Firebase JMX

Firebase Version: 1.4

JMX 3
Client Registry 3
  Invocation Target 3
  Methods 3
    Get Number Of Clients 3
    Kick Player 3
    Kick Player From Table 3
    Get Remote Address 4
    Is Logged In 4
    Get Seated Tables 4
    Get Watching Tables 4
    Get Screenname 5
    Is Local 5
Server Instance 5
  Invocation Target 5
  Methods 5
    Add Node 5
    Remove Node 6
    Get Live Nodes 6

JMX

This document details the public JMX interfaces for a Firebase server.

Client Registry

Invocation Target

Object Name:	com.cubeia.firebase.clients:type=ClientRegistry
Interface:	com.cubeia.firebase.service.clientreg.state.StateClientRegistryMBean

Methods

Get Number Of Clients

Signature:

```
public int getNumberOfClients();
```

Returns the number of clients on the local server.

Kick Player

Signature:

```
public void kickPlayer(int playerId);
```

Kick the client with the given player id from the system. The client will be removed from all tables and the socket will be closed. The client will receive a Forced Logout Packet.

Note: This method only applies to the local client registry. This means that you need to invoke this method on the server where the client is logged in.

Kick Player From Table

Signature:

```
public void kickPlayerFromTable(int playerId, int tableId);
```

Kick a player from a table. The player will be kicked from the table only. The client will not be logged out or disconnected from the system.

Note: The player will be removed from the table regardless of where the session is located. However, the Kick Player Packet, which is a notification of the kick, will only be sent to client if the session in the same server as the invocation target.

Get Remote Address

Signature:

```
public String getRemoteAddressText(int clientId);
```

Returns the client's remote address in String format (SocketAddress.toString).

The method will return null if the client is not logged in (I.e. Not found in the client registry).

Note: This method only applies to the local client registry. This means that you need to invoke this method on the server where the client is logged in. If the client is not logged in locally a NullPointer will be thrown.

Is Logged In

Signature:

```
public boolean isLoggedIn(int clientId);
```

Returns true if the client with the given id has a session in on the system, this includes clients that are connected and have lost connections but are not yet reaped.

This lookup is system wide.

Get Seated Tables

Signature:

```
public Map<Integer, Integer> getSeatedTables(int playerId);
```

Returns the tables and seats the player is seated at.
The map will consist of `<tableid:seat>`

Works system wide.

Get Watching Tables

Signature:

```
public List<Integer> getWatchingTables(int playerId);
```

Returns a list of tables that the player is watching. The list will not contain tables the player is seated at, but only tables where the player is registered as a watcher.

Works system wide.

Get Screenname

Signature:

```
public String getScreenname(int clientId);
```

Returns the screenname of the client with the given client id or null if not found. The client must be logged in, if the client is not logged in then null will be returned.

Works system wide.

Is Local

Signature:

```
public boolean isLocal(int clientId);
```

Returns true if the client is managed by this local node. Returns false otherwise. The check includes client sessions currently connected and client sessions waiting for reconnect. (See is logged in).

The method only checks the local node so it is not a system wide call.

Server Instance

Invocation Target

Object Name:	com.cubeia.firebase:type=ServerInstance
Interface:	com.cubeia.firebase.server.jmx.LocalServerMBean

Methods

Add Node

Signature:

```
public void addNode(String nodeType, String id);
```

Adds a node to the current server. The node type must be one of “mtt”, “game”, or “client” and the node id must be unique across the cluster. You cannot have multiple nodes of the same type on a single server.

Remove Node

Signature:

```
public void removeNode(String id);
```

Stop and destroy a node on the local server.

Get Live Nodes

Signature:

```
public String[] getLiveNodes();
```

This method returns the currently running nodes of the server in a '<type>:<id>' string format.