

# HatPluginMySql

## Table of contents

---

Русская версия .....	3
Описание .....	3
Плагины .....	3
Плагин HatPluginMySql .....	3
Установка плагина .....	3
Подключение плагина к проекту .....	6
Пример автотеста .....	7
Класс: TesterMySql .....	12
Конструктор .....	12
TesterMySql .....	13
Методы .....	13
ConnectionOpenAsync .....	14
ConnectionCloseAsync .....	15
GetCountEntriesAsync .....	16
GetEntriesAsync .....	17
GetEntriesFromTableAsync .....	18
GetDataTableAsync .....	19
SetEntryAsync .....	20
EditEntryAsync .....	21
RemoveEntryAsync .....	22
FindEntryAsync .....	23
AssertHaveInTableAsync .....	24
AssertDontHaveInTableAsync .....	25

## Русская версия

---

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

---

### Описание

# HatPluginMySql

---

HatPluginMySql - плагин для браузера Hat, который позволяет выполнять автоматизированное тестирование данных в базе данных MySQL.

Системные требования:

плагин совместим с браузером Hat начиная с версии 1.4

: <https://somovstudio.github.io/ru/>

GitFlic: <https://gitflic.ru/project/somovstudio/hat/>

27.03.2026) 1.5.0.3 (

---

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

---

### 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**

# HatPluginMySql

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 path "Локальный диск (C:) > Hat". The left pane shows the directory structure, with "plugin" highlighted. The right pane shows a list of files and folders, with a red box highlighting the contents of the "plugin" subdirectory. A red arrow points from the "plugin" folder in the left pane to the highlighted area in the right pane.

Имя	Дата изменения
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
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

!

---

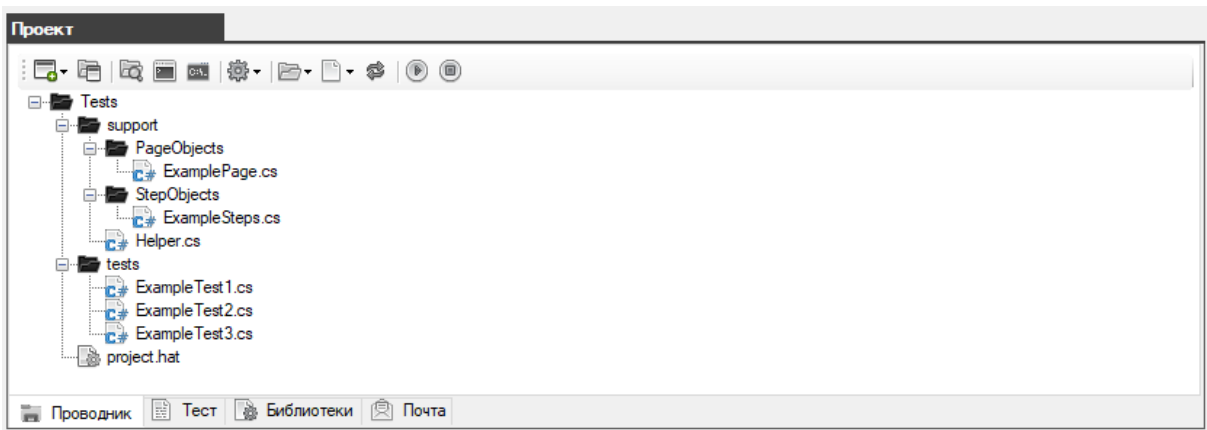
Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

---

## 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 database structure is visible, including 'test\_db' and 'test\_table'. The main area displays a successful query: 'SELECT \* FROM `test\_table`'. Below the query, there are options to 'Показать все' (Show all) and a dropdown for 'Количество строк' (Number of rows) set to 25. A table of results is shown with columns 'id', 'name', 'age', and 'post'.

	id	name	age	post
<input type="checkbox"/>	1	John	30	Driver
<input type="checkbox"/>	2	Dave	35	Manager
<input type="checkbox"/>	3	Paul	45	Director

: 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=;database=test_db")	В процессе	Подключение к базе данных и открытие соединения
ConnectionOpenAsync("server=127.0.0.1;uid=root;pwd=;database=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 sqlQuertInsert)

**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");
```