

clientmanager

 Apr 03, 2024 08:46

Supported Clients

SmartClient WebClient NGClient

Methods Summary

Broadcaster	<code>getBroadcaster(name, channelName, callback)</code>	Get a broadcast object giving it a (nick)name and on a specific channel, the callback is used for getting messages of other clients on that channel The function gets 3 arguments (nickName, message, channelName)
JSClientInformation	<code>getClientInformation()</code>	Returns the current client JSClientInformation object.
Array	<code>getConnectedClients()</code>	Returns an array of JSClientInformation elements describing the clients connected to the server.
Array	<code>getConnectedClients(clientInfoFilter)</code>	/* Returns an array of JSClientInformation elements describing the clients connected to the server filtered by the a client info string.
JSClientInformation	<code>getLockedByClient(datasource, pks)</code>	Get client that locked the record from a specific datasource or null if record is not locked.
JSDataSet	<code>getLocks()</code>	Get a dataset will all locks on the server.
Boolean	<code>isInMaintenanceMode()</code>	Returns true if the server is in maintenance mode, false otherwise.
void	<code>releaseLocks(clientId)</code>	Release all locks acquired by a client WARNING: use with care
void	<code>sendMessageToAllClients(message)</code>	Sends a message to all connected clients.
void	<code>sendMessageToClient(clientId, message)</code>	Sends a message to a specific client, identified by its clientId.
void	<code>shutdownAllClients()</code>	Shuts down all connected clients.
void	<code>shutdownClient(clientId)</code>	Shuts down a specific client, identified by its clientId.
void	<code>shutdownClient(clientId, forceUnregister)</code>	Shuts down a specific client, identified by its clientId.

Methods Details

getBroadcaster(name, channelName, callback)
Get a broadcast object giving it a (nick)name and on a specific channel, the callback is used for getting messages of other clients on that channel
The function gets 3 arguments (nickName, message, channelName)

Parameters

- `String` name The nickname for this user on this channel
- `String` channelName The channel name where should be listened to (and send messages to)
- `Function` callback The callback when for incoming messages

Returns

`Broadcaster` BroadCaster

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
function callback(nickName, message, channelName) {
    application.output('message received from ' + nickName + ' on channel ' + channelName + ': ' + message)
}
var broadcaster = plugins.clientmanager.getBroadcaster("nickname", "mychatchannel", callback);
broadcaster.broadcastMessage("Hallo");
```

getClientInformation()
Returns the current client JSClientInformation object. Note this is snapshot information, client information will not get updated.

Returns

`JSClientInformation`

Supported Clients

SmartClient,WebClient,NGClient

Sample

getConnectedClients()

Returns an array of JSClientInformation elements describing the clients connected to the server. Note this is snapshot information on connected clients, client information will not get updated.

Returns

[Array](#) JSClientInformation[]

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
//Returns an array of JSClientInformation elements describing the clients connected to the server.
var clients = plugins.clientmanager.getConnectedClients();
application.output("There are " + clients.length + " connected clients.");
for (var i = 0; i < clients.length; i++)
    application.output("Client has clientId '" + clients[i].getClientID() + "' and has connected from host '" + clients[i].getHostAddress() + "'.");
```

getConnectedClients(clientInfoFilter)

/*
Returns an array of JSClientInformation elements describing the clients connected to the server filtered by the a client info string.
This way you can ask for a specific set of clients that have a specific information added to there client information.
Note this is snapshot information on connected clients, client information will not get updated.

Parameters

[String](#) clientInfoFilter The filter string

Returns

[Array](#) JSClientInformation[]

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
//Returns an array of JSClientInformation elements describing the clients connected to the server.
var clients = plugins.clientmanager.getConnectedClients();
application.output("There are " + clients.length + " connected clients.");
for (var i = 0; i < clients.length; i++)
    application.output("Client has clientId '" + clients[i].getClientID() + "' and has connected from host '" + clients[i].getHostAddress() + "'.");
```

getLockedByClient(datasource, pks)

Get client that locked the record from a specific datasource or null if record is not locked.

Parameters

[String](#) datasource;
[Array](#) pks ;

Returns

[JSClientInformation](#) Client information

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var client = plugins.clientmanager.getLockedByClient(foundset.getDataSource(),record.getPKs());
```

getLocks()

Get a dataset will all locks on the server. The dataset will have four columns: datasource, acquireDate, clientId, pkHash.
Each row in the dataset will be a lock.

Returns

[JSDataSet](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var locks = plugins.clientmanager.getLocks();
```

isInMaintenanceMode()

Returns true if the server is in maintenance mode, false otherwise.

Returns

[Boolean](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
//Returns true if the server is in maintenance mode, false otherwise.
if (plugins.maintenance.isInMaintenanceMode())
    application.output("Server is in maintenance mode.");
else
    application.output("Server is not in maintenance mode.");
```

releaseLocks(clientId)

Release all locks acquired by a client

WARNING: use with care

Parameters

[String](#) clientId;

Supported Clients

SmartClient, WebClient, NGClient

Sample**sendMessageToAllClients(message)**

Sends a message to all connected clients.

Parameters

[String](#) message;

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
//Sends a message to all connected clients.
plugins.clientmanager.sendMessageToAllClients("Hello, all clients!");
```

sendMessageToClient(clientId, message)

Sends a message to a specific client, identified by its clientId. The clientIds are retrieved by calling the getConnectedClients method.

Parameters

[String](#) clientId ;
[String](#) message;

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
//Sends a message to a specific client, identified by its clientId. The clientIds are retrieved by calling the
getConnectedClients method.
var clients = plugins.clientmanager.getConnectedClients();
for (var i=0; i<clients.length; i++)
    plugins.clientmanager.sendMessageToClient(clients[i].getClientId(), "Hello, client " + clients[i].
getClientID() + "!");
```

shutDownAllClients()

Shuts down all connected clients. This method returns immediately, it does not wait until the client shuts down.

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
//Shuts down all connected clients. This method returns immediately, it does not wait until the client shuts
down.
plugins.clientmanager.shutDownAllClients();
```

shutDownClient(clientId)

Shuts down a specific client, identified by its clientId. The clientIds are retrieved by calling the getConnectedClients method. This method returns immediately, it does not wait until the client shuts down.

Parameters

[String](#) clientId;

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
//Shuts down a specific client, identified by its clientId. The clientIds are retrieved by calling the
getConnectedClients method. This method returns immediately, it does not wait until the client shuts down.
var clients = plugins.clientmanager.getConnectedClients();
for (var i=0; i<clients.length; i++)
    plugins.clientmanager.shutDownClient(clients[i].getClientId());
```

shutDownClient(clientId, forceUnregister)

Shuts down a specific client, identified by its clientId. The clientIds are retrieved by calling the getConnectedClients method. This method returns immediately, it does not wait until the client shuts down. If forceUnregister is true, the client will unregister itself from server. Beware this should be used only if you are sure client is already closed (cannot connect anymore)

Parameters

[String](#) clientId ;
[Boolean](#) forceUnregister client is forced to unregister from server

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
//Shuts down a specific client, identified by its clientId. The clientIds are retrieved by calling the
getConnectedClients method. This method returns immediately, it does not wait until the client shuts down.
var clients = plugins.clientmanager.getConnectedClients();
for (var i=0; i<clients.length; i++)
    plugins.clientmanager.shutDownClient(clients[i].getClientId());
```