGE);
```

```
// Шаг[в процессе]: комментарий
```

```
int step = tester.SendMessage("действия", Tester.PROCESS, "комментарий",  
Tester.IMAGE_STATUS_PROCESS);
```

```
// Шаг[успешно]: комментарий
```

```
int step = tester.SendMessage("действия", Tester.PASSED, "комментарий",  
Tester.IMAGE_STATUS_PASSED);
```

```
// Шаг[выполнено]: комментарий
```

```
int step = tester.SendMessage("действия", Tester.COMPLETED, "комментарий",  
Tester.IMAGE_STATUS_MESSAGE);
```

```
// Шаг[остановлено]: комментарий
```

```
int step = tester.SendMessage("действия", Tester.STOPPED, "комментарий",  
Tester.IMAGE_STATUS_WARNING);
```

```
// Шаг[неудача]: действие комментариев
```

```
int step = tester.SendMessage("действия", Tester.FAILED, "комментарий",  
Tester.IMAGE_STATUS_FAILED);
```

```
// Шаг[предупреждение]: действие комментариев
```

```
int step = tester.SendMessage("действия", Tester.WARNING, "комментарий",  
Tester.IMAGE_STATUS_WARNING);
```

## SendMessageDebug [изменён начиная с версии 1.3.0]

SendMessageDebug (старая версия метода до обновления 1.3.0)

**Описание:** метод выводит отладочное сообщение которое может быть отключено для вывода в отчет и письмо

При выполнении в системную консоль выводится action если status равен FAILED, WARNING, "", null, в противном случае выводится только status и comment (без action).

**Синтаксис:** SendMessageDebug(string actionRus, string actionEng, string status, string commentRus, string commentEng, int image)

**Пример:**

```
int step = tester.SendMessageDebug("действия", "action", Tester.PROCESS, "комментарий", "comment", Tester.IMAGE_STATUS_PROCESS);
```

```
/* Примеры вывода в консоль */
```

```
// действие комментарий
```

```
int step = tester.SendMessageDebug("действия", "action", null, "комментарий", "comment", Tester.IMAGE_STATUS_MESSAGE);
```

```
int step = tester.SendMessageDebug("действия", "action", "", "комментарий", "comment", Tester.IMAGE_STATUS_MESSAGE);
```

```
// Шаг[в процессе]: комментарий
```

```
int step = tester.SendMessageDebug("действия", "action", Tester.PROCESS, "комментарий", "comment", Tester.IMAGE_STATUS_PROCESS);
```

```
// Шаг[успешно]: комментарий
```

```
int step = tester.SendMessageDebug("действия", "action", Tester.PASSED, "комментарий", "comment", Tester.IMAGE_STATUS_PASSED);
```

```
// Шаг[выполнено]: комментарий
```

```
int step = tester.SendMessageDebug("действия", "action", Tester.COMPLETED, "комментарий", "comment", Tester.IMAGE_STATUS_MESSAGE);
```

```
// Шаг[остановлено]: комментарий
```

```
int step = tester.SendMessageDebug("действия", "action", Tester.STOPPED, "комментарий", "comment", Tester.IMAGE_STATUS_WARNING);
```

```
// Шаг[неудача]: действие комментарий
```

```
int step = tester.SendMessageDebug("действия", "action", Tester.FAILED, "комментарий", "comment", Tester.IMAGE_STATUS_FAILED);
```

```
// Шаг[предупреждение]: действие комментарий
```

```
int step = tester.SendMessageDebug("действия", "action", Tester.WARNING, "комментарий", "comment", Tester.IMAGE_STATUS_WARNING);
```

## EditMessage [удалён начиная с версии 1.3.0]

`EditMessage` (старый метода, начиная с версии 1.3.0 был удален)

**Описание:** метод изменяет ранее выведенное сообщения в таблице процесса выполнения теста

При выполнении в системную консоль выводится action если status равен FAILED, WARNING, "" , null,

в противном случае выводится только status и comment (без action).

Так же action можно указывать как null если он был указан в `SendMessage`.

**Синтаксис:** `EditMessage(int index, string action, string status, string comment, int image)`

**Пример:**

```
tester.EditMessage(step, "текст действия", Tester.PASSED, "текст комментария",
Tester.IMAGE_STATUS_PASSED);
```

```
tester.EditMessage(step, null, Tester.FAILED, "текст комментария",
Tester.IMAGE_STATUS_FAILED);
```

```
/* Примеры вывода в консоль */
```

```
// действие комментарий
```

```
tester.EditMessage(step, "действия", null, "комментарий", Tester.IMAGE_STATUS_MESSAGE);
```

```
tester.EditMessage(step, "действия", "", "комментарий", Tester.IMAGE_STATUS_MESSAGE);
```

```
// Шаг[в процессе]: комментарий
```

```
tester.EditMessage(step, "действия", Tester.PROCESS, "комментарий",
Tester.IMAGE_STATUS_PROCESS);
```

```
// Шаг[успешно]: комментарий
```

```
tester.EditMessage(step, "действия", Tester.PASSED, "комментарий",
Tester.IMAGE_STATUS_PASSED);
```

```
// Шаг[выполнено]: комментарий
```

```
tester.EditMessage(step, "действия", Tester.COMPLETED, "комментарий",
Tester.IMAGE_STATUS_MESSAGE);
```

```
// Шаг[остановлено]: комментарий
```

```
tester.EditMessage(step, "действия", Tester.STOPPED, "комментарий",
Tester.IMAGE_STATUS_WARNING);
```

```
// Шаг[неудача]: действие комментарий
```

```
tester.EditMessage(step, "действия", Tester.FAILED, "комментарий",
Tester.IMAGE_STATUS_FAILED);
```

```
// Шаг[предупреждение]: действие комментарий
```

```
tester.EditMessage(step, "действия", Tester.WARNING, "комментарий",
Tester.IMAGE_STATUS_WARNING);
```

## EditMessageDebug [удалён начиная с версии 1.3.0]

`EditMessageDebug` (старый метода, начиная с версии 1.3.0 был удален)

**Описание:** метод изменяет ранее выведенное отладочное сообщение которое может быть отключено для вывода в отчет и письмо

При выполнении в системную консоль выводится action если status равен FAILED, WARNING, "", null,

в противном случае выводится только status и comment (без action)

Так же action можно указывать как null если он был указан в `SendMessageDebug`.

**Синтаксис:** `EditMessageDebug(int index, string actionRus, string actionEng, string status, string commentRus, string commentEng, int image)`

**Пример:**

```
tester.EditMessageDebug(step, "действия", "action", Tester.PASSED, "комментарий",
"comment", Tester.IMAGE_STATUS_PASSED);
tester.EditMessageDebug(step, null, null, Tester.FAILED, "комментарий", "comment",
Tester.IMAGE_STATUS_FAILED); // если действие не меняется можно ставить null
```

*/\* Примеры вывода в консоль \*/*

*// действие комментарий*

```
tester.EditMessageDebug(step, "действия", "action", null, "комментарий", "comment",
Tester.IMAGE_STATUS_MESSAGE);
tester.EditMessageDebug(step, "действия", "action", "", "комментарий", "comment",
Tester.IMAGE_STATUS_MESSAGE);
```

*// Шаг[в процессе]: комментарий*

```
tester.EditMessageDebug(step, "действия", "action", Tester.PROCESS, "комментарий",
"comment", Tester.IMAGE_STATUS_PROCESS);
```

*// Шаг[успешно]: комментарий*

```
tester.EditMessageDebug(step, "действия", "action", Tester.PASSED, "комментарий",
"comment", Tester.IMAGE_STATUS_PASSED);
```

*// Шаг[выполнено]: комментарий*

```
tester.EditMessageDebug(step, "действия", "action", Tester.COMPLETED, "комментарий",
"comment", Tester.IMAGE_STATUS_MESSAGE);
```

*// Шаг[остановлено]: комментарий*

```
tester.EditMessageDebug(step, "действия", "action", Tester.STOPPED, "комментарий",
"comment", Tester.IMAGE_STATUS_WARNING);
```

*// Шаг[неудача]: действие комментарий*

```
tester.EditMessageDebug(step, "действия", "action", Tester.FAILED, "комментарий",
"comment", Tester.IMAGE_STATUS_FAILED);
```

*// Шаг[предупреждение]: действие комментарий*

```
tester.EditMessageDebug(step, "действия", "action", Tester.WARNING, "комментарий",
```

```
"comment", Tester.IMAGE_STATUS_WARNING);
```

---

Created with the Personal Edition of HelpNDoc: [Full-featured EPub generator](#)

---

### **DefineTestStop [изменён начиная с версии 1.3.0]**

**DefineTestStop** (старая версия метода до обновления 1.3.0)

**Описание:** метод проверяет статус процесса (остановлен или нет)

**Синтаксис:** DefineTestStop(int stepIndex)

**Пример:**

```
int step = tester.SendMessage("текст действия", Tester.PROCESS, "текст комментария",  
Tester.IMAGE_STATUS_PROCESS);  
if (tester.DefineTestStop(step) == true) return;
```

---

Created with the Personal Edition of HelpNDoc: [iPhone web sites made easy](#)

---