Server-side function

CallRemote

This function allows you to call functions that have been exported with HTTP access by other NRP servers. The calls are asynchronous so you do not get an immediate result from the call, instead a callback function you specify is called when the call returns.

You can also use this function to access a standard web page on a server you own by specifying the URL. The arguments you specify to callRemote are passed to the web page using HTTP POST as a JSON array. This will always have an array as the root element. The page must return a JSON formated *array* in the page's body with the results of the call (or an empty array if there are no results).

You can use the PHP SDK to create PHP pages that can be called by this function. See the PHP SDK page for an example.

In addition, it is possible to use this function to get information about a resource in the NRP community, besides other things. Check out the Community Resources article.

In the case when the call fails, a string containing "ERROR" followed by an integer containing the error reason will be passed to the callback function. The reason for failure will be similar to errors found with websites - file not found, server not found and timeouts.

Syntax

```
bool callRemote ( string host[, string queueName = "default" ][, int connectionAttempts = 10, int connectTimeout = 10000 ], s
tring resourceName, string functionName, callback callbackFunction, [ arguments... ] )
```

OR

```
bool callRemote ( string URL[, string queueName = "default" ][, int connectionAttempts = 10, int connectTimeout = 10000 ], ca
llback callbackFunction, [ arguments... ] )
```

Required Arguments

- **host:** This is a host name - including the **HTTP** port - of the server you wish to connect to.
- **resourceName:** This is a name of the resource that contains the exported function you want to call.
- **functionName:** This is a string with the name of the function which you want to call.
- **URL:** A full URL in the format *http://hostname/path/file.ext*. A port can be specified with a colon followed by a port number appended to the hostname.
- **callbackFunction:** This is the function that should receive the data returned from the remote function call. The argument list should match the format of the data returned. The callback function will be passed a string containing "ERROR" followed by an integer indicating the error code when an error occurs calling the function. A list of error codes can be found here.

Optional Arguments

NOTE: When using optional arguments, you might need to supply all arguments before the one you wish to use. For more information on optional arguments, see optional arguments.

ADDED/UPDATED IN VERSION 1.5.3-9.11270 :

- **queueName:** Name of the queue to use. Any name can be used. If not set, the queue name is "default". Requests in the same queue are processed in order, one at a time.

- **connectionAttempts:** Number of times to retry if the remote host does not respond. *In the case of a non-responding remote server, each connection attempt will timeout after 6 seconds. Therefore, the default setting of 10 connection attempts means it will be 60 seconds before your script gets a callback about the error. Reducing this value to 2 for example, will decrease that period to 12 seconds*
- **connectTimeout:** Number of milliseconds each connection attempt will take before timing out
- **arguments:** Any arguments you may want to pass to the function when it is called. Any number of arguments of can be specified, each being passed to the designated function. Most data types can be passed, including tables. The only values that cannot be passed are 'userdata' values such as xmlnodes - elements and resources can be passed though may be misinterpreted on other game servers (or cause warnings).

Returns

Returns *true* if the function has been called, *false* otherwise.