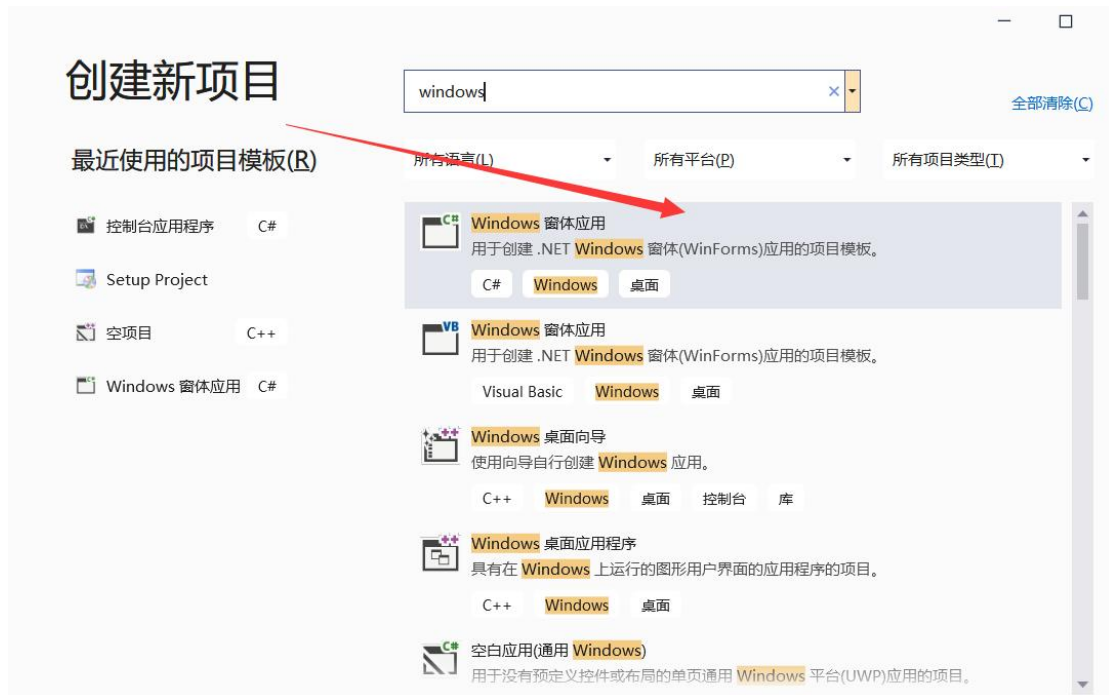


手把手带你完成服务器端

Designed by skuuklzc

第一步：创建项目



其他信息

Windows 窗体应用 C# Windows 桌面

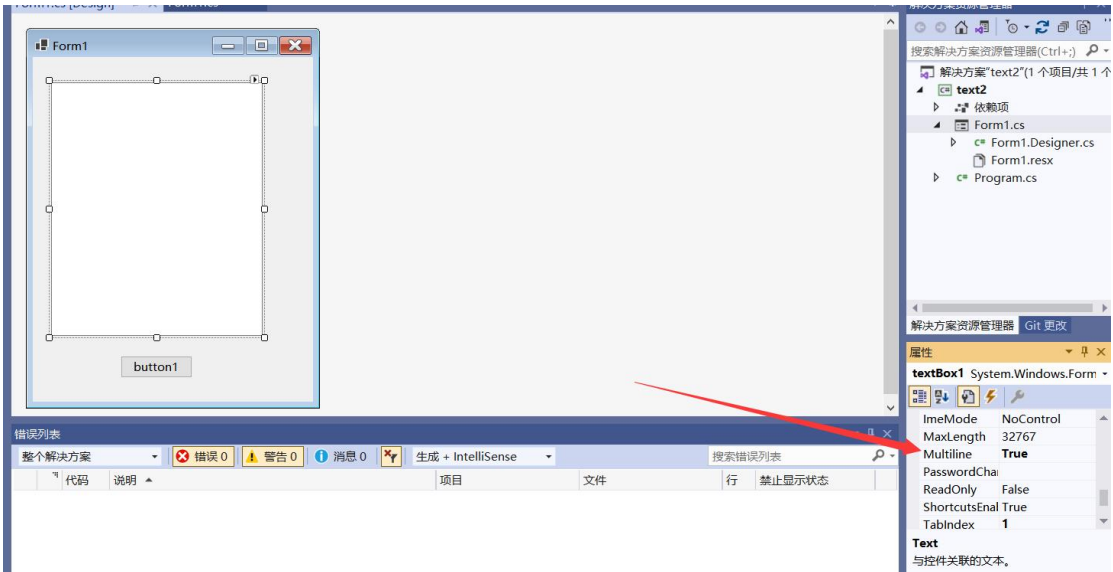
目标框架 ⓘ

.NET Core 3.1 (不受支持)

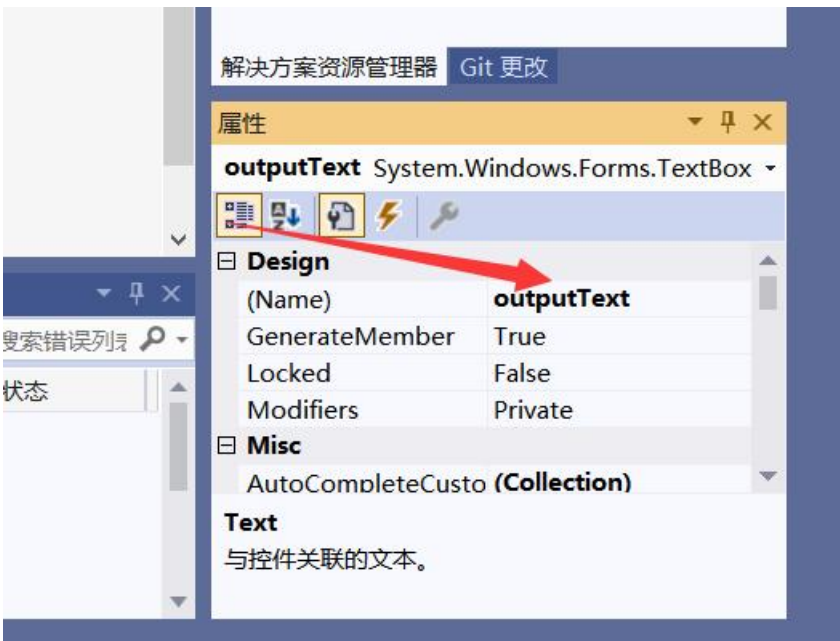


第二步：design 我们的窗口。简单一下就是这样，一个 button，一个 textbox
在左侧工具箱能找到这两个组件。







ps：在设计 textbox 时，记得把这个设为 true，不然你的文本框只有一行
给他们简单改个名字，这个还是在右下角可以改，不改也行。



改名在这里：



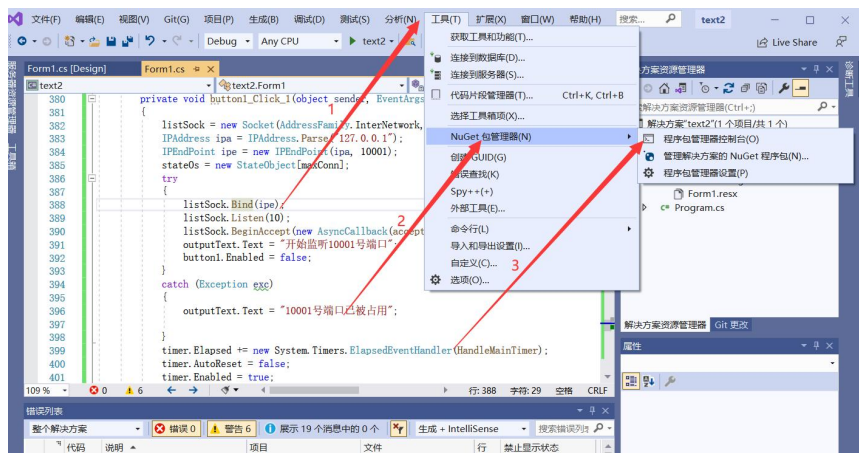
第三步：设计搞定了，直接使用抄袭法加拼凑法，把 server01 代码和注册登录代码整合在一起。这里提供整合好的框架代码。是在另一个文件里面：服务器.cs
自己动手整合，好处多多

 serve01服务器源码.rar	2022-12-13	永久	3.37KB	紫霜神...	60次	
 Server01.exe	2022-12-13	永久	16KB	紫霜神...	81次	
 注册登录代码.rar	2022-12-02	永久	3.86KB	紫霜神...	74次	

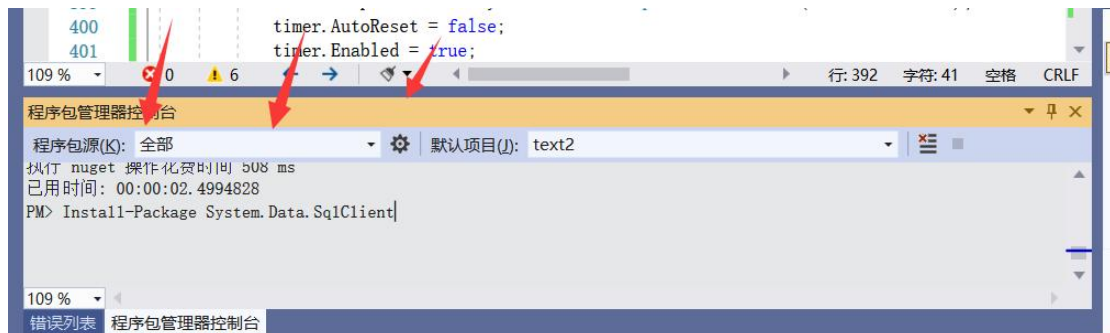
打开文件：



不要慌，正常操作而已：



最下面有个这个：输入这个命令 `Install-Package System.Data.SqlClient`，回车就可以安装包了。



还是有错怎么办：

```

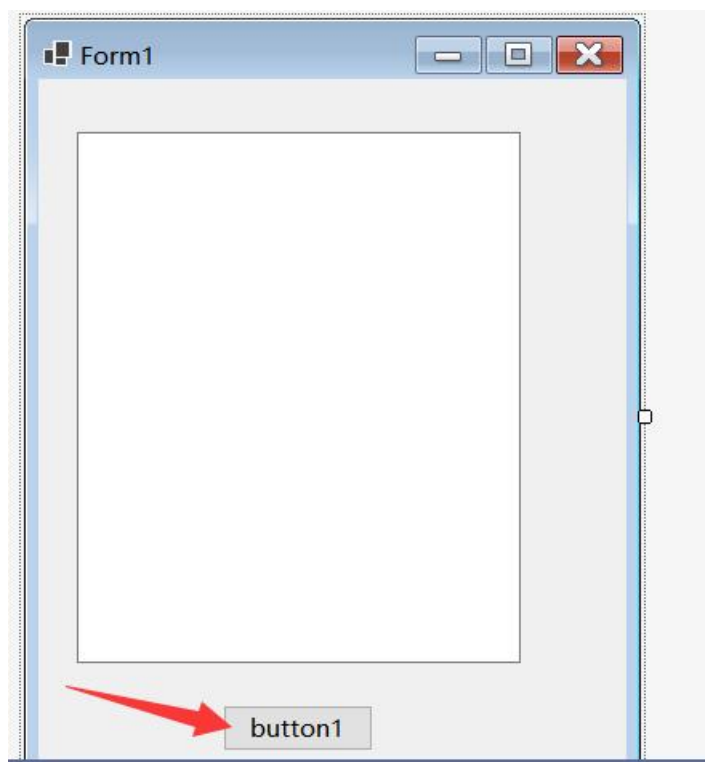
10 using System.Data.SqlClient;
11
12
13 namespace text2
14 {
15     3 个引用
16     public partial class Form1 : Form
17     {
18         const byte REGIST_REQUEST = 1;
19         const byte REGIST_FAIL = 2;
20         const byte REGIST_SUCCESS = 3;
21         const byte LOGIN_REQUEST = 4;
22         const byte LOGIN_FAIL, USERDONTEXIST = 5;

```

命名检查是否正确。

终于没错误了，开开心心去运行。

这按钮点不动怎么回事

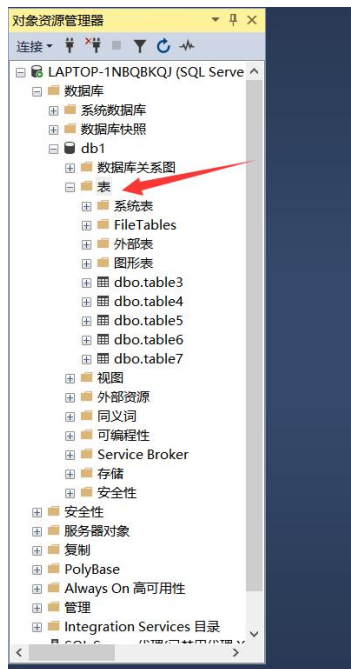


双击它，看看这个点击响应函数是不是空的，如果是空的，直接把 button1_Click 函数剪切给它

这下终于没问题了。

第四步：打开 sql server 创建表（ps：这步注意点噢，看我建了那么多表就知道我犯了多少错了）

右键创建新表



设置表的字段：要想代码不报错，和我一模一样，不一样也行，改也挺简单的。但是是否允许 null 值必须和我一样。

LAPTOP-1NBQBKQJ...b1 - dbo.table5			
列名	数据类型	允许 Null 值	
userName	nchar(20)	<input type="checkbox"/>	
passwordMD5	varchar(32)	<input type="checkbox"/>	
userIP	varchar(50)	<input checked="" type="checkbox"/>	
online	int	<input checked="" type="checkbox"/>	
score	int	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

设置好了记得保存

保存好了右键刷新下，就有刚才创建的表了

右键选择前 1000 行可以检查下列名对不对，一开始是个空表

100 %					
结果 消息					
	userName	passwordMD5	userIP	online	score
1	aaa	698D51A19D8A121CE581499D7B701668	NULL	0	5
2	bbb	698D51A19D8A121CE581499D7B701668	NULL	0	3
3	ccc	698D51A19D8A121CE581499D7B701668	127.0.0.1:12286	1	0

第五步：修改框架代码：

```
private void button1_Click(object sender, EventArgs e)
{
    listSock = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
    IPAddress ipa = IPAddress.Parse("127.0.0.1");
    IPEndPoint ipe = new IPEndPoint(ipa, 10001);
    stateOs = new StateObject[maxConn];
    try
    {
        listSock.Bind(ipe);
        listSock.Listen(10);
        listSock.BeginAccept(new AsyncCallback(acceptCb), listSock);
        outputText.Text = "开始监听10001号端口";
        button1.Enabled = false;
    }
    catch (Exception exc)
    {
        outputText.Text = "10001号端口已被占用";
    }
}
```

这个名字熟悉吧，这是设计窗口 forms 时的按钮名字，改成自己的按钮名。

```
void recvCb(IAsyncResult iar)
{
    String recvStr = "";
    StateObject stateo1 = (StateObject)iar.AsyncState;
    Socket client = stateo1.sock;
    if (client.Connected == false) { outputText.Text = "连接失败"; return; }
    try
    {
        int recvNum = stateo1.sock.EndReceive(iar);
        recvStr = System.Text.Encoding.ASCII.GetString(stateo1.buffer, 0, recvNum);
        if (recvStr.IndexOf("\0") != -1)
        {
            recvStr = recvStr.Substring(0, recvStr.IndexOf("\0"));
        }
        string[] args = recvStr.Split(' ');

        if ("registRequest" == args[0])
        {
            Regist(args[1], args[2], stateo1);
        }
        if ("loginRequest" == args[0])
        {
            Login(args[1], args[2], stateo1);
        }
        if ("createRoom" == args[0])
        {
            NewRoom(stateo1);
        }
        if ("heartBeat" == args[0])
        {
            outputText.Text = "收到一个心跳包" + num.ToString();
            num++;
        }
    }
}
```

这两个协议是注册请求和登录请求的协议，在群问价注册登录代码里面的服务器代码里他们长这样：


```

switch (buff_type)
{
    case REGIST_REQUEST:
        Regist(values[0], values[1], tempSock);
        break;
    case LOGIN_REQUEST:
        Login(values[0], values[1], tempSock);
        break;
}
// 自此开始到网络传输中

```

如果想用原来的协议名的话就自己小改一下代码吧~
和我一样的话，那 Network 脚本就要修改一下
点击注册按钮：

```

public void SendBtnClicked()
{
    string inputStr = inputText.text;
    var spaceIndex = inputStr.IndexOf(" ");
    if (spaceIndex != -1) { recvStr = "用户名存在空格"; return; }
    String passwordMD5Str = GetMD5String(inputPassword.text);
    String sendStr = "registRequest " + inputText.text + " " + passwordMD5Str;
    sendBuff = System.Text.Encoding.Default.GetBytes(sendStr);
    // sendBuff[0] = REGIST_REQUEST;

    clientSock.Send(sendBuff);
}

```

点击登录按钮：

```

public void LoginBtnClicked()
{
    // SceneManager.LoadScene(1);
    String passwordMD5Str = GetMD5String(inputPassword.text);
    String sendStr = "loginRequest " + inputText.text + " " + passwordMD5Str;
    sendBuff = System.Text.Encoding.Default.GetBytes(sendStr);
    userName = inputText.text;
    // sendBuff[0] = LOGIN_REQUEST;
    clientSock.Send(sendBuff);
}

```

从这里开始，到后面每个涉及到数据库 sql 语句的，都得改成自己的表名：

```

}
void Regist(string userName, string passwordMD5, StateObject stateo1)
{
    //string userName = "aacc";
    //string passwordMD5 = "1234567890ABCDEF1234567890ABCDEF";
    SqlConnection conn = new SqlConnection(connStr);
    conn.Open();
    string sqlStr = "SELECT * FROM table5 WHERE userName='" + userName + "'";
    SqlCommand command = new SqlCommand(sqlStr, conn);
    SqlDataReader odrReader = command.ExecuteReader();
    command.Dispose();
    if (odrReader.HasRows)
        SendRegistFail(stateo1);
    else

```

改完了，大功告成了！

第六步、试运行程序

当我点击注册时，客户端收到注册成功的提示，代表注册成功



此时，我们去到 sql server 看看有没有写进数据库

100 %

结果		消息			
	userName	passwordMD5	userIP	online	score
1	aaa	698D51A19D8A121CE581499D7B701668	NULL	0	5
2	bbb	698D51A19D8A121CE581499D7B701668	NULL	0	3
3	ddd	698D51A19D8A121CE581499D7B701668	NULL	0	0
4	ccc	698D51A19D8A121CE581499D7B701668	NULL	0	0

可以看到，该用户已写进我们的数据库，这样注册功能就实现了。
还有更多的调试细节，不予一一展示。