Introduction

The api is accessed via websocket using UDP datagrams. An initial connection *handshake* command must be made before the software will accept further commands. Once connection is established further handshakes are not neccessary until the *closeSocket* command is sent or the software is restarted.

Datagram format

The command datagram is composed of a JSON string with three elements: command, amount, data.

1. **Command:** A string containing the action to perform.
2. **Amount:** An integer, used in increment and set actions.
3. **Data:** A string containing the name of a template or file to recall. Not currently used.

```
{"command":"handshake","amount":0,"data":""}
```

The command datagrams employ AES BCM PKCS#7 encryption and are base64 encoded. Likewise, any messages from the software will be likewise encoded. The encryption key is provided in the scoreboard 'Web Interface' settings.

On successful connection, rather than a JSON string the software will return the plaintext identifier of which sport it is running. eg. "BasketBall", "Hockey"

Commands

[Amount] is the value of amount in the command message.
[TEAM] is one of two values; either 'home' or 'away'.

All commands receive no acknowledgement except *handshake* and *settingsPort*.

Command	Amount
handshake	Open connection to the scoreboard.
closeSocket	End connection to the scoreboard.
playHorn	Sound the short horn.
clockStartStop	Start main clock timer.
mainClockSet	Total time, in seconds, to set the main clock.
shotClockSet	Total time, in seconds, to set the shot clock.

Command	Amount
shotClockStartStop	Toggle Shot Clock.
shotClockReset	Set shotclock to game default.
shotClockResetShort	Set shotclock to the 1/2 game default.
resetTimer	Reset main & shot clock to the game defaults.
clockReset	Reset main & shot clock to the game defaults.
gameReset	Reset all counters and times to game defaults.
cancelTimeOut	Cancel any running TimeOuts.
showWorldClock	Show the real world time in the game clock.
showWorldClock24	Show the real world time, 24 hour, in the game clock.

Command	Amount
[TEAM]Score	Updates [TEAM] score by [Amount].
[TEAM]ScoreReset	Set [TEAM] score to 0.
[TEAM]ScoreSet	Set [TEAM] score to [Amount].
[TEAM]Fouls	Update [TEAM] score by [Amount].
[TEAM]FoulsReset	Set [TEAM] fouls to 0.
[TEAM]FoultsSet	Set [TEAM] fouls to [Amount].
[TEAM]TimeOuts	Update [TEAM] TOL by [Amount].
[TEAM]TimeOutsReset	Set [TEAM] TOL to game default.
[TEAM]TimeOutsReset	Set [TEAM] TOL to [Amount].
[TEAM]TimeOut	If there is a TOL remaining, trigger a TimeOut for the [TEAM] team.

Command	Amount
period	Update Period by [Amount].
periodReset	Set Period to 1.
periodSet	Set Period to [Amount].
possLeft	Set Possession marker to Left.
possRight	Set Possession marker to Right.
possClear	Set Possession marker to None.

Game Data

Once a successful connection has been made to the scoreboard, if 'Output data' is enabled in the Web Interface settings, the scoreboard will begin sending UDP datagram containing game status to the address that made the initial connection. To prevent this, or to end this stream you can send the *closeSocket* command. After closing the socket you will need to send the *handshake* command again if you wish to send further commands.

Game status datagrams are returned as a JSON string containing objects for game, home, and away.

```
{
    "game":{
        "clockLeft":"12",
        "clockRight":"00",
        "clockMiddle":":",
        "period":"1",
        "shotClock":"24.00",
        "timeOutClock":"30",
        "timeOut":false,
        "clockRunning":false,
        "shotClockRunning":false,
        "shotClockDisabled":false
        },
    "away":{
        "name":"AWAY",
        "score":"00",
        "fouls":"0",
        "tol":"3",
        "poss":"false"
        },
```

```json
    "home":{
        "name":"HOME",
        "score":"02",
        "fouls":"0",
        "tol":"3",
        "poss":"false"
        }
    }
```