

uexSocketMgr

socket 功能

方法

createUDPSocket 创建 UDP 对象

createTCPSocket 创建 TCP 对象

closeSocket 关闭 Socket 对象

setTimeout 设置超时时间

setInetAddressAndPort 设置接收方的 IP 地址和端口

sendData 发送数据

回调方法

cbCreateUDPSocket 创建 UDP 对象的回调方法

cbCreateTCPSocket 创建 TCP 对象的回调方法

cbSendData 发送数据的回调方法

cbConnected 设置接收方的 IP 地址和端口的回调方法

监听方法

onData 接收数据的监听方法

onDisconnected 连接断开的监听方法

createUDPSocket

创建 UDP 对象

uexSocketMgr.createUDPSocket(id, port, type)

说明：

创建 UDP 对象，同一 id 只能被创建一次

参数：

id: (*Number 类型*) 必选 唯一标识符

port: (*Number 类型*) 必选 本地端口号

type: (*Number 类型*) 可选 数据类型：0-String；1-Base64 数据格式。默认为 0

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<title>socket 功能</title>
```

```
<script type="text/javascript">
```

```
var socketUDP = 1;
```

```
var socketTCP = 2;
```

```

function creatUDP(port){
    uexSocketMgr.createUDPSocket(socketUDP,port);
}
window.uexOnload = function(){
    uexSocketMgr.cbCreateUDPSocket = function(opCode,dataType,data){
        if(data == 0){
            alert("创建 UDP 成功");
        }else{
            alert("创建 UDP 失败");
        }
    }
}
</script>
</head>
<body>
    <span>绑定本地的 UDP 端口号: </span>
    <input class="textbox" type="text" id = "localPort" value="45666" >
    <input      class="btn"      type="button"      value="      创      建      UDP"
onclick="creatUDP(document.getElementById('localPort').value);">
</body>
</html>

```

createTCPSocket

创建 TCP 对象

uexSocketMgr.createTCPSocket(id, type)

说明:

创建 TCP 对象, 同一 id 只能被创建一次

参数:

id: (*Number* 类型) 必选 唯一标识符

type: (*Number* 类型) 可选 数据类型: 0-String; 1-Base64 数据格式。默认为 0

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

```

<!DOCTYPE HTML>
<html>
<head>
<title>socket 功能</title>
<script type="text/javascript">

```

```

var socketUDP = 1;
var socketTCP = 2;
function creatTCP(){
    tcp = uexSocketMgr.createTCPSocket(socketTCP);
}
window.uexOnload = function(){
    uexSocketMgr.cbCreateTCPSocket = function(opCode,dataType,data){
        if(data == 0){
            alert("创建 TCP 成功");
        }else{
            alert("创建 TCP 失败");
        }
    }
}
</script>
</head>
<body>
    <input class="btn" type="button" value="创建 TCP" onclick="creatTCP();">
</body>
</html>

```

closeSocket

关闭 Socket 对象

uexSocketMgr.closeSocket(id)

说明：

关闭 Socket 对象

参数：

id: (*Number 类型*)必选 唯一标识符

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

```

<!DOCTYPE HTML>
<html>
<head>
<title>socket 功能</title>
<script type="text/javascript">
    var socketUDP = 1;
    var socketTCP = 2;
    function closeUDP(){

```

```

        uexSocketMgr.closeSocket(socketUDP);
    }
    function closeTCP(){
        uexSocketMgr.closeSocket(socketTCP);
    }
</script>
</head>
<body>
    <input class="btn" type="button" value="关闭 UDP" onclick="closeUDP();">
    <input class="btn" type="button" value="关闭 TCP" onclick="closeTCP();">
</body>
</html>

```

setTimeout

设置超时时间

uexSocketMgr.setTimeout(id, timeOut)

说明：

设置超时时间

参数：

id: (*Number 类型*) 必选 唯一标识符

serverURL: (*Number 类型*) 必选 超时时间

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

setInetAddressAndPort

设置接收方的 IP 地址和端口

uexSocketMgr.setInetAddressAndPort(id, ip, port)

说明：

设置接收方的 IP 地址和端口

参数：

id: (*Number 类型*) 必选 唯一标识符

ip: (*String 类型*) 必选 接收方 IP 地址

port: (*Number 类型*) 必选 接收方端口

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例:

```
<!DOCTYPE HTML>
<html>
<head>
<title>socket 功能</title>
<script type="text/javascript">
    var socketUDP = 1;
    var socketTCP = 2;
    function setUDPIPandPort(ip,port){
        uexSocketMgr.setInetAddressAndPort(socketUDP,ip,port);
    }
    function setTCPIPandPort(ip,port){
        uexSocketMgr.setInetAddressAndPort(socketTCP,ip,port);
    }
    window.uexOnload = function(){
        uexSocketMgr.cbConnected = function(opCode,dataType,data){
            if(data == 0){
                alert("opCode:" + "设置对方的 ip 和端口成功");
            }else{
                alert("opCode:" + "设置对方的 ip 和端口失败");
            }
        }
    }
</script>
</head>
<body>
    <span>对方的 ip: </span>
    <input class="textbox" type="text" id = "udpId" value="192.168.1.255" >
    <span>对方的 port: </span>
    <input class="textbox" type="text" id = "udpPort" value="3001">
    <input class="btn" type="button" value=" 设 置 ip 和 port"
onclick="setUDPIPandPort(document.getElementById('udpId').value,document.getElementById('
udpPort').value);">
    <span>对方的 ip: </span>
    <input class="textbox" type="text" id = "tcpId" value="192.168.1.28" >
    <span>对方的 port: </span>
    <input class="textbox" type="text" id = "tcpPort" value="2009">
    <input class="btn" type="button" value=" 设 置 ip 和 port"
onclick="setTCPIPandPort(document.getElementById('tcpId').value,document.getElementById('t
cpPort').value);">
</body>
</html>
```

sendData

发送数据

uexSocketMgr.sendData(id, msg)

说明：

发送数据

参数：

id: (*Number* 类型)必选 上传对象的唯一标识符

msg: (*String* 类型)必选 要发送的数据

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<title>socket 功能</title>
```

```
<script type="text/javascript">
```

```
    var socketUDP = 1;
```

```
    var socketTCP = 2;
```

```
    function sendUDP(data){
```

```
        uexSocketMgr.sendData(socketUDP,data);
```

```
    }
```

```
    function sendTCP(data){
```

```
        uexSocketMgr.sendData(socketTCP,data);
```

```
    }
```

```
    window.uexOnload = function(){
```

```
        uexSocketMgr.onData = function(opCode,text){
```

```
            if(opCode == socketUDP){
```

```
                document.getElementById('getUDPData').innerHTML = text;
```

```
            }
```

```
            if(opCode == socketTCP){
```

```
                document.getElementById('getTCPData').innerHTML =text;
```

```
            }
```

```
        }
```

```
        uexSocketMgr.onDisconnected = function(opCode,text){
```

```
            if(opCode == socketUDP){
```

```
                alert('UDP 连接断开');
```

```
            }
```

```
            if(opCode == socketTCP){
```

```
                alert('TCP 连接断开');
```

```

    }
}
uexSocketMgr.cbSendData = function(opCode,dataType,data){
    if(opCode == socketUDP){
        if(data == 0){
            alert("UDP 发送成功");
        }else{
            alert("UDP 发送失败");
        }
    }
    if(opCode == socketTCP){
        if(data == 0){
            alert("TCP 发送成功");
        }else{
            alert("TCP 发送失败");
        }
    }
}
}
</script>
</head>
<body>
    <span>输入要发送的 UDP 数据: </span>
    <textarea class="tcxx" id="udpData">发送给 UDP 服务器的数据</textarea>
    <input class="btn" type="button" value="发 送 UDP"
onclick="sendUDP(document.getElementById('udpData').value);">
    <span>接收到的 UDP 数据为: </span>
    <div class="tcxx" id="getUDPData"></div>
    <input class="btn" type="button" value="关闭 UDP" onclick="closeUDP();">
    <span>输入要发送的 TCP 数据: </span>
    <textarea class="tcxx" id="tcpData">发送给 TCP 服务器的数据</textarea>
    <input class="btn" type="button" value="发 送 TCP"
onclick="sendTCP(document.getElementById('tcpData').value);">
    <span>接收到的 TCP 数据为: </span>
    <div class="tcxx" id="getTCPData"></div>
    <input class="btn" type="button" value="关闭 TCP" onclick="closeTCP();">
</body>
</html>

```

cbCreateUDPSocket

创建 UDP 对象的回调方法

uexSocketMgr.cbCreateUDPSocket(opId, dataType, data)

参数:

opId: (*Number 类型*)必选 UDP 对象的唯一标识符

dataType: (*Number 类型*)必选 参数类型 详见 CONSTANT 中 Callback 方法数据类型

data: (*Number 类型*)必选 返回 uex.cSuccess 或者 uex.cFailed , 详见 CONSTANT 中

Callback int 类型数据

版本支持:

3.0.0+

cbCreateTCPSocket

创建 TCP 对象的回调方法

uexSocketMgr.cbCreateTCPSocket(opId, dataType, data)

参数:

opId: (*Number 类型*)必选 TCP 对象的唯一标识符

dataType: (*Number 类型*)必选 参数类型 详见 CONSTANT 中 Callback 方法数据类型

data: (*Number 类型*)必选 返回 uex.cSuccess 或者 uex.cFailed , 详见 CONSTANT 中

Callback int 类型数据

版本支持:

3.0.0+

cbSendData

发送数据的回调方法

uexSocketMgr.cbSendData(opId, dataType, data)

参数:

opId: (*Number 类型*)必选 Socket 对象的唯一标识符

dataType: (*Number 类型*)必选 参数类型 详见 CONSTANT 中 Callback 方法数据类型

data: (*Number 类型*)必选 返回 uex.cSuccess 或者 uex.cFailed , 详见 CONSTANT 中

Callback int 类型数据

版本支持:

3.0.0+

cbConnected

设置接收方的 IP 地址和端口的回调方法

uexSocketMgr.cbConnected(opId, dataType, data)

参数:

opId: (*Number 类型*)必选 Socket 对象的唯一标识符

dataType: (*Number 类型*)必选 参数类型 详见 CONSTANT 中 Callback 方法数据类型

data: (*Number 类型*)必选 返回 uex.cSuccess 或者 uex.cFailed , 详见 CONSTANT 中

Callback int 类型数据

版本支持:

3.0.0+

onData

接收数据的监听方法

uexSocketMgr.onData(opId, data)

参数:

opId: (*Number 类型*)必选 Socket 对象的唯一标识符

data: (*String 类型*)必选 接收到的数据, json 字符串, 格式为:

{host:192.168.1.888, port:9999, data:接收到的消息}

各字段含义如下:

参数	是否必须	说明
host	是	对方 IP 地址
port	是	对方端口
data	是	接受到得消息

版本支持:

3.0.0+

onDisconnected

连接断开的监听方法

uexSocketMgr.onDisconnected(opId)

参数:

opId: (*Number 类型*)必选 Socket 对象的唯一标识符

版本支持:

3.0.0+