**MibewMob (Mibew Mobile) Client Interface Documentation for v 1.6.x**

In order to use the Mibew Mobile app against a Mibew 1.6.x installation, a new interface was created for the app to communicate with the server. This can be found in the <mibew>/mobile folder. Below is a brief description of the interface and its parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Command**  **(cmd)** | **Req Type** | **Parameters** | **Notes** |
| isalive | GET | * apiver | e.g cmd=isalive&apiver=1003 |
| login | GET | * username * password * deviceuuid |  |
| logout | GET | * oprtoken |  |
| visitorlist | GET | * oprtoken * activevisitors * stealth | activevisitors is a JSON array of active chatthread ids that have been sync’ed to the device .  This returns a list of threads that have changed since the last sync to the device. |
| visitornotification | GET | * oprtoken * activevisitors * stealth | This returns true if there is a change of thread state since the last sync to the device. It is basically the same logic as visitorlist, only that the list is not returned. |
| startchat | GET | * oprtoken * threadid * token * viewonly * force |  |
| newmessages | GET | * oprtoken * threadid * token * typed | Retrieves new messages that have not yet been sync’ed to the device |
| syncserveroperator | GET | * oprtoken | Retrieves the server and operator info for sync’ing to the device |
| synccannedmessages | GET | * oprtoken * cannedmsghashes | cannedmsghashes is a semi-colon delimited list of chatresponse ids and CRC hashes, e.g 1,CRC1;2,CRC2;6,CRC3 |
| postmessage | POST | * oprtoken * threadid * token * messageidl * message | messageidl (l for local) is the unique messageid from the device database. It is useful for sync’ing |
| closethread | POST | * oprtoken * threadid |  |
| ack-messages | POST | * oprtoken * messageids | messageids is a comma-separated list of server messageids. This is to acknowledge that the messages where received by the device |

Other Noteworthy Points

1. All responses are JSON-encoded.
2. When a POST request is issued, the parameters are not passed as request parameters with the URL. Instead, the request parameters are passed in the body of the request as JSON objects. For example, the request for ack-messages will be:

{

"cmd":"ack-messages",

"oprtoken":"ABCDEF",

"messageids":"7,8,9,10,11"

}

1. The “Content-Type” header of the POST request should be set to “application/json”