

GoEasy Presence Reference

一、用途

1. 订阅客户端上下线实时状态变化
2. 获取当前在线客户数量和在线客户列表

二、操作步骤

1. 注册 userId（如果不需要在线用户列表，可以跳过）

```
var goeasy = new GoEasy({  
  
    appkey: '{appkey}',  
  
    userId: '{user.uuid}',  
  
    userData: '{用户附加信息, 字符串类型, 不超过 300K}'  
  
});
```

2. 订阅用户在线状态

2.1 示例代码

```
goeasy.subscribePresence({  
  
    channel: 'channel_A',  
  
    onPresence: function (presenceEvents) {
```

//返回值 presenceEvents 示例

```
{  
  "clientAmount": 2, //在线客户端数量  
  "userAmount": 1, //在线用户数量  
  "events": [  
    {  
      "clientAmount": 1,  
      "userAmount": 1,  
      "events": [  
        {  
          "action": "join",  
          "userId": "user_id_1", //用户 id  
          "userData": "derek"    //用户附加信息  
        },  
        {  
          "action": "offline",  
          "userId": "user_id_2",
```

```
        "userData": "Jam"
    }
]
};
]
}
}
});
```

2.2 action 说明:

Join: 客户端订阅 channel 成功
leave: 客户端取消 channel 成功
offline: 客户端下线或断网
online: 之前订阅成功的客户端，在断网后，重连成功

2.3 相关事件监听:

```
goeasy.subscribePresence({
    channel: 'channel_A',
```

```
//订阅 Presence 成功

onSuccess: function () {

    alert('Presence subscription success.');
```

},

```
//订阅 Presence 失败

onFailed: function (error) {

    alert('Presence subscription failure.');
```

},

```
//当有客户端状态发生变化

onPresence: function (presenceEvents) {

    var data = JSON.stringify(presenceEvents)

    alert('Presence:' + data );

}

});
```

2.4 取消订阅用户在线状态

```
goeasy.unsubscribePresence({
    channel: 'demo_channel',
```

```
//取消成功
onSuccess: function () {
    alert("取消订阅 presence 成功.")
},
```

```
//取消失败
onFailed: function (error) {
    alert("取消订阅 presence 失败.")
}
```

```
});
```

3. 查询当前在线用户

3.1 Javascript

```
goeasy.hereNow(
    {
        channels: ['channel_A', "channel_B"], //要查询的 channel 列表

        includeUsers: true, //可选项, 是否返回用户列表, 默认 false

        distinct: true      //可选项, 是否过滤相同的 userId, 默认 false
    },
    function (response) {
        //response 示例
```

```
{  
  
  "code": 200,  
  
  "content": {  
  
    "channels": {  
  
      "channel_A": {  
  
        "channel": "channel_A",  
  
        "clientAmount": 1, //在线客户端数量  
  
        "userAmount": 1,  //在线用户数量  
  
        "users": [          //在线用户列表  
  
          {  
  
            "id": "user_id_1", //用户 id  
  
            "data": "derek"    //用户附加信息  
  
          }  
  
        ]  
  
      },  
  
      "channel_B": {  
  
        "channel": "channel_B",
```

```

        "clientAmount": 0,

        "userAmount": 0,

        "users": []

    }

}

}

}

}

);

```

3.2 Rest API

Request:

`http://<Rest Host>/presence?appkey={appkey}&channel={channel}&includeusers={includeusers}&distinct={distinct}`

参数:

channel //必须项，一个请求可以包含多个 channel
includeusers //是否返回用户列表，可选项，默认 false
distinct //是否过滤相同的 userId，可选项，默认 false

Response JSON:

//response 示例

```
{
  "code": 200,
  "content": {
    "channels": {
      "channel_A": {
        "channel": "channel_A",
        "clientAmount": 1, //在线客户端数量
        "userAmount": 1,  //在线用户数量
        "users": [        //在线用户列表
          {
            "id": "user_id_1", //用户 id
            "data": "derek"    //用户附加信息
          }
        ]
      },
    },
  },
}
```



```

        "channel_B": {
            "channel": "channel_B",
            "clientAmount": 0,
            "userAmount": 0,
            "users": []
        }
    }
}

```

3.3 Java

升级 Java sdk 至 0.3.10

```

goEasy.presence(Arrays.asList("channel_A", "channel_B"),

    false, // 是否包含用户

    false, // 是否过滤重复 userId

    new PresenceListener() {

        @Override

        public void onSuccess(Presences presences) {

```

//presences 示例

```
{  
  "channels": {  
    "channel_A": {  
      "channel": "channel_A",  
      "clientAmount": 1, //在线客户端数量  
      "userAmount": 1,  //在线用户数量  
      "users": [        //在线用户列表  
        {  
          "id": "user_id_1", //用户 id  
          "data": "derek"    //用户附加信息  
        }  
      ]  
    },  
    "channel_B": {  
      "channel": "channel_B",  
      "clientAmount": 0,  

```

```
        "userAmount": 0,

        "users": []

    }

}

}

}

    public void onFailed(GoEasyError error) {

        throw new UnsupportedOperationException("Not yet implemented");

    }

});
```

三、资费

1. 服务器每推送一条 presenceEvents，扣减消息量 1 条
2. 每个客户端收到一条一条 presenceEvents 扣减消息量 1 条
3. 每查询一次当前在线用户，减扣消息量 1 条