**Unity3D教程：Photon服务器入门（二）**

Posted on 2013年07月11日 by U3d / [Unity3D 基础教程](http://www.unitymanual.com/category/manual/unity3d-%e5%9f%ba%e7%a1%80%e6%95%99%e7%a8%8b) /被围观 4 次

[Unity3D教程：Photon服务器入门（一）](http://www.unitymanual.com/7317.html)中主要介绍了服务器的简单知识，配置服务器和客户端连接。

这篇以一个简单的用户登录为基础介绍客户端请求服务器，服务器响应操作。

一、服务器端

按照上一篇教程我们配置好简单的photon服务器，但是只能用于连接服务器和断开服务器操作，其他的基本没有提到，今天是要在上一讲基础上添加内容。

主要是在MyPeer.cs类的OnOperationRequest方法中实现，代码如下：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 01 | | | **using** System; | |
| 02 | | | **using** System.Collections.Generic; | |
| 03 | | | **using** Photon.SocketServer; | |
| 04 | | | **using** PhotonHostRuntimeInterfaces; | |
| 05 | | | **namespace** MyServer | |
| 06 | | | { | |
| 07 | | | **using** Message; | |
| 08 | | | **using** System.Collections; | |
| 09 | | | **public** **class** MyPeer : PeerBase | |
| 10 | | | { | |
| 11 | | | Hashtable userTable; | |
| 12 | | | **public** MyPeer(IRpcProtocol protocol,IPhotonPeer photonPeer) | |
| 13 | | | : **base**(protocol, photonPeer) | |
| 14 | | | { | |
| 15 | | | userTable = new Hashtable(); | |
| 16 | | | userTable.**Add**("user1","pwd1"); | |
| 17 | | | userTable.**Add**("user2", "pwd2"); | |
| 18 | | | userTable.**Add**("user3","pwd3"); | |
| 19 | | | userTable.**Add**("user4", "pwd4"); | |
| 20 | | | userTable.**Add**("user5","pwd5"); | |
| 21 | | | } | |
| 22 | | | **protected** overridevoid OnDisconnect(PhotonHostRuntimeInterfaces.DisconnectReason reasonCode,**string** reasonDetail) | |
| 23 | | | { | |
| 24 | | | } | |
| 25 | | | **protected** **override** **void** OnOperationRequest(OperationRequest operationRequest, SendParameters sendParameters) | |
| 26 | | | { | |
| 27 | | | **switch** (operationRequest.OperationCode) { | |
| 28 | | | **case** (**byte**)OpCodeEnum.Login: | |
| 29 | | | **string** uname = (**string**)operationRequest.Parameters[(**byte**)OpKeyEnum.UserName]; | |
| 30 | | | **string** pwd = (**string**)operationRequest.Parameters[(**byte**)OpKeyEnum.PassWord]; | |
| 31 | | | **if** (userTable.ContainsKey(uname) && userTable[uname].Equals(pwd))*//login success* | |
| 32 | | | { | |
| 33 | | | SendOperationResponse(new OperationResponse((**byte**)OpCodeEnum.LoginSuccess,**null**),new SendParameters()); | |
| 34 | | | }*//Unity3D教程手册：www.unitymanual.com* | |
| 35 | | | **else** | |
| 36 | | | { *//login fauled* | |
| 37 | | | SendOperationResponse(new OperationResponse((**byte**)OpCodeEnum.LoginFailed,**null**), new SendParameters()); | |
| 38 | | | } | |
| 39 | | | **break**; | |
| 40 | | | } | |
| 41 | | | } | |
| 42 | | | } | |
| 43 | | | } | |
|  |  | |  |

|  |  |
| --- | --- |
| 01 | **using** System; |
| 02 | **using** System.Collections.Generic; |
| 03 | **using** Photon.SocketServer; |
| 04 | **using** PhotonHostRuntimeInterfaces; |
| 05 |  |
| 06 |  |
| 07 | **namespace** MyServer |
| 08 | { |
| 09 | **using** Message; |
| 10 | **using** System.Collections; |
| 11 |  |
| 12 | **public** **class** MyPeer : PeerBase |
| 13 | { |
| 14 | Hashtable userTable; |
| 15 |  |
| 16 | *//Unity3D教程手册：www.unitymanual.com* |
| 17 | **public** MyPeer(IRpcProtocol protocol,IPhotonPeer photonPeer) |
| 18 | : **base**(protocol, photonPeer) |
| 19 | { |
| 20 | userTable = new Hashtable(); |
| 21 | userTable.**Add**("user1", "pwd1"); |
| 22 | userTable.**Add**("user2", "pwd2"); |
| 23 | userTable.**Add**("user3", "pwd3"); |
| 24 | userTable.**Add**("user4", "pwd4"); |
| 25 | userTable.**Add**("user5", "pwd5"); |
| 26 | } |
| 27 |  |
| 28 | **protected** **override** **void** OnDisconnect(PhotonHostRuntimeInterfaces.DisconnectReason reasonCode, **string** reasonDetail) |
| 29 | { |
| 30 |  |
| 31 | } |
| 32 |  |
| 33 | **protected** **override** **void** OnOperationRequest(OperationRequest operationRequest, SendParameters sendParameters) |
| 34 | { *//Unity3D教程手册：www.unitymanual.com* |
| 35 | **switch** (operationRequest.OperationCode) { |
| 36 | **case** (**byte**)OpCodeEnum.Login: |
| 37 | **string** uname = (**string**)operationRequest.Parameters[(**byte**)OpKeyEnum.UserName]; |
| 38 | **string** pwd = (**string**)operationRequest.Parameters[(**byte**)OpKeyEnum.PassWord]; |
| 39 |  |
| 40 | **if** (userTable.ContainsKey(uname) && userTable[uname].Equals(pwd))*//login success* |
| 41 | { |
| 42 | SendOperationResponse(new OperationResponse((**byte**)OpCodeEnum.LoginSuccess, **null**),new SendParameters()); |
| 43 | } |
| 44 | **else** |
| 45 | { *//login fauled* |
| 46 | SendOperationResponse(new OperationResponse((**byte**)OpCodeEnum.LoginFailed, **null**), new SendParameters()); |
| 47 | } |
| 48 | **break**; |
| 49 | } |
| 50 | } |
| 51 | } |
| 52 | } |

OnOperationRequest方法中验证用户名和密码，然后发送响应给客户端.需要用到的枚举一个类如下：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 01 | | | **using** System; | |
| 02 | | | **using** System.Collections.Generic; | |
| 03 | | | **using** System.Linq; | |
| 04 | | | **using** System.Text; | |
| 05 | | | **namespace** MyServer.Message | |
| 06 | | | { | |
| 07 | | | **enum** OpCodeEnum : **byte** | |
| 08 | | | { | |
| 09 | | | *//login* | |
| 10 | | | Login = 249, | |
| 11 | | | LoginSuccess = 248, | |
| 12 | | | LoginFailed = 247, | |
| 13 | | | *//room* | |
| 14 | | | Create = 250, | |
| 15 | | | **Join** = 255, | |
| 16 | | | Leave = 254, | |
| 17 | | | RaiseEvent = 253, | |
| 18 | | | SetProperties = 252, | |
| 19 | | | GetProperties = 251 | |
| 20 | | | } | |
| 21 | | | **enum** OpKeyEnum : **byte** | |
| 22 | | | { | |
| 23 | | | RoomId = 251, | |
| 24 | | | UserName = 252, | |
| 25 | | | PassWord = 253 | |
| 26 | | | } | |
| 27 | | | } | |
|  |  | |  |

|  |  |
| --- | --- |
| 01 | **using** System; |
| 02 | **using** System.Collections.Generic; |
| 03 | **using** System.Linq; |
| 04 | **using** System.Text; |
| 05 |  |
| 06 | **namespace** MyServer.Message |
| 07 | { |
| 08 | **enum** OpCodeEnum : **byte** |
| 09 | { |
| 10 | *//login* |
| 11 | Login = 249, |
| 12 | LoginSuccess = 248, |
| 13 | LoginFailed = 247, |
| 14 |  |
| 15 | *//room* |
| 16 | Create = 250, |
| 17 | **Join** = 255, |
| 18 | Leave = 254, |
| 19 | RaiseEvent = 253, |
| 20 | SetProperties = 252, |
| 21 | GetProperties = 251 |
| 22 | } |
| 23 |  |
| 24 |  |
| 25 | **enum** OpKeyEnum : **byte** |
| 26 | { |
| 27 | RoomId = 251, |
| 28 | UserName = 252, |
| 29 | PassWord = 253 |
| 30 | } |
| 31 | } |

二、客户端

客户端过程需要请求服务器并接收服务器的响应下面上代码，就一个类搞定：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 001 | | | **using** UnityEngine; | |
| 002 | | | **using** System.Collections; | |
| 003 | | | **using** System.Collections.Generic; | |
| 004 | | | **using** ExitGames.Client.Photon; | |
| 005 | | | **public** **class** TestConnection : MonoBehaviour,IPhotonPeerListener { | |
| 006 | | | **public** **string** address; | |
| 007 | | | PhotonPeer peer; | |
| 008 | | | ClientState state = ClientState.DisConnect; | |
| 009 | | | **string** username =""; | |
| 010 | | | **string** password = ""; | |
| 011 | | | *// Use this for initialization* | |
| 012 | | | **void** Start () { | |
| 013 | | | peer = new PhotonPeer(**this**,ConnectionProtocol.Udp); | |
| 014 | | | } | |
| 015 | | | *// Update is called once per frame* | |
| 016 | | | **void** Update () { | |
| 017 | | | peer.Service(); | |
| 018 | | | } | |
| 019 | | | **public** GUISkin skin; | |
| 020 | | | **void** OnGUI(){ | |
| 021 | | | GUI.skin = skin; | |
| 022 | | | **switch**(state){ | |
| 023 | | | **case** ClientState.DisConnect: | |
| 024 | | | GUI.Label(new Rect(Screen.width/2,Screen.height/2-30,300,40),"click the button to connect."); | |
| 025 | | | **if**(GUI.Button(new Rect(Screen.width/2,Screen.height/2,100,30),"Connect")){ | |
| 026 | | | peer.Connect(address,"MyServer"); | |
| 027 | | | state = ClientState.Connecting; | |
| 028 | | | } | |
| 029 | | | **break**; | |
| 030 | | | **case** ClientState.Connecting: | |
| 031 | | | GUI.Label(new Rect(Screen.width/2,Screen.height/2-30,300,40),"Connecting to Server..."); | |
| 032 | | | **break**; | |
| 033 | | | **case** ClientState.Connected: | |
| 034 | | | GUILayout.BeginArea(new Rect((Screen.width-500)/2,(Screen.height-400)/2,500,400)); | |
| 035 | | | *//* | |
| 036 | | | GUILayout.BeginVertical(); | |
| 037 | | | GUILayout.Label("Connect Success! Please Login."); | |
| 038 | | | *//draw username* | |
| 039 | | | GUILayout.BeginHorizontal(); | |
| 040 | | | GUILayout.Label("UserName:"); | |
| 041 | | | username = GUILayout.TextField(username); | |
| 042 | | | GUILayout.EndVertical(); | |
| 043 | | | *//draw password* | |
| 044 | | | GUILayout.BeginHorizontal(); | |
| 045 | | | GUILayout.Label("Password:"); | |
| 046 | | | password = GUILayout.TextField(password); | |
| 047 | | | GUILayout.EndVertical(); | |
| 048 | | | *//draw buttons* | |
| 049 | | | GUILayout.BeginHorizontal(); | |
| 050 | | | **if**(GUILayout.Button("login")){ | |
| 051 | | | userLogin(username,password); | |
| 052 | | | } | |
| 053 | | | **if**(GUILayout.Button("canel")){ | |
| 054 | | | state = ClientState.DisConnect; | |
| 055 | | | } | |
| 056 | | | GUILayout.EndVertical(); | |
| 057 | | | GUILayout.EndVertical(); | |
| 058 | | | GUILayout.EndArea(); | |
| 059 | | | **break**; | |
| 060 | | | **case** ClientState.ConnectFailed: | |
| 061 | | | GUI.Label(new Rect(Screen.width/2,Screen.height/2-30,300,40),"Connect Failed."); | |
| 062 | | | **break**; | |
| 063 | | | **case** ClientState.LoginSuccess: | |
| 064 | | | GUILayout.BeginArea(new Rect((Screen.width-500)/2,(Screen.height-400)/2,500,400)); | |
| 065 | | | GUILayout.Label("Login Success!"); | |
| 066 | | | GUILayout.EndArea(); | |
| 067 | | | **break**; | |
| 068 | | | **case** ClientState.LoginFailed: | |
| 069 | | | GUILayout.BeginArea(new Rect((Screen.width-500)/2,(Screen.height-400)/2,500,400)); | |
| 070 | | | GUILayout.Label("Login Failed!"); | |
| 071 | | | GUILayout.EndArea(); | |
| 072 | | | **break**; | |
| 073 | | | } | |
| 074 | | | } | |
| 075 | | | #region My Method | |
| 076 | | | IEnumerator connectFailedHandle(){ | |
| 077 | | | **yield** **return** new WaitForSeconds(1); | |
| 078 | | | state = ClientState.DisConnect; | |
| 079 | | | } | |
| 080 | | | **void** userLogin(**string** uname,**string** pwd){ | |
| 081 | | | Debug.Log("userLogin"); | |
| 082 | | | Dictionary<**byte**,**object**> param =new Dictionary<**byte**,**object**>(); | |
| 083 | | | param.**Add**((**byte**)OpKeyEnum.UserName,uname); | |
| 084 | | | param.**Add**((**byte**)OpKeyEnum.PassWord,pwd); | |
| 085 | | | peer.OpCustom((**byte**)OpCodeEnum.Login,param,**true**); | |
| 086 | | | } | |
| 087 | | | IEnumerator loginFailedHandle(){ | |
| 088 | | | **yield** **return** new WaitForSeconds(1); | |
| 089 | | | Debug.Log("loginFailedHandle"); | |
| 090 | | | state = ClientState.Connected; | |
| 091 | | | } | |
| 092 | | | #endregion | |
| 093 | | | #region IPhotonPeerListener implementation | |
| 094 | | | **public** **void** DebugReturn (DebugLevel level,**string** message) | |
| 095 | | | { | |
| 096 | | | } | |
| 097 | | | **public** **void** OnOperationResponse (OperationResponse operationResponse) | |
| 098 | | | { | |
| 099 | | | **switch**(operationResponse.OperationCode) | |
| 100 | | | { | |
| 101 | | | **case** (**byte**)OpCodeEnum.LoginSuccess: | |
| 102 | | | Debug.Log("login success!"); | |
| 103 | | | state = ClientState.LoginSuccess; | |
| 104 | | | **break**; | |
| 105 | | | **case** (**byte**)OpCodeEnum.LoginFailed: | |
| 106 | | | Debug.Log("login Failed!"); | |
| 107 | | | state = ClientState.LoginFailed; | |
| 108 | | | StartCoroutine(loginFailedHandle()); | |
| 109 | | | **break**; | |
| 110 | | | } | |
| 111 | | | } | |
| 112 | | | **public** **void** OnStatusChanged (StatusCode statusCode) | |
| 113 | | | {*//Unity3D教程手册：www.unitymanual.com* | |
| 114 | | | **switch**(statusCode){ | |
| 115 | | | **case** StatusCode.Connect: | |
| 116 | | | Debug.Log("Connect Success! Time:"+Time.time); | |
| 117 | | | state = ClientState.Connected; | |
| 118 | | | **break**; | |
| 119 | | | **case** StatusCode.Disconnect: | |
| 120 | | | state = ClientState.ConnectFailed; | |
| 121 | | | StartCoroutine(connectFailedHandle()); | |
| 122 | | | Debug.Log("Disconnect! Time:"+Time.time); | |
| 123 | | | **break**; | |
| 124 | | | } | |
| 125 | | | } | |
| 126 | | | **public** **void** OnEvent (EventData eventData) | |
| 127 | | | { | |
| 128 | | | } | |
| 129 | | | #endregion | |
| 130 | | | } | |
| 131 | | | **public** **enum** ClientState :**byte**{ | |
| 132 | | | DisConnect, | |
| 133 | | | Connecting, | |
| 134 | | | Connected, | |
| 135 | | | ConnectFailed, | |
| 136 | | | LoginSuccess, | |
| 137 | | | LoginFailed | |
| 138 | | | } | |
| 139 | | | **public** **enum** OpCodeEnum :**byte** | |
| 140 | | | { | |
| 141 | | | *//login* | |
| 142 | | | Login = 249, | |
| 143 | | | LoginSuccess = 248, | |
| 144 | | | LoginFailed = 247, | |
| 145 | | | *//room* | |
| 146 | | | Create = 250, | |
| 147 | | | **Join** = 255, | |
| 148 | | | Leave = 254, | |
| 149 | | | RaiseEvent = 253, | |
| 150 | | | SetProperties = 252, | |
| 151 | | | GetProperties = 251 | |
| 152 | | | } | |
| 153 | | | **public** **enum** OpKeyEnum :**byte** | |
| 154 | | | { | |
| 155 | | | RoomId = 251, | |
| 156 | | | UserName = 252, | |
| 157 | | | PassWord = 253 | |
| 158 | | | } | |
|  |  | |  |

|  |  |
| --- | --- |
| 001 | **using** UnityEngine; |
| 002 | **using** System.Collections; |
| 003 | **using** System.Collections.Generic; |
| 004 |  |
| 005 | **using** ExitGames.Client.Photon; |
| 006 |  |
| 007 |  |
| 008 |  |
| 009 |  |
| 010 |  |
| 011 | **public** **class** TestConnection : MonoBehaviour,IPhotonPeerListener { |
| 012 | **public** **string** address; |
| 013 |  |
| 014 | PhotonPeer peer; |
| 015 | ClientState state = ClientState.DisConnect; |
| 016 |  |
| 017 | **string** username = ""; |
| 018 | **string** password = ""; |
| 019 | *// Use this for initialization* |
| 020 | **void** Start () { |
| 021 | peer = new PhotonPeer(**this**,ConnectionProtocol.Udp); |
| 022 | } |
| 023 |  |
| 024 | *// Update is called once per frame* |
| 025 | **void** Update () { |
| 026 | peer.Service(); |
| 027 | } |
| 028 |  |
| 029 | **public** GUISkin skin; |
| 030 | **void** OnGUI(){ |
| 031 | GUI.skin = skin; |
| 032 |  |
| 033 | **switch**(state){ |
| 034 | **case** ClientState.DisConnect: |
| 035 | GUI.Label(new Rect(Screen.width/2,Screen.height/2-30,300,40),"click the button to connect."); |
| 036 | **if**(GUI.Button(new Rect(Screen.width/2,Screen.height/2,100,30),"Connect")){ |
| 037 | peer.Connect(address,"MyServer"); |
| 038 | state = ClientState.Connecting; |
| 039 | } |
| 040 | **break**; |
| 041 | **case** ClientState.Connecting: |
| 042 | GUI.Label(new Rect(Screen.width/2,Screen.height/2-30,300,40),"Connecting to Server..."); |
| 043 | **break**; |
| 044 | **case** ClientState.Connected: |
| 045 | GUILayout.BeginArea(new Rect((Screen.width-500)/2,(Screen.height-400)/2,500,400)); |
| 046 |  |
| 047 | *//* |
| 048 | GUILayout.BeginVertical(); |
| 049 |  |
| 050 | GUILayout.Label("Connect Success! Please Login."); |
| 051 | *//draw username* |
| 052 | GUILayout.BeginHorizontal(); |
| 053 | GUILayout.Label("UserName:"); |
| 054 | username = GUILayout.TextField(username); |
| 055 | GUILayout.EndVertical(); |
| 056 |  |
| 057 | *//draw password* |
| 058 | GUILayout.BeginHorizontal(); |
| 059 | GUILayout.Label("Password:"); |
| 060 | password = GUILayout.TextField(password); |
| 061 | GUILayout.EndVertical(); |
| 062 |  |
| 063 | *//draw buttons* |
| 064 | GUILayout.BeginHorizontal(); |
| 065 | **if**(GUILayout.Button("login")){ |
| 066 | userLogin(username,password); |
| 067 | } |
| 068 |  |
| 069 |  |
| 070 | **if**(GUILayout.Button("canel")){ |
| 071 | state = ClientState.DisConnect; |
| 072 | } |
| 073 | GUILayout.EndVertical(); |
| 074 |  |
| 075 | GUILayout.EndVertical(); |
| 076 | GUILayout.EndArea(); |
| 077 |  |
| 078 | **break**; |
| 079 | **case** ClientState.ConnectFailed: |
| 080 | GUI.Label(new Rect(Screen.width/2,Screen.height/2-30,300,40),"Connect Failed."); |
| 081 |  |
| 082 | **break**; |
| 083 | **case** ClientState.LoginSuccess: |
| 084 | GUILayout.BeginArea(new Rect((Screen.width-500)/2,(Screen.height-400)/2,500,400)); |
| 085 | GUILayout.Label("Login Success!"); |
| 086 | GUILayout.EndArea(); |
| 087 | **break**; |
| 088 | **case** ClientState.LoginFailed: |
| 089 | GUILayout.BeginArea(new Rect((Screen.width-500)/2,(Screen.height-400)/2,500,400)); |
| 090 | GUILayout.Label("Login Failed!"); |
| 091 | GUILayout.EndArea(); |
| 092 | **break**; |
| 093 | } |
| 094 | } |
| 095 |  |
| 096 | #region My Method |
| 097 | IEnumerator connectFailedHandle(){ |
| 098 | **yield** **return** new WaitForSeconds(1); |
| 099 | state = ClientState.DisConnect; |
| 100 | } |
| 101 |  |
| 102 | **void** userLogin(**string** uname,**string** pwd){ |
| 103 | Debug.Log("userLogin"); |
| 104 | Dictionary<**byte**,**object**> param = new Dictionary<**byte**, **object**>(); |
| 105 | param.**Add**((**byte**)OpKeyEnum.UserName,uname); |
| 106 | param.**Add**((**byte**)OpKeyEnum.PassWord,pwd); |
| 107 | peer.OpCustom((**byte**)OpCodeEnum.Login,param,**true**); |
| 108 | } |
| 109 |  |
| 110 | IEnumerator loginFailedHandle(){ |
| 111 | **yield** **return** new WaitForSeconds(1); |
| 112 | Debug.Log("loginFailedHandle"); |
| 113 | state = ClientState.Connected; |
| 114 | } |
| 115 | #endregion |
| 116 |  |
| 117 |  |
| 118 |  |
| 119 |  |
| 120 | #region IPhotonPeerListener implementation |
| 121 | **public** **void** DebugReturn (DebugLevel level, **string** message) |
| 122 | { |
| 123 |  |
| 124 | } |
| 125 |  |
| 126 | **public** **void** OnOperationResponse (OperationResponse operationResponse) |
| 127 | { |
| 128 | **switch**(operationResponse.OperationCode) |
| 129 | { |
| 130 | **case** (**byte**)OpCodeEnum.LoginSuccess: |
| 131 | Debug.Log("login success!"); |
| 132 | state = ClientState.LoginSuccess; |
| 133 | **break**; |
| 134 | **case** (**byte**)OpCodeEnum.LoginFailed: |
| 135 | Debug.Log("login Failed!"); |
| 136 | state = ClientState.LoginFailed; |
| 137 | StartCoroutine(loginFailedHandle()); |
| 138 | **break**; |
| 139 | } |
| 140 | } |
| 141 |  |
| 142 | **public** **void** OnStatusChanged (StatusCode statusCode) |
| 143 | { |
| 144 | **switch**(statusCode){ |
| 145 | **case** StatusCode.Connect: |
| 146 | Debug.Log("Connect Success! Time:"+Time.time); |
| 147 | state = ClientState.Connected; |
| 148 | **break**; |
| 149 | **case** StatusCode.Disconnect: |
| 150 | state = ClientState.ConnectFailed; |
| 151 | StartCoroutine(connectFailedHandle()); |
| 152 | Debug.Log("Disconnect! Time:"+Time.time); |
| 153 | **break**; |
| 154 | } |
| 155 | } |
| 156 |  |
| 157 | **public** **void** OnEvent (EventData eventData) |
| 158 | { |
| 159 |  |
| 160 | } |
| 161 | #endregion |
| 162 | } |
| 163 |  |
| 164 | **public** **enum** ClientState : **byte**{ |
| 165 | DisConnect, |
| 166 | Connecting, |
| 167 | Connected, |
| 168 | ConnectFailed, |
| 169 | LoginSuccess, |
| 170 | LoginFailed |
| 171 | } |
| 172 |  |
| 173 | **public** **enum** OpCodeEnum : **byte** |
| 174 | { |
| 175 | *//login* |
| 176 | Login = 249, |
| 177 | LoginSuccess = 248, |
| 178 | LoginFailed = 247, |
| 179 |  |
| 180 | *//room* |
| 181 | Create = 250, |
| 182 | **Join** = 255, |
| 183 | Leave = 254, |
| 184 | RaiseEvent = 253, |
| 185 | SetProperties = 252, |
| 186 | GetProperties = 251 |
| 187 | } |
| 188 |  |
| 189 |  |
| 190 | **public** **enum** OpKeyEnum : **byte** |
| 191 | { |
| 192 | RoomId = 251, |
| 193 | UserName = 252, |
| 194 | PassWord = 253 |
| 195 | } |