

网络层目录结构

Net – 网络层根目录，所有网络层相关代码在此目录下

Http – http链接相关功能目录，主要是登陆注册

RequestBase – 网络请求基础类

RespBase – 网络响应基础类

Request – 网络请求类

Resp – 网络响应类

Test – 测试类

网络层使用

1.登录注册

LoginManager必须搭载到游戏物体上

LoginManager为单例

设置回调接口

请求前设置httpFinishedDelegate回调。

请求成功后LoginManager 会调用 httpFinishedDelegate;

登录接口

```
public void Login(string username, string password)
```

注册接口

```
public void Regist(string username, string password)
```

注册成功后会自动调用login接口进行登录

2.游戏服务器

SocketHelper必须搭载到游戏物体上并且不能被销毁

SocketHelper为单例

1.添加协议流程

在NetProtocols脚本中定义协议

2.添加请求

在Request目录中创建请求脚本

请求必须继承自Request类

请求必须重写GetProtocol和Serialize方法

GetProtocol方法中返回NetProtocols中定义的协议

Serialize为序列化函数，该函数必须首先执行base.Serialize(writer);

3.添加响应

在Resp目录中创建响应脚本

响应必须继承自Resp类

响应必须重写GetProtocol和Deserialize方法

GetProtocol方法中返回NetProtocols中定义的协议

Deserialize为反序列化函数，该函数必须首先执行
`base.Deserialize(writer);`

4.注册响应

新增的响应必须在RegisterResp脚本的RegisterAll方法中注册
`ProtoManager.Instance.AddProtocol<新增Resp脚本>(新增协议);`

5.接受响应事件

对象想要接受一个服务器的响应，需要在请求之前使用此方法进行注册

`ProtoManager.Instance.AddRespDelegate(协议,回调代理);`

回调代理定义：`responseDelegate(Resp resp);`

在对象销毁时，或想要对象停止接收事件调用此方法即可