HatPluginMySql

Copyright © 2026 by Somov Studio. All Rights Reserved.

Table of contents

English version

Created with the Personal Edition of HelpNDoc: Easily create Help documents

Description

HatPluginMySql

HatPluginMySql - a plugin for the Hat browser that allows you to perform automated testing of data in a MySQL database.

System requirements:
the plugin is compatible with the Hat browser starting from version 1.4

The official page: https://somovstudio.github.io/

Download the app from GitHub: https://github.com/SomovStudio/Hat

The version of this documentation corresponds to the browser version 1.5.0.3 (update 03/27/2026)

Created with the Personal Edition of HelpNDoc: Produce online help for Qt applications

Plugins

Created with the Personal Edition of HelpNDoc: Full-featured Help generator

HatPluginMySql

Created with the Personal Edition of HelpNDoc: Easily create HTML Help documents

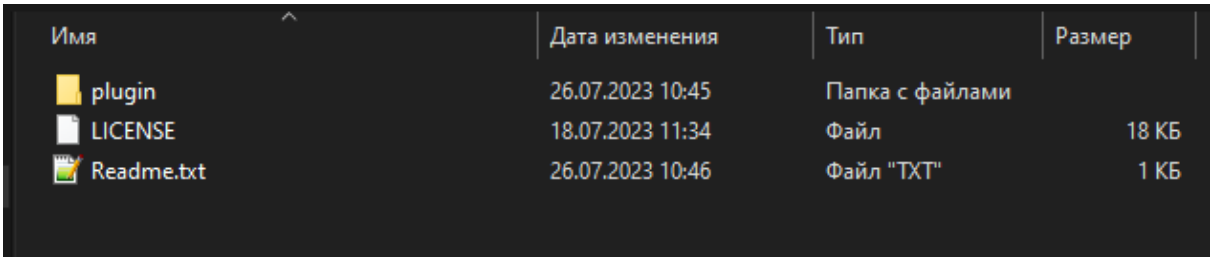
Installing the plugin

Installing the plugin HatPluginMySql

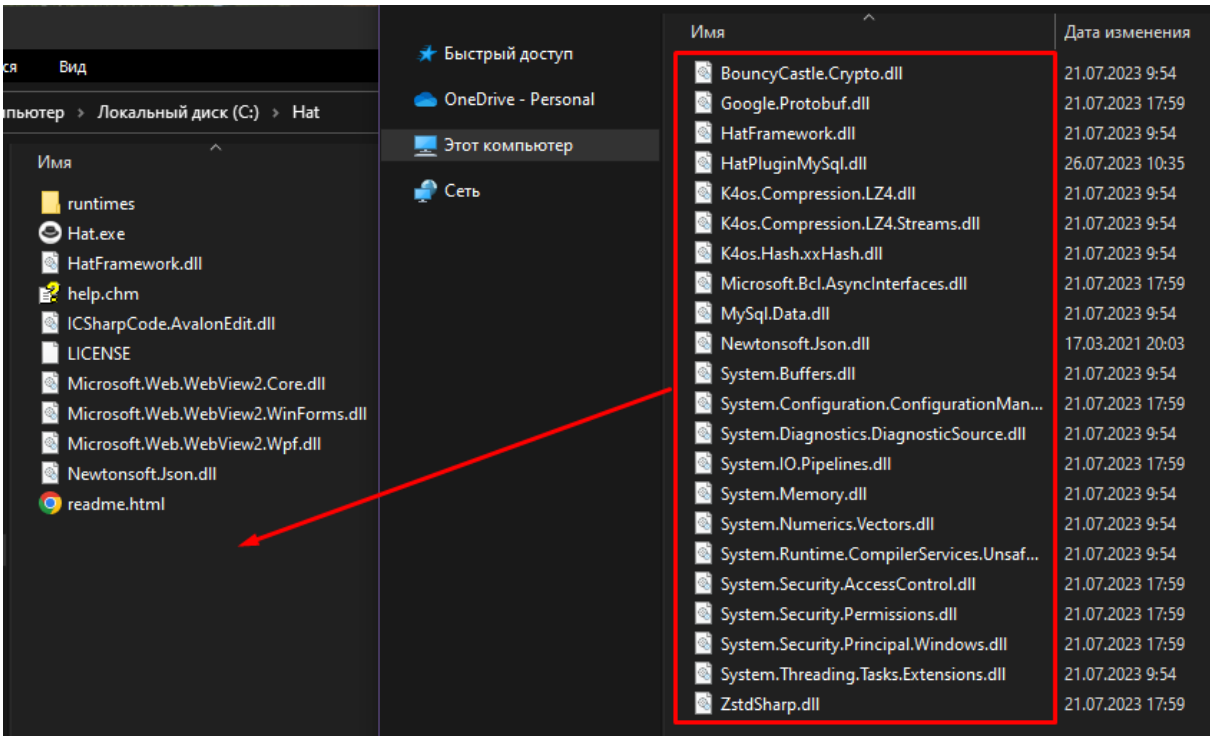
1. Download and install the browser **Hat**
link: https://github.com/SomovStudio/Hat/releases

2. Download the plugin **HatPluginMySql**
link: https://github.com/SomovStudio/HatPluginMySql/releases

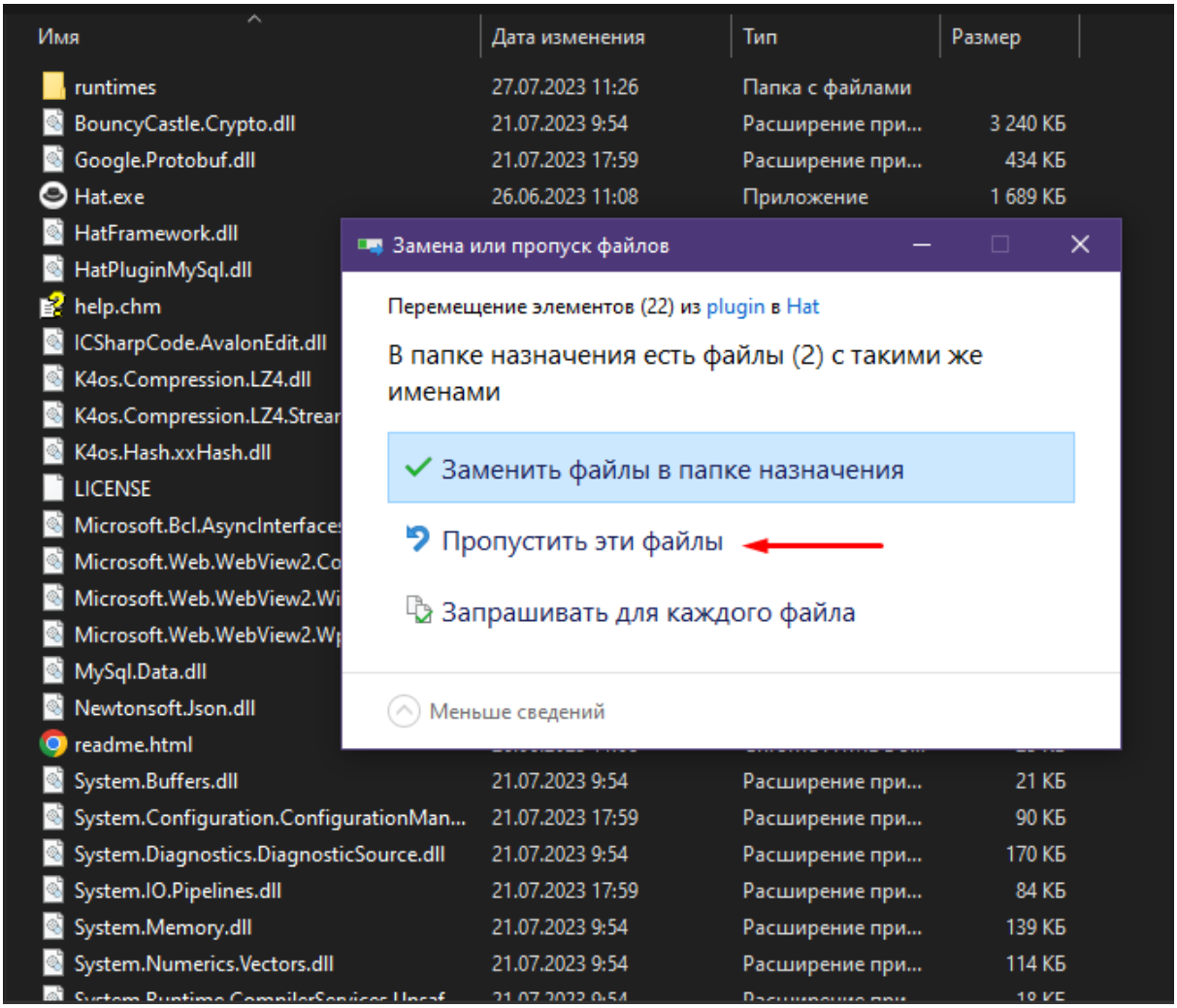
3. Unpack the archive **HatPluginMySql-1.0.0.zip**



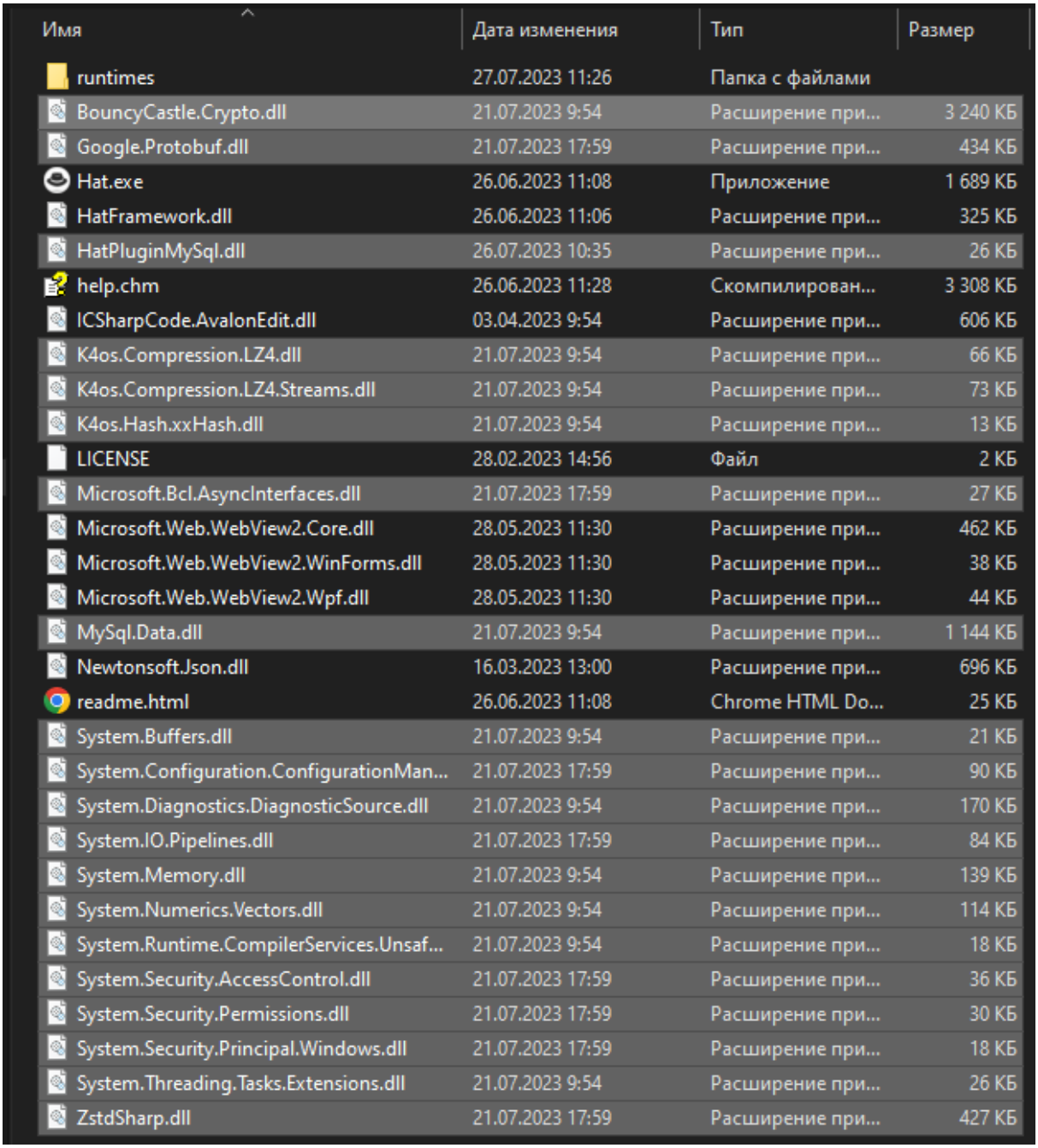
4. Copy the contents of the **plugin** folder to the root of the **Hat** browser folder



When copying, you may be asked about replacing existing files.
In this case, it is recommended to select "Skip these files".



As a result, the following files should be in the root of the folder



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

This completes the installation of the plugin!

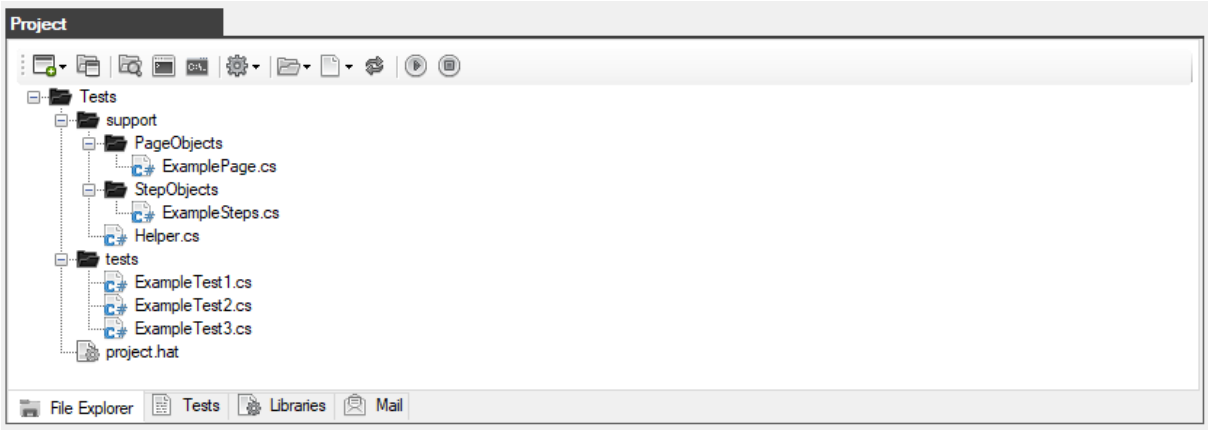
Created with the Personal Edition of HelpNDoc: Easily create CHM Help documents

Connecting the plugin to the project

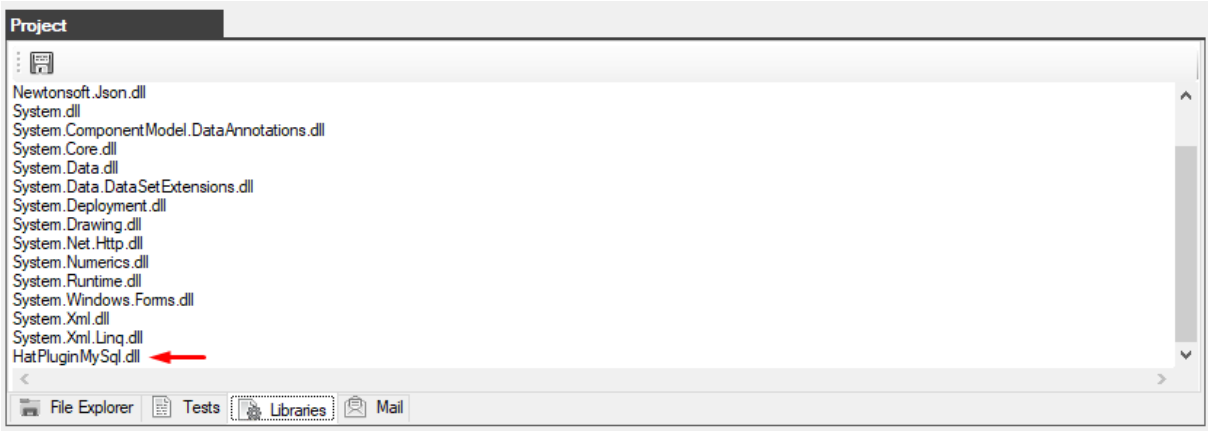
Connecting the HatPluginMySql plugin to the autotest project

1. Launch the browser **Hat**

2. Create a new project or open an existing project



3. Go to the Libraries tab and enter the name of one file at the end of the list **HatPluginMySql.dll**



4. Click the "Save" button.

This completes the plug-in connection to the project

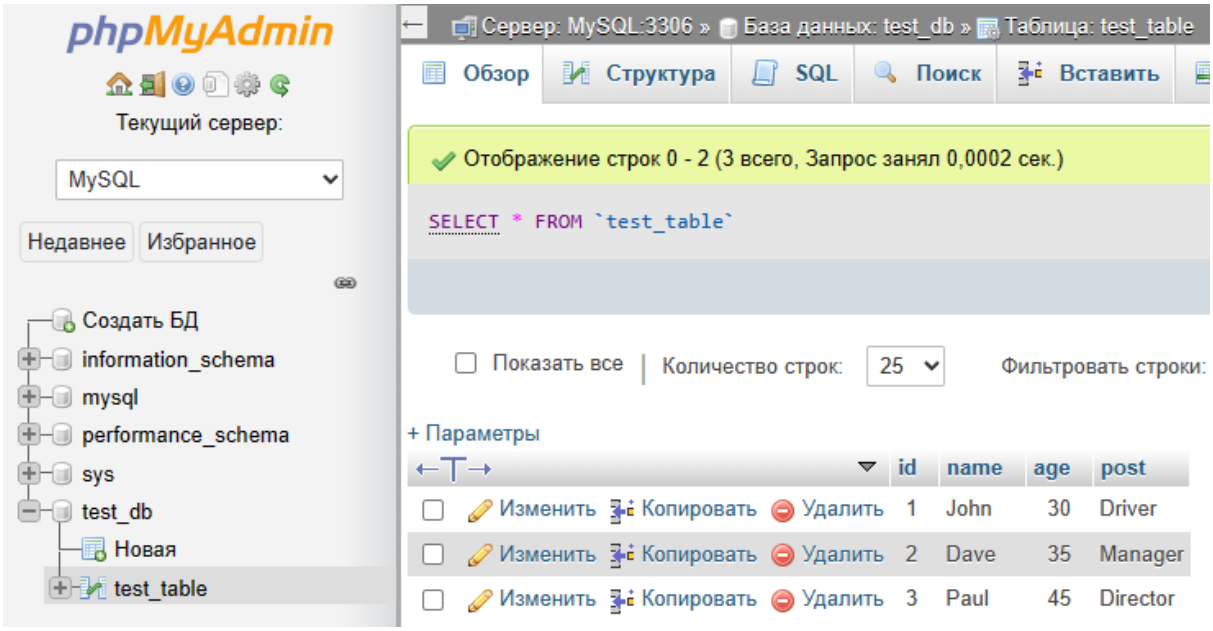
Created with the Personal Edition of HelpNDoc: Free EBook and documentation generator

An example of an autotest

An example of an autotest

After the plugin has been installed and connected to the project, let's consider a simple autotest.

We have created a simple database in MySQL named test_db
There is one test_table table in this database with the fields: id, name, age, post



Now let's describe a simple autotest that will add, modify and delete data in the database while checking the correctness of the results.

Файл: 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("Test #3 checking the
database");
            await tester.BrowserFullScreenAsync();
        }

        public async Task test()
        {
                DataTable dataTable = null;
            List<List<string>> entries = new
List<List<string>>();
                TesterMySql testerMySql = new
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)
            {
                    entries =
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();
        }
    }
}
```

Please note that for the autotests to work correctly, you need to connect the library

```
using HatPluginMySql;
```

First, there is a connection to the database

```
TesterMySql testerMySql = new TesterMySql(tester);
await
testerMySql.ConnectionOpenAsync
("server=127.0.0.1;uid=root;pwd=;database=test_db");
```

Data manipulation is performed using different methods

```
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'");
```

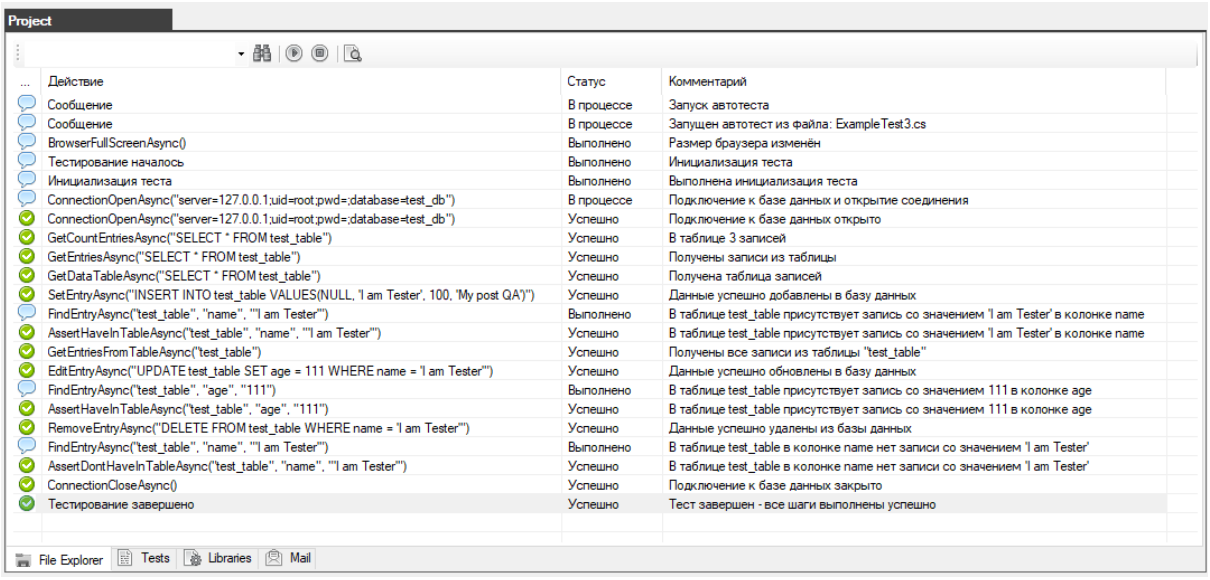
Special methods for checking the presence or absence of data in the database

```
await testerMySql.AssertHaveInTableAsync("test_table",
"name", "'I am Tester'");
await testerMySql.AssertDontHaveInTableAsync("test_table",
"name", "'I am Tester'")
```

At the end, the work with the database is completed by closing the connection to it

```
await testerMySql.ConnectionCloseAsync();
```

Let's run the autotest and look at the progress of the check and the result



All actions were completed, all checks were successful.

The test was completed successfully.

Created with the Personal Edition of HelpNDoc: Full-featured EBook editor

Class: TesterMySql

Created with the Personal Edition of HelpNDoc: Free EBook and documentation generator

Constructor

Created with the Personal Edition of HelpNDoc: Free help authoring environment

TesterMySql

TesterMySql
Description: the main class of working with a MySQL database
Syntax: TesterMySql(Tester tester)

Example:
Tester tester = new Tester(browserWindow);
TesterMySql testerMySql = new TesterMySql(tester);

Created with the Personal Edition of HelpNDoc: Easily create Help documents

Methods

Created with the Personal Edition of HelpNDoc: Free help authoring environment

ConnectionOpenAsync

ConnectionOpenAsync
Description: the method opens a connection to the MySQL database
Syntax: ConnectionOpenAsync(string connectionString)
Return value: no return value

Example:
```
await testerMySql.ConnectionOpenAsync("server=127.0.0.1;uid=root;pwd=;database=test_db");
```

Created with the Personal Edition of HelpNDoc: Full-featured multi-format Help generator

ConnectionCloseAsync

ConnectionCloseAsync
Description: the method closes the connection to the MySQL database
Syntax: ConnectionCloseAsync()
Return value: no return value

Example:
await testerMySql.ConnectionCloseAsync();

Created with the Personal Edition of HelpNDoc: iPhone web sites made easy

GetCountEntriesAsync

GetCountEntriesAsync
Description: the method returns the number of records in the table after executing the query
Syntax: GetCountEntriesAsync(string sqlQuertSelect)
Return value: int

Example:
int count = await testerMySql.GetCountEntriesAsync("SELECT * FROM test_table");

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

GetEntriesAsync

GetEntriesAsync

Description: the method returns a list of records from the table after executing the query
Syntax: GetEntriesAsync(string sqlQuertSelect)
Return value: List

Example:

```
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: Easily create CHM Help documents

GetEntriesFromTableAsync

GetEntriesFromTableAsync
Description: the method returns a list of records from the specified database table
Syntax: GetEntriesFromTableAsync(string tableName)
Return value: List

Example:

```
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);
```

Created with the Personal Edition of HelpNDoc: Create iPhone web-based documentation

GetDataTableAsync

GetDataTableAsync

Description: the method returns a table of records from the database table after executing the query
Syntax: GetDataTableAsync(string sqlQuertSelect)
Return value: DataTable

Example:

```
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());
```

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

SetEntryAsync

SetEntryAsync
Description: the method executes a query that adds data to the database table and returns the record number
Syntax: SetEntryAsync(string sqlQuertInsert)
Return value: int

Example:
int result = await testerMySql.SetEntryAsync("INSERT INTO test_table VALUES(NULL, 'I am Tester', 100, 'My post QA')");

Created with the Personal Edition of HelpNDoc: Easily create iPhone documentation

EditEntryAsync

EditEntryAsync
Description: the method executes a query that modifies the data in the database table and returns the record number
Syntax: EditEntryAsync(string sqlQuertUpdate)
Return value: int

Example:
int result = await testerMySql.EditEntryAsync("UPDATE test_table SET age = 111 WHERE name = 'I am Tester'");

Created with the Personal Edition of HelpNDoc: Full-featured EBook editor

RemoveEntryAsync

RemoveEntryAsync
Description: the method executes a query that deletes the data in the database table and returns the record number
Syntax: RemoveEntryAsync(string sqlQuertDelete)
Return value: int

Example:
int result = await testerMySql.RemoveEntryAsync("DELETE FROM test_table WHERE name = 'I am Tester'");

Created with the Personal Edition of HelpNDoc: Write EPub books for the iPad

FindEntryAsync

FindEntryAsync

Description: the method searches for data in the specified database table and returns the boolean value of the search result
Syntax: FindEntryAsync(string tableName, string columnName, string value)
Return value: bool (true or false)

Example:

```
bool result = await testerMySql.FindEntryAsync("test_table", "name", "'I am Tester'");
```

Created with the Personal Edition of HelpNDoc: Easily create Help documents

AssertHaveInTableAsync

AssertHaveInTableAsync

Description: the method checks the data in the specified database table and returns the boolean value of the search result, in case of a negative result, the check will be considered a failure

Syntax: AssertHaveInTableAsync(string tableName, string columnName, string value)

Return value: bool (true or false)

Example:

```
bool result = await testerMySql.AssertHaveInTableAsync("test_table", "name", "'I am Tester'");
```

Created with the Personal Edition of HelpNDoc: Free Kindle producer

AssertDontHaveInTableAsync

AssertDontHaveInTableAsync

Description: the method checks the data in the specified database table and returns the boolean value of the search result, in case of a positive result, the check will be considered a failure

Syntax: AssertDontHaveInTableAsync(string tableName, string columnName, string value)

Return value: bool (true or false)

Example:

```
bool result = await testerMySql.AssertDontHaveInTableAsync("test_table", "name", "'I am Tester'");
```

Created with the Personal Edition of HelpNDoc: Free Web Help generator