configure_data_share_storage_service module
1、开发环境
2、目录结构及文件说明
3、类和函数接口说明
class server_socket_library
server_socket_library.parse_msg(pid, msg)
start_socket_thread(socket_handle, ip, port)
class client_socket_library
client_socket_library.parse_msg(pid, msg)
client_login(pid, table_name)
client_logout(client_obj, pid)
connect_sqlite()
close_sqlite()
create_table(table_name, sql)
add_confdata(table_name, conf_data)
del_confdata(table_name, primary_key)
chg_confdata(table_name, primary_key, column_name, value)
qry_confdata(table_name)
send_change_notify(table_name, ip, pid=2020)
4、调用接口样例

configure_data_share_storage_service module

Created on Wed Jul 8 10:06:41 2020

@author: 何健29996

@description:

配置数据共享存储模块

满足：

1.对外支持多线程/多进程的数据安全读写访问；

2.对外提供数据更改通知机制，能及时让外部获取数据的变化。

3.数据存储要求支持加密安全，加密算法要求可扩展；

说明：

1.配置数据存储在SQLite里，模块读取出来后转换为JSON格式给业务模块使用

1、开发环境

语言：python
编译器：python3.7.4
python库：anadonda3(version 1.7.3)
需要额外安装：pycrypto、rsa

2、目录结构及文件说明

- 日志目录：log

  包含配置数据共享存储模块日志confdata_sharestroage_service.log和通信日志socket_library.log

- 数据目录：data

  包含rsa加密的公钥public.pem和私钥private.pem文件，以及数据库文件confdatashare.db

- configure_data_share_storage_service.py

  配置数据共享存储模块服务程序，负责收发客户端的消息，进行操作数据库，通知修改机制

- encryption_algorithm_library.py：

  加密算法库文件

- socket_library.py

  socket通信库，发送消息和接收消息

- example_client.py

  客户端调用接口样例程序

- example_server.py

  服务端开启接收消息服务样例程序

3、类和函数接口说明

class server_socket_library

基类：socket_library.socket_library

服务端继承socket_library.

重写socket_library的parse_msg方法.

server_socket_library.parse_msg(pid, msg)

解析消息.

服务端对接收的消息进行解析，主要是客户端连接和断开的消息.

pid : int

哪个进程发来的消息

msg : str

消息内容

start_socket_thread(socket_handle, ip, port)

开启socket通信线程.

守护线程，负责接收消息.

socket_handle : object

socket套接字对象

ip : str

ip地址

port : int

socket监听端口

class client_socket_library

基类：socket_library.socket_library

客户端继承socket_library.

重写socket_library的parse_msg方法.

client_socket_library.parse_msg(pid, msg)

解析消息.

客户端对接收的消息进行解析，主要是弹出数据库内容修改通知框

pid : int

哪个进程发来的消息

msg : str

消息内容，格式应该为change,table_name

client_login(pid, table_name)

客户端登录.

客户端登录，与服务建立连接，并告诉服务端当前进程关注的业务数据表名称.

pid : int

当前进程pid

table_name : str

创建的表的名称

client_obj : object

客户端socket对象

client_logout(client_obj, pid)

客户端注销.

客户端退出时，与服务断开连接.

client_obj : object

客户端socket对象

pid : int

当前进程pid

connect_sqlite()

连接sqlite数据库.

连接数据库,并初始化数据库游标.

bool

成功返回True,失败False

close_sqlite()

关闭sqlite数据库.

create_table(table_name, sql)

创建数据表.

table_name : str

创建的表的名称

sql : str

合法的sql语句

bool

成功返回True,失败False

add_confdata(table_name, conf_data)

增加配置数据.

table_name : str

数据表名称

conf_data : str

配置数据

bool

成功返回True,失败返回False

del_confdata(table_name, primary_key)

删除配置数据.

table_name : str

数据表名称

primary_key : int

需要修改的配置数据的主键

bool

成功返回True,失败返回False

chg_confdata(table_name, primary_key, column_name, value)

修改配置数据.

table_name : str

数据表名称

primary_key : str

需要修改的配置数据的主键

column_name : str

需要修改的配置列名

value : str

修改后的值

bool

成功返回True,失败返回False

qry_confdata(table_name)

查询配置数据.

table_name : str

数据表名称

result_json : str

json格式的数据表内容

send_change_notify(table_name, ip, pid=2020)

发送修改通知.

数据库修改通知机制.

table_name : str

数据表名称

ip : str

ip地址

pid : int

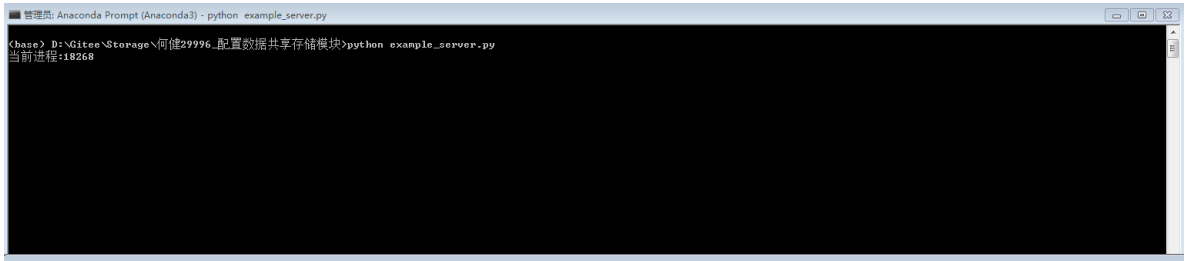
数据表被哪个进程修改，默认2020

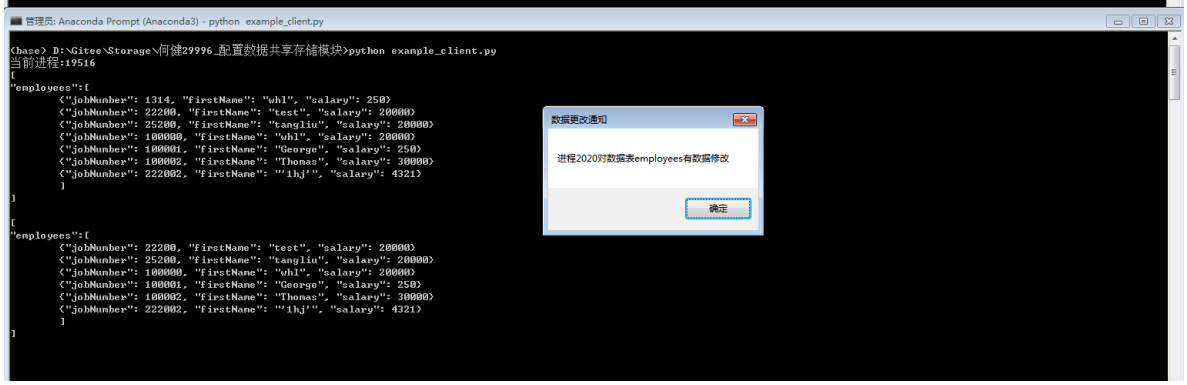
4、调用接口样例

见本地的example_server.py和example_client.py文件

- 首先运行example_server.py，打开服务端socket通信接口
- 然后运行example_client.py，在这个文件可以使用上述api的接口

运行见图：


管理员: Anaconda Prompt (Anaconda3) - python example_server.py
(base) D:\Gitee\Storage\何健29996_配置数据共享存储模块>python example_server.py
当前进程:18268


管理员: Anaconda Prompt (Anaconda3) - python example_client.py
(base) D:\Gitee\Storage\何健29996_配置数据共享存储模块>python example_client.py
当前进程:19516
{
"employees":[
{"jobNumber": 1314, "firstName": "whl", "salary": 250}
{"jobNumber": 22200, "firstName": "test", "salary": 20000}
{"jobNumber": 25200, "firstName": "tangliu", "salary": 20000}
{"jobNumber": 100000, "firstName": "whl", "salary": 20000}
{"jobNumber": 100001, "firstName": "George", "salary": 250}
{"jobNumber": 100002, "firstName": "Thomas", "salary": 30000}
{"jobNumber": 222002, "firstName": "'ihj'", "salary": 4321}
]
}
{
"employees":[
{"jobNumber": 22200, "firstName": "test", "salary": 20000}
{"jobNumber": 25200, "firstName": "tangliu", "salary": 20000}
{"jobNumber": 100000, "firstName": "whl", "salary": 20000}
{"jobNumber": 100001, "firstName": "George", "salary": 250}
{"jobNumber": 100002, "firstName": "Thomas", "salary": 30000}
{"jobNumber": 222002, "firstName": "'ihj'", "salary": 4321}
]
}

数据更改通知
进程2020对数据表employees有数据修改
确定