# How to Get Kiosk “Call for Help” events

**Server IP Address: 10.65.3.104 (Web layer)**

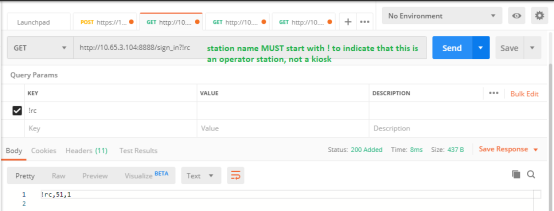
**Default Server Port: 8888**

**HTTP Method: GET**

**APIs:**

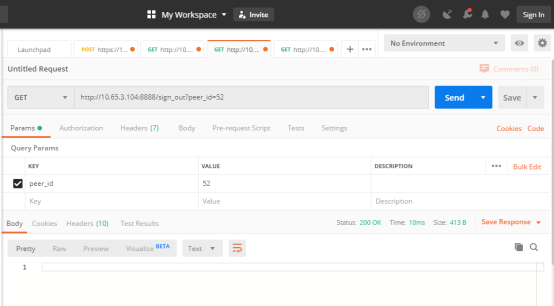
|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | sign\_in | **URI** | / sign\_in?[name] |
| **Param** | **Value** | **Description** | |
| name | string | User name identifier used to login to WebRTC STUN server | |
| **Request Sample** | | <http://10.65.3.104:8888/sign_in?!octopus> | |
| **Explain** | | * Used for signing in to WebRTC STUN server * User name must start with “**!**”an exclamation sign to differentiate it from a kiosk | |
| **Response Sample** | | !octopus,3,1 | |
| **Response Format** | | Name, peer\_id, status ( split by “,” )   * **name**: User name used for sign in * **peer\_id**: Session ID identifier * **status**: 1 means client successfully connected to the server,   0 means client failed to connect / disconnected to the  server | |

**Login Success**

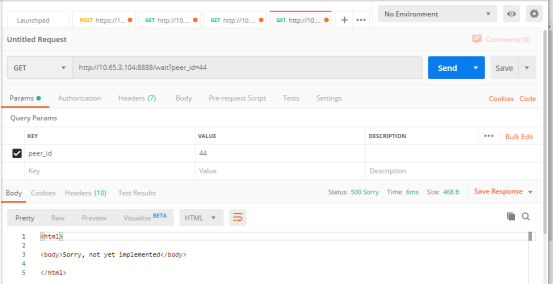


|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | sign\_out | **URI** | / sign\_out?peer\_id=[peer\_id] |
| **Param** | **Value** | **Description** | |
| peer\_id | integer | This function is used for disconnecting from WebRTC STUN server. | |
| **Request Sample** | | <http://10.65.3.104:8888/sign_out?peer_id=3> | |
| **Explain** | | * Sign out from WebRTC STUN server | |
| **Response Sample** | | No response. | |
| **Response Format** | | No response. | |

**Sign out Success**

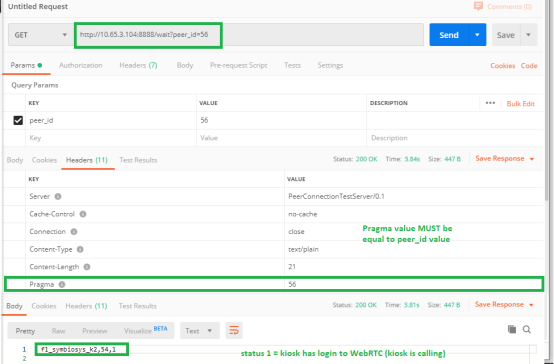


**Sign out Fail, due to incorrect peer id**

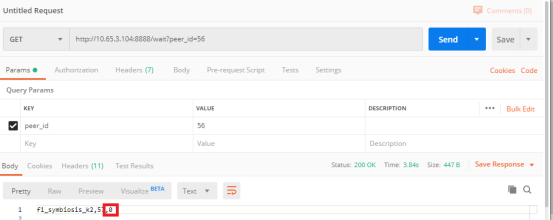


|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | wait | **URI** | / wait?peer\_id=[id] |
| **Param** | **Value** | **Description** | |
| Peer\_id | integer | This function is used to indicate the WebRTC STUN server that a client (operator station or a kiosk) has connected / disconnected.  In the case of kiosks, a WebRTC connection event can be interpreted as a “kiosk calling” event | |
| **Request Sample** | | <http://10.65.3104:8888/wait?peer_id=3> | |
| **Explain** | | * This function is used to received message kiosk sign in / sign out notifications | |
| **Response Sample** | | Bio\_centros\_k1 , 6 , 1 | |
| **Response Header** | | Pragma: [peer\_id] | |
| **Response Format** | | If pragma response header equals your own peer\_id value, it means a kiosk login/logout message (kiosk calling, kiosk hang up event)  Otherwise, if pragma header value equals to a different peer\_id value, ignore the message  Name, peer\_id, status (split by “,”)   * **name**: User name which just signed in or out, ignore any client whose name starts with ! * **peer\_id**: client seesion id * **status**: 1 means client login/connected (aka kiosk calling)   0 means client logout/disconnect  If there’s more than one kiosk calling ,it will be displayed as a new line | |

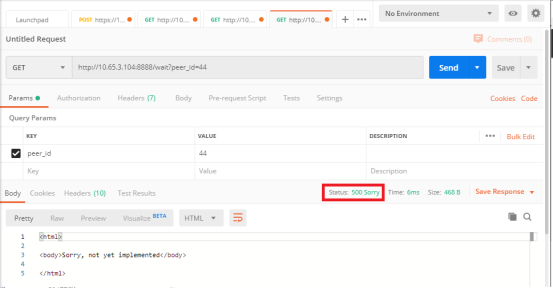
**Kiosk Calling Event (status = 1 && pragma = peer) OK**



**Kiosk logout Event (Status =0 ) , IGNORE**



**Kiosk Calling Event Fail due to incorrect peer id**



# Listen to Kiosk Call for Help Events

**Step 1: Send CGI to Sign in to WebRTC STUN server**

* Call *http://[ip]:[port]/sign\_in?[!username]  
  (username must start with ! )*   
  (e.g. : http://10.65.3.104:8888/sign\_in?!octopus)
* Receive sign in success response (e.g.: !octopus,3,1)
* Save peer id from response to local variable ( e.g. : 3 )

**Step 2: Continuously call Wait CGI to monitor any kiosks connecting to the WebRTC server**

* Call *http://[ip]:[port]/wait?peer\_id=[id]*(e.g. http://10.65.3.104:8888/wait?peer\_id=3)
* Wait for response (a successful response or a timeout)
  + If no response , the CGI will timeout, call the wait CGI right after
  + If there’s a message, check that:
    1. If header->pragma value equals to peer\_id value  
       (e.g. pragma: 3)
* check the body message (e.g. kiosk\_1,8,1)
* If the client name starts with “!”, skip it and call “wait” API again.
* If the status equals 0, skip it and call “wait” API again
* Else if client name doesn’t start with ! and status is 1 , it means this kiosk click call for help button. (p.s. name = kiosk\_1)
* Call wait API again
  + 1. If header 🡪 pragma is different to own peer\_id, skip this message and call wait API again.

**Step 3 : Send CGI to Sign out from WebRTC STUN Server**

* Call *http://[ip]:[port]/sign\_out?peer\_id=[id]*(e.g. http://10.65.3.104:8888/sign\_out?peer\_id=3)