


PutRequest

 Nov 22, 2019 14:27

Supported Clients

SmartClient WebClient NGClient

Methods Summary

Boolean	<code>addFile(parameterName, jsFile)</code>	Add a file to the post.
Boolean	<code>addFile(parameterName, fileName, jsFile)</code>	Add a file to the post.
Boolean	<code>addFile(parameterName, fileName, fileLocation)</code>	Add a file to the post.
Boolean	<code>addHeader(headerName, value)</code>	Add a header to the request.
Boolean	<code>addParameter(name, value)</code>	Add a parameter to the post.
void	<code>executeAsyncRequest(username, password, workstation, domain, successCallbackMethod, errorCallbackMethod)</code>	Execute the request method asynchronous using windows authentication.
void	<code>executeAsyncRequest(username, password, workstation, domain, successCallbackMethod, errorCallbackMethod, callbackExtraArgs)</code>	Execute the request method asynchronous using windows authentication.
void	<code>executeAsyncRequest(username, password, successCallbackMethod, errorCallbackMethod)</code>	Execute the request method asynchronous.
void	<code>executeAsyncRequest(username, password, successCallbackMethod, errorCallbackMethod, callbackExtraArgs)</code>	Executes the request method asynchronous.
void	<code>executeAsyncRequest(successCallbackMethod, errorCallbackMethod)</code>	Execute the request method asynchronous.
void	<code>executeAsyncRequest(successCallbackMethod, errorCallbackMethod, callbackExtraArgs)</code>	Executes the request method asynchronous.
Response	<code>executeRequest()</code>	Execute the request method.
Response	<code>executeRequest(userName, password)</code>	Execute the request method.
Response	<code>executeRequest(userName, password, workstation, domain)</code>	Execute a request method using windows authentication.
void	<code>setBodyContent(content)</code>	Set the body of the request.
void	<code>setBodyContent(content, mimeType)</code>	Set the body of the request and content mime type.
void	<code>setCharset(charset)</code>	Set the charset used when posting.
Boolean	<code>setFile(file)</code>	Set a file to put.
Boolean	<code>setFile(filePath)</code>	Set a file to put.
void	<code>usePreemptiveAuthentication(b)</code>	Whatever to use preemptive authentication (sending the credentials in the header, avoiding the server request to the client - useful when uploading files, as some http servers would cancel the first request from the client, if too big, as the authentication request to the client was not yet sent)

Methods Details

`addFile(parameterName, jsFile)`

Add a file to the post.

Parameters

```
String parameterName ;
Object jsFile      ;
```

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
poster.addFile('myFileParamName','manual.doc','c:/temp/manual_01a.doc')
poster.addFile(null,'postXml.xml','c:/temp/postXml.xml') // sets the xml to post

var f = plugins.file.convertToJSFile('./somefile02.txt')
if (f && f.exists()) poster.addFile('myTxtFileParamName','somefile.txt', f)

f = plugins.file.convertToJSFile('./anotherfile_v2b.txt')
if (f && f.exists()) poster.addFile('myOtherTxtFileParamName', f)
```

addFile(parameterName, fileName, jsFile)

Add a file to the post.

Parameters

```
String parameterName;
String fileName      ;
Object jsFile        ;
```

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
poster.addFile('myFileParamName','manual.doc','c:/temp/manual_01a.doc')
poster.addFile(null,'postXml.xml','c:/temp/postXml.xml') // sets the xml to post

var f = plugins.file.convertToJSFile('./somefile02.txt')
if (f && f.exists()) poster.addFile('myTxtFileParamName','somefile.txt', f)

f = plugins.file.convertToJSFile('./anotherfile_v2b.txt')
if (f && f.exists()) poster.addFile('myOtherTxtFileParamName', f)
```

addFile(parameterName, fileName, fileLocation)

Add a file to the post.

Parameters

```
String parameterName;
String fileName      ;
String fileLocation  ;
```

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
poster.addFile('myFileParamName','manual.doc','c:/temp/manual_01a.doc')
poster.addFile(null,'postXml.xml','c:/temp/postXml.xml') // sets the xml to post

var f = plugins.file.convertToJSFile('./somefile02.txt')
if (f && f.exists()) poster.addFile('myTxtFileParamName','somefile.txt', f)

f = plugins.file.convertToJSFile('./anotherfile_v2b.txt')
if (f && f.exists()) poster.addFile('myOtherTxtFileParamName', f)
```

addHeader(headerName, value)

Add a header to the request.

Parameters

```
String headerName;
String value      ;
```

Returns[Boolean](#)**Supported Clients**

SmartClient,WebClient,NGClient

Sample

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

addParameter(name, value)

Add a parameter to the post.

Parameters[String](#) name;[String](#) value;**Returns**[Boolean](#)**Supported Clients**

SmartClient,WebClient,NGClient

Sample

```
poster.addParameter('name','value')
poster.addParameter(null,'value') //sets the content to post
```

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**Supported Clients**

SmartClient,WebClient,NGClient

Sample

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

executeAsyncRequest(username, password, workstation, domain, successCallbackMethod, errorCallbackMethod, callbackExtraArgs)

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 followed by any 'callbackExtraArgs' that were given.

If no response is received (request errors out), the errorCallbackMethod is called with exception message as parameter followed by any 'callbackExtraArgs' that were given.

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 to be called after response is received
Function errorCallbackMethod to be called if request errors out
Array callbackExtraArgs extra arguments that will be passed to the callback methods; can be used to identify from which request the response arrived when using the same callback method for multiple requests. Please use only simple JSON arguments (primitive types or array/objects of primitive types)

Supported Clients

SmartClient, WebClient, NGClient

Sample

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

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

Supported Clients

SmartClient, WebClient, NGClient

Sample

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

executeAsyncRequest(username, password, successCallbackMethod, errorCallbackMethod, callbackExtraArgs)

Executes the request method asynchronous. Success callback method will be called when response is received. Response is sent as parameter in callback followed by any 'callbackExtraArgs' that were given. If no response is received (request errors out), the errorCallbackMethod is called with exception message as parameter followed by any 'callbackExtraArgs' that were given.

Parameters

String username the user name
String password the password
Function successCallbackMethod to be called after response is received
Function errorCallbackMethod to be called if request errors out
Array callbackExtraArgs extra arguments that will be passed to the callback methods; can be used to identify from which request the response arrived when using the same callback method for multiple requests. Please use only simple JSON arguments (primitive types or array/objects of primitive types)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
method.executeAsyncRequest(globals.successCallback,globals.errorCallback, [callIDInt])
```

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

Supported Clients

SmartClient,WebClient,NGClient

Sample

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

executeAsyncRequest(successCallbackMethod, errorCallbackMethod, callbackExtraArgs)

Executes the request method asynchronous.

Success callback method will be called when response is received. Response is sent as parameter in callback followed by any 'callbackExtraArgs' that were given.

If no response is received (request errors out), the errorCallbackMethod is called with exception message as parameter followed by any 'callbackExtraArgs' that were given.

Parameters

Fun successCal callbackMethod to be called after response is received
ctionbackMethod

Fun errorCallba callbackMethod to be called if request errors out
ctionckMethod

Arr callbackExt extra arguments that will be passed to the callback methods; can be used to identify from which request the response arrived when
ay raArgs using the same callback method for multiple requests. Please use only simple JSON arguments (primitive types or array/objects of primitive types)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
method.executeAsyncRequest(globals.successCallback,globals.errorCallback, [callIDInt])
```

executeRequest()

Execute the request method.

Returns

Response

Supported Clients

SmartClient,WebClient,NGClient

Sample

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

To be able to reuse the client, the response must be closed if the content is not read via getResponseBody or getMediaData:

```
response.close()
```

executeRequest(userName, password)

Execute the request method.

Parameters

String userName the user name

String password the password

Returns

Response

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var response = method.executeRequest()

To be able to reuse the client, the response must be
closed if the content is not read via getResponseBody
or getMediaData:

response.close()
```

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](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

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

setBodyContent(content)

Set the body of the request.

Parameters

[String](#) content;

Supported Clients

SmartClient, WebClient, NGClient

Sample

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

setBodyContent(content, mimeType)

Set the body of the request and content mime type.

Parameters

[String](#) content ;
[String](#) mimeType;

Supported Clients

SmartClient, WebClient, NGClient

Sample

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

setCharset(charset)

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

Parameters

[Object](#) charset;

Supported Clients

SmartClient, WebClient, NGClient

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(file)

Set a file to put.

Parameters

Object file;

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
putRequest.setFile(jsFileInstance)
```

setFile(filePath)

Set a file to put.

Parameters

String filePath;

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

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

usePreemptiveAuthentication(b)

Whatever to use preemptive authentication (sending the credentials in the header, avoiding the server request to the client - useful when uploading files, as some http servers would cancel the first request from the client, if too big, as the authentication request to the client was not yet sent)

Parameters

Boolean b;

Supported Clients

SmartClient,WebClient,NGClient

Sample