

---

# **Trio-MySQL Documentation**

***Release 0.7.2***

**Yutaka Matsubara and GitHub contributors**

**Mar 10, 2018**



---

## Contents

---

<b>1</b>	<b>User Guide</b>	<b>1</b>
1.1	Installation . . . . .	1
1.2	Examples . . . . .	1
1.3	Resources . . . . .	2
1.4	Development . . . . .	3
<b>2</b>	<b>API Reference</b>	<b>5</b>
2.1	Connection Object . . . . .	5
2.2	Cursor Objects . . . . .	5
<b>3</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>



The Trio-MySQL user guide explains how to install Trio-MySQL and how to contribute to the library as a developer.

## 1.1 Installation

The last stable release is available on PyPI and can be installed with `pip`:

```
$ pip install trio_mysql
```

### 1.1.1 Requirements

- Python – one of the following:
  - CPython `>= 3.5`
  - PyPy `>= 5.5`
- MySQL Server – one of the following:
  - MySQL `>= 4.1` (tested with only 5.5~)
  - MariaDB `>= 5.1`

## 1.2 Examples

### 1.2.1 CRUD

The following examples make use of a simple table

```
CREATE TABLE `users` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `email` varchar(255) COLLATE utf8_bin NOT NULL,  
    `password` varchar(255) COLLATE utf8_bin NOT NULL,  
    PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin  
AUTO_INCREMENT=1 ;
```

```
import trio_mysql.cursors  
  
# Connect to the database  
connection = trio_mysql.connect(host='localhost',  
                                user='user',  
                                password='passwd',  
                                db='db',  
                                charset='utf8mb4',  
                                cursorclass=trio_mysql.cursors.DictCursor)  
  
async with connection:  
    async with connection.transaction():  
        async with connection.cursor() as cursor:  
            # Create a new record  
            sql = "INSERT INTO `users` (`email`, `password`) VALUES (%s, %s)"  
            await cursor.execute(sql, ('webmaster@python.org', 'very-secret'))  
            # Transactions are auto-committed if they're exited without  
            # error.  
  
        async with connection.cursor() as cursor:  
            # Read a single record  
            sql = "SELECT `id`, `password` FROM `users` WHERE `email`=%s"  
            await cursor.execute(sql, ('webmaster@python.org',))  
            result = await cursor.fetchone()  
            print(result)  
  
            # When reading, you should periodically commit (or roll back) so  
            # that the database can release any read locks.  
            await connection.commit()  
            # In this case it's superfluous because we end the connection  
            # anyway.
```

This example will print:

```
{'password': 'very-secret', 'id': 1}
```

## 1.3 Resources

DB-API 2.0: <http://www.python.org/dev/peps/pep-0249>

MySQL Reference Manuals: <http://dev.mysql.com/doc/>

MySQL client/server protocol: <http://dev.mysql.com/doc/internals/en/client-server-protocol.html>

Gitter discussion: <https://gitter.im/python-trio/general>

## 1.4 Development

You can help developing Trio-MySQL by [contributing on GitHub](#).

### 1.4.1 Building the documentation

Go to the `docs` directory and run `make html`.

### 1.4.2 Test Suite

If you would like to run the test suite, create a database for testing like this:

```
mysql -e 'create database test_trio_mysql DEFAULT CHARACTER SET utf8 DEFAULT COLLATE_
↪utf8_general_ci;'
mysql -e 'create database test_trio_mysql2 DEFAULT CHARACTER SET utf8 DEFAULT COLLATE_
↪utf8_general_ci;'
```

Then, copy the file `.travis/database.json` to `tests/databases.json` and edit the new file to match your MySQL configuration:

```
$ cp .travis/database.json tests/databases.json
$ $EDITOR tests/databases.json
```

To run all the tests, execute the script `runtests.py`:

```
$ python runtests.py
```

A `tox.ini` file is also provided for conveniently running tests on multiple Python versions:

```
$ tox
```





## CHAPTER 2

---

### API Reference

---

If you are looking for information on a specific function, class or method, this part of the documentation is for you. For more information, please read the [Python Database API specification](#).

#### 2.1 Connection Object

#### 2.2 Cursor Objects



## CHAPTER 3

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### t

`trio_mysql.connections`, 5  
`trio_mysql.cursors`, 5



## T

`trio_mysql.connections` (module), [5](#)  
`trio_mysql.cursors` (module), [5](#)