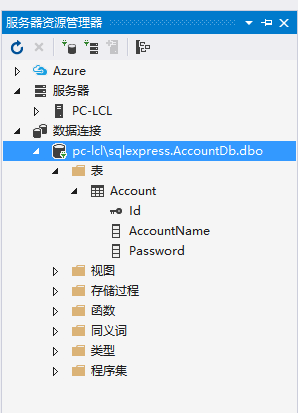
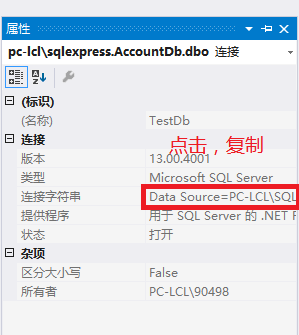
# **Unity C# 连接SQL Server数据库，实现获取和添加登录注册的用户列表。**

[参考：C#操作SQL Server数据库](http://www.cnblogs.com/rainman/archive/2012/03/13/2393975.html" \t "https://blog.csdn.net/l773575310/article/details/_blank) http://www.cnblogs.com/rainman/archive/2012/03/13/2393975.html  
[Github：C#操作SQL Server练习](https://github.com/ZeroChiLi/ConnectSQLServer/" \t "https://blog.csdn.net/l773575310/article/details/_blank) https://github.com/ZeroChiLi/ConnectSQLServer/  
[Github：MyGameServer（服务器、连接数据库）](https://github.com/ZeroChiLi/MyGameServer/" \t "https://blog.csdn.net/l773575310/article/details/_blank) https://github.com/ZeroChiLi/MyGameServer/  
[Github：PhotonChatRoom（Unity客户端）](https://github.com/ZeroChiLi/PhotonChatRoom" \t "https://blog.csdn.net/l773575310/article/details/_blank) https://github.com/ZeroChiLi/PhotonChatRoom  
[配合使用上一篇Blog：Unity 使用Photon Server 创建一个简单聊天室](http://blog.csdn.net/l773575310/article/details/72858315" \t "https://blog.csdn.net/l773575310/article/details/_blank)https://blog.csdn.net/l773575310/article/details/72858315

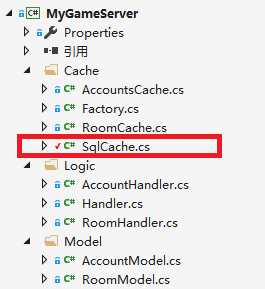
　　本文主要介绍如何连接本地数据库服务器，并且查询修改数据库数据，最后加入到之前做的聊天室服务器中，实现客户端登录注册帐号时会查询数据库，在对数据做操作时，经常容易抛异常，主要是因为数据库命令字符串有误，或者数据库命令没有绑定SqlConnection（数据库连接）。

# **工程连接数据库**

　　连接数据库这玩意的连接字符串弄了半天没成功，最后发现工程里面数据库属性里面有。我用的VS2015,SQL Server数据库，具体方法如下：VS 2015菜单中 视图->服务器资源管理器->数据连接->找到要连接数据库或者创建一个，右键属性->连接->连接字符串，复制即可。   
 

# **SqlCache 类，静态类，处理Sql的连接，查询，插入与删除。**

## **添加这个类到Cache层，因为他实际上也是处理数据的存储的。**



**数据连接的初始化，在静态构造函数。**

//数据库连接，所有数据库的操作都通过个成员变量操作。

static readonly private SqlConnection SqlConnection;

//静态构造函数初始化数据库连接,在使用任何静态成员前自动调用

static SqlCache() {

//视图->服务器资源管理器->数据连接->数据库右键->属性->连接->连接字符串

const string constr = "Data Source=PC-LCL\\SQLEXPRESS;Initial Catalog=TestDb;Integrated Security=True;Pooling=False";

SqlConnection = new SqlConnection(constr);

MyApplication.Log("连接数据库"); //输出到日志

}

OpenSql()、CloseSql()。打开、关闭数据库连接，在每次需要对数据库直接操作时调用。

//打开数据库连接，需要先判断是否是关闭，否则打开已经打开连接的数据库会抛异常。

if (SqlConnection.State == System.Data.ConnectionState.Closed)SqlConnection.Open();

//同理关闭连接数据库。

if (SqlConnection.State == System.Data.ConnectionState.Open) SqlConnection.Close();

IsMatchAccount(string accountName, string password = null)。查找是否存在帐号，或者查找帐号和密码是否匹配。之所以写在同一个方法，是这两种操作其实后者只是多了检查密码。

static public bool IsMatchAccount(string accountName, string password = null) {

if (!OpenSql()) return false; //打开数据连接

bool isMatch = false; //返回结果

StringBuilder commandStr = new StringBuilder(); //查询命令，就是SQL语法

commandStr.Append(" Select \* From Account");

commandStr.Append(" Where AccountName = '" + accountName + "'");

if (!string.IsNullOrEmpty(password)) //加了密码就是检测帐号密码同时匹配

commandStr.Append(" And Password = '" + password + "'");

try {

//执行命令添加到sqlCommand，执行并读取结果

SqlCommand sqlCommand = new SqlCommand(commandStr.ToString(), SqlConnection);

SqlDataReader reader = sqlCommand.ExecuteReader();

if (reader.Read()) isMatch = true; //如果有查到返回true

MyApplication.Log("查询用户存在、或账户密码匹配成功");

}

catch (Exception e) { MyApplication.Log("查询用户存在 失败了啊 " + e.ToString()); }

finally { CloseSql(); } //关闭数据连接

return isMatch;

}

AddAccount(string accountName, string password)、DeleteAccount(string accountName)。添加、删除用户。过程其实和上面的差不多，只是数据命令不同，已经在开头需要做一个保险判断（正常情况不会出现）。

//在添加用户前判断是否已经存在该帐号

if (IsMatchAccount(accountName)) return false;

//在删除用户前也要判断，不存在的话就删不了

if (!IsMatchAccount(accountName)) return false;

GetAccountList()。获取用户列表，就是把数据的用户全部遍历一遍把数据存到List<AccountModel>中。

SqlCommand sqlCommand = new SqlCommand("Select \* from Account", SqlConnection);

SqlDataReader reader = sqlCommand.ExecuteReader();

while (reader.Read())

accountList.Add(new AccountModel(reader["AccountName"].ToString(), reader["Password"].ToString()));

最后在AccountsCache类中添加两句话，就可以实现客户端间接访问数据库信息，添加用户到数据了，其实就是一开始把数据的用户都拿出来存到个列表里，除了添加用户，其他操作都是操作这个临时表，而不是查数据库了。

// AccountsCache类中 //构造函数中添加一句

public AccountsCache() {

accountModelList = new List<AccountModel>();

clientModelDict = new Dictionary<MyClientPeer, AccountModel>();

//把数据库的用户都加到列表里。

accountModelList = SqlCache.GetAccountList();

}

//注册时，添加帐号同时，添加到数据库中。

public void Add(string accountName, string password) {

accountModelList.Add(new AccountModel(accountName, password));

SqlCache.AddAccount(accountName, password); //添加到数据库

}

# **测试**

|  |  |
| --- | --- |
| **数据库原本数据。** | IMG_259 |
| **启动MyGameServer服务器，在客户端登录“阿迪王”帐号。** | IMG_260 |
| **登录成功。** | IMG_261 |
| **退出，注册一个新的帐号。** | IMG_262 |
| **刷新数据库，会发现有新的数据（最后一行）。** | IMG_263 |
| **再次登录新账号，没毛病。** | IMG_264 |