

HeadRequest

Method Summary

| | |
|----------|--|
| Boolean | addHeader (headerName, value) Add a header to the request. |
| void | executeAsyncRequest (username, password, workstation, domain, successCallbackMethod, errorCallbackMethod) Execute the request method asynchronous using windows authentication. |
| void | executeAsyncRequest (username, password, successCallbackMethod, errorCallbackMethod) Execute the request method asynchronous. |
| void | executeAsyncRequest (successCallbackMethod, errorCallbackMethod) Execute the request method asynchronous. |
| Response | executeRequest () Execute the request method. |
| Response | executeRequest (userName, password) Execute the request method. |
| Response | executeRequest (userName, password, workstation, domain) Execute a request method using windows authentication. |

Method Details

addHeader

Boolean **addHeader** (headerName, value)
Add a header to the request.

Parameters

{String} headerName
{String} value

Returns

Boolean

Sample

```
method.addHeader('Content-type','text/xml; charset=ISO-8859-1')
```

executeAsyncRequest

void **executeAsyncRequest** (username, password, workstation, domain, successCallbackMethod, errorCallbackMethod)
Execute the request method asynchronous using windows authentication. Success callback method will be called when response is received. Response is sent as parameter in callback. If no response is received (request errors out), the errorCallbackMethod is called with exception message as parameter.

Parameters

{String} username - the user name
{String} password - the password
{String} workstation - The workstation the authentication request is originating from.
{String} domain - The domain to authenticate within.
{Function} successCallbackMethod - callbackMethod to be called after response is received
{Function} errorCallbackMethod - callbackMethod to be called if request errors out

Returns

void

Sample

```
method.executeAsyncRequest('username','password','mycomputername','domain',globals.successCallback,globals.errorCallback)
```

executeAsyncRequest

void **executeAsyncRequest** (username, password, successCallbackMethod, errorCallbackMethod)
Execute the request method asynchronous. Success callback method will be called when response is received. Response is sent as parameter in callback. If no response is received (request errors out), the errorCallbackMethod is called with exception message as parameter.

Parameters

{String} username - the user name
 {String} password - the password
 {Function} successCallbackMethod - callbackMethod to be called after response is received
 {Function} errorCallbackMethod - callbackMethod to be called if request errors out

Returns

void

Sample

```
method.executeAsyncRequest(globals.successCallback,globals.errorCallback)
```

executeAsyncRequest

void **executeAsyncRequest** (successCallbackMethod, errorCallbackMethod)

Execute the request method asynchronous. Success callback method will be called when response is received. Response is sent as parameter in callback. If no response is received (request errors out), the errorCallbackMethod is called with exception message as parameter.

Parameters

{Function} successCallbackMethod - callbackMethod to be called after response is received
 {Function} errorCallbackMethod - callbackMethod to be called if request errors out

Returns

void

Sample

```
method.executeAsyncRequest(globals.successCallback,globals.errorCallback)
```

executeRequest

[Response](#) **executeRequest** ()

Execute the request method.

Returns

[Response](#)

Sample

```
var response = method.executeRequest()
```

executeRequest

[Response](#) **executeRequest** (userName, password)

Execute the request method.

Parameters

{String} userName - the user name
 {String} password - the password

Returns

[Response](#)

Sample

```
var response = method.executeRequest()
```

executeRequest

[Response](#) **executeRequest** (userName, password, workstation, domain)

Execute a request method using windows authentication.

Parameters

{String} userName - the user name
 {String} password - the password
 {String} workstation - The workstation the authentication request is originating from.
 {String} domain - The domain to authenticate within.

Returns

[Response](#)

Sample

```
var response = method.executeRequest('username','password','mycomputername','domain');
```