Instructions for Connecting to the VAST Challenge 2014 Mini-Challenge 3 Streaming Data Server

Last updated April 30, 2014

# Overview

The VAST Challenge 2014 Mini-Challenge 3 data server will provide a Web Socket (<http://www.websocket.org/>) connection. Client libraries exist in multiple languages. Sample clients are provided for contestants in the following languages:

* Java
* Javascript
* node.js Javascript

There are four different datasets available:

**Segment 0** contains approximately 10 minutes of test data not related to the challenge. This exists simply to allow you to test your connection. You can connect and play this test data as many times as you want, but it should not be used for any of your analysis.

**Segment 1** contains the messages from 1700 to 1830 on January 23.

**Segment 2** contains the messages from 1830 to 2000 on January 23.

**Segment 3** contains the messages from 2000 to shortly after 2130 on January 23.

## How to Connect

Contact [VASTChal2014MC3@vacommunity.org](mailto:VASTChal2014MC3@vacommunity.org) to obtain the name of the web server to connect to and the user ID to use.

Given a server url (ws://hostname:socket/), open a connection.

Once the connection is valid, the client must provide a JSON object with the user id and connection information. The format is as follows:

{ uid : “username”,

segment : XX,

reset : true | false }

The **uid** is the user ID value you will be sent in email.

The **segment** is a number 0-3. The segment number corresponds to the data segments described above.

The **reset** parameter can be set to *true* or *false*. If **reset** is *true*, then the stream will start from the beginning of the stream. If **reset** is *false,*then the stream will resume in approximately where it left off.

**Troubleshooting suggestions**

* This can be useful if the data stream is interrupted while in progress. Note that **reset** can be set to *true* for segment 0, 1 and 2. If you send a “reset:true” for segment 3, the stream will not play.
* A single **uid** may only have a single connection open – playing a single segment. If a second connection is opened for the same **uid** value, the second connection will over-ride and affect both connections.
* If running the javascript clients, you may be prompted if you want to reset. “OK” equates to “true” and “Cancel” equates to “false”.

# Data Format

When the system recognizes your username, you will receive a JSON object back that looks like

{ type : “control”,

body : [ {state :”OK”} ]

}

If your username is not recognized or has already played a stream that is limited, or you try to reset a stream that cannot be reset, the state will be “BAD”.

Immediately after the user id message, the data will start being sent from the server. No acknowledgement is required after a message.

Messages are JSON objects in the form:

{ “type” : “control | mbdata | ccdata”,

“body” : [

{ “date”:”YYYYMMDDHHMISS”, “author”:”authorname”, “message”:”microblog/command message”},

…..

]

}

The array contains 1-N messages blocks in each JSON object.

**Type**:

control – a control message relating to the health of the stream. Non content-bearing.

mbdata – “micro-blog” data

ccdata – emergency control center data

Each message contains appropriate fields for its type

Microblog data contains

**date** - YYYYMMDDHHMISS – 24 hour clock

**author** – the account name the message was posted from

**message** – text content of the message

**latitude** – [Optional] – latitude of the poster’s location

**longitude** - [Optional] - longitude of the poster’s location

Control Center data contains

**date** - YYYYMMDDHHMISS – 24 hour clock

**message** – the message sent to the control center

**location** – location information in string form