

# PutRequest

## Method Summary

Boolean	<a href="#">addHeader</a> (headerName, value) Add a header to the request.
void	<a href="#">executeAsyncRequest</a> (username, password, workstation, domain, successCallbackMethod, errorCallbackMethod) Execute the request method asynchronous using windows authentication.
void	<a href="#">executeAsyncRequest</a> (username, password, successCallbackMethod, errorCallbackMethod) Execute the request method asynchronous.
void	<a href="#">executeAsyncRequest</a> (successCallbackMethod, errorCallbackMethod) Execute the request method asynchronous.
Response	<a href="#">executeRequest</a> () Execute the request method.
Response	<a href="#">executeRequest</a> (userName, password) Execute the request method.
Response	<a href="#">executeRequest</a> (userName, password, workstation, domain) Execute a request method using windows authentication.
void	<a href="#">setBodyContent</a> (content) Set the body of the request.
void	<a href="#">setBodyContent</a> (content, mimeType) Set the body of the request and content mime type.
void	<a href="#">setCharset</a> (charset) Set the charset used when posting.
Boolean	<a href="#">setFile</a> (filePath) Set a file to put.

## 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');
```

**setBodyContent**

void **setBodyContent** (content)

Set the body of the request.

**Parameters**

`{String}` content

**Returns**

void

**Sample**

```
method.setBodyContent (content)
```

**setBodyContent**

void **setBodyContent** (content, mimeType)

Set the body of the request and content mime type.

**Parameters**

`{String}` content  
`{String}` mimeType

**Returns**

void

**Sample**

```
method.setBodyContent (content, 'text/xml')
```

**setCharset**

void **setCharset** (charset)

Set the charset used when posting. If this is null or not called it will use the default charset (UTF-8).

**Parameters**

charset

**Returns**

void

**Sample**

```
var client = plugins.http.createNewHttpClient();
var poster = client.createPostRequest('https://twitter.com/statuses/update.json');
poster.addParameter('status',scopes.globals.textToPost);
poster.addParameter('source','Test Source');
poster.setCharset('UTF-8');
var httpCode = poster.executeRequest(scopes.globals.twitterUserName, scopes.globals.twitterPassword).
getStatusCode() // httpCode 200 is ok
```

**setFile**

`Boolean` **setFile** (filePath)

Set a file to put.

**Parameters**

`{String}` filePath

---

**Returns**[Boolean](#)**Sample**

```
putRequest.setFile('c:/temp/manual_01a.doc')
```