**什么是 PyMySQL？**

PyMySQL 是在 Python3.x 版本中用于连接 MySQL 服务器的一个库，Python2中则使用mysqldb。

PyMySQL 遵循 Python 数据库 API v2.0 规范，并包含了 pure-Python MySQL 客户端库。

**PyMySQL 安装**

在使用 PyMySQL 之前，我们需要确保 PyMySQL 已安装。

PyMySQL 下载地址：https://github.com/PyMySQL/PyMySQL。

如果还未安装，我们可以使用以下命令安装最新版的 PyMySQL：

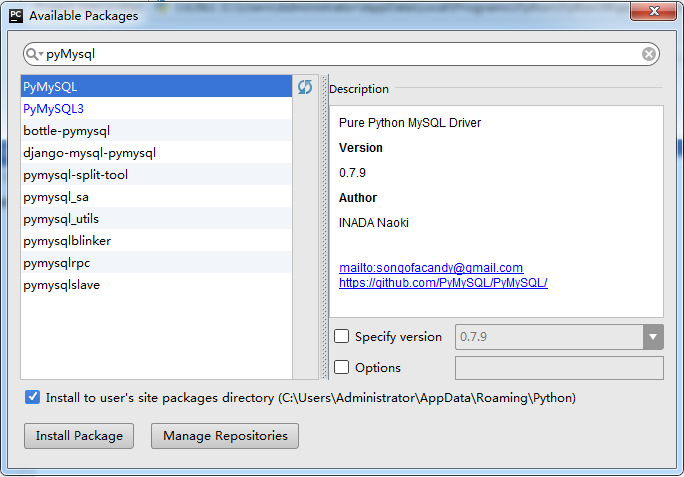
$ pip install PyMySQL

==================**pymysql**===================

由于 MySQLdb 模块还不支持 Python3.x，所以 Python3.x 如果想连接MySQL需要安装 pymysql 模块。

pymysql 模块可以通过 pip 安装。但如果你使用的是 pycharm IDE，则可以使用 project python 安装第三方模块。（pip –v pip list pip install pymysql）

[**File**] >> [**settings**] >> [**Project: python**] >> [**Project Interpreter**] >> [**Install按钮**]



由于Python统一了数据库连接的接口，所以 pymysql 和 MySQLdb 在使用方式上是类似的：

[复制代码](javascript:void(0);)

pymysql.Connect()参数说明

host(str): MySQL服务器地址

port(int): MySQL服务器端口号

user(str): 用户名

passwd(str): 密码

db(str): 数据库名称

charset(str): 连接编码

connection对象支持的方法

cursor() 使用该连接创建并返回游标

commit() 提交当前事务

rollback() 回滚当前事务

close() 关闭连接

cursor对象支持的方法

execute(op) 执行一个数据库的查询命令

fetchone() 取得结果集的下一行

fetchmany(size) 获取结果集的下几行

fetchall() 获取结果集中的所有行

rowcount() 返回数据条数或影响行数

close() 关闭游标对象

[复制代码](javascript:void(0);)

==================**MySQL**===================

首先在连接数据库之前，先创建一个交易表，方便测试 pymysql 的功能：

[复制代码](javascript:void(0);)

DROP TABLE IF EXISTS `trade`;

CREATE TABLE `trade` (

`id` int(4) unsigned NOT NULL AUTO\_INCREMENT,

`name` varchar(6) NOT NULL COMMENT '用户真实姓名',

`account` varchar(11) NOT NULL COMMENT '银行储蓄账号',

`saving` decimal(8,2) unsigned NOT NULL DEFAULT '0.00' COMMENT '账户储蓄金额',

`expend` decimal(8,2) unsigned NOT NULL DEFAULT '0.00' COMMENT '账户支出总计',

`income` decimal(8,2) unsigned NOT NULL DEFAULT '0.00' COMMENT '账户收入总计',

PRIMARY KEY (`id`),

UNIQUE KEY `name\_UNIQUE` (`name`)

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=utf8;

INSERT INTO `trade` VALUES (1,'乔布斯','18012345678',0.00,0.00,0.00);

[复制代码](javascript:void(0);)

==================**Python**===================

使用Python脚本实现增删改查和事务处理，源码如下：

[复制代码](javascript:void(0);)

import pymysql.cursors

# 连接数据库

connect = pymysql.Connect(

host='localhost',

port=3306,

user='root',

passwd='112476',

db='python',

charset='utf8'

)

# 获取游标

cursor = connect.cursor()

# 插入数据

sql = "INSERT INTO trade (name, account, saving) VALUES ( '%s', '%s', %.2f )"

data = ('雷军', '13512345678', 10000)

cursor.execute(sql % data)

connect.commit()

print('成功插入', cursor.rowcount, '条数据')

# 修改数据

sql = "UPDATE trade SET saving = %.2f WHERE account = '%s' "

data = (8888, '13512345678')

cursor.execute(sql % data)

connect.commit()

print('成功修改', cursor.rowcount, '条数据')

# 查询数据

sql = "SELECT name,saving FROM trade WHERE account = '%s' "

data = ('13512345678',)

cursor.execute(sql % data)

for row in cursor.fetchall():

print("Name:%s\tSaving:%.2f" % row)

print('共查找出', cursor.rowcount, '条数据')

# 删除数据

sql = "DELETE FROM trade WHERE account = '%s' LIMIT %d"

data = ('13512345678', 1)

cursor.execute(sql % data)

connect.commit()

print('成功删除', cursor.rowcount, '条数据')

# 事务处理

sql\_1 = "UPDATE trade SET saving = saving + 1000 WHERE account = '18012345678' "

sql\_2 = "UPDATE trade SET expend = expend + 1000 WHERE account = '18012345678' "

sql\_3 = "UPDATE trade SET income = income + 2000 WHERE account = '18012345678' "

try:

cursor.execute(sql\_1) # 储蓄增加1000

cursor.execute(sql\_2) # 支出增加1000

cursor.execute(sql\_3) # 收入增加2000

except Exception as e:

connect.rollback() # 事务回滚

print('事务处理失败', e)

else:

connect.commit() # 事务提交

print('事务处理成功', cursor.rowcount)

# 关闭连接

cursor.close()

connect.close()