

HttpClient

Method Summary

DeleteRequest	createDeleteRequest(url) Creates a new delete request (a request to delete a resource on server).
GetRequest	createGetRequest(url) Creates a new get request (retrieves whatever information is stored on specified url).
HeadRequest	createHeadRequest(url) Creates a new head request (similar to get request, must not contain body content).
OptionsRequest	createOptionsRequest(url) Creates a new options request (a request for information about communication options).
PostRequest	createPostRequest(url) Create a new post request (Origin server should accept/process the submitted data.
PutRequest	createPutRequest(url) Creates a new put request (similar to post request, contains information to be submitted).
TraceRequest	createTraceRequest(url) Creates a new trace request (debug request, server will just echo back).
Cookie	getCookie(cookieName) Get a cookie by name.
Cookie[]	getCookies() Get all cookies from this client.
void	setClientProxyCredentials(userName, password) Set proxy credentials.
Boolean	setCookie(cookieName, cookieValue) Add cookie to the this client.
Boolean	setCookie(cookieName, cookieValue, domain) Add cookie to the this client.
Boolean	setCookie(cookieName, cookieValue, domain, path) Add cookie to the this client.
Boolean	setCookie(cookieName, cookieValue, domain, path, maxAge) Add cookie to the this client.
Boolean	setCookie(cookieName, cookieValue, domain, path, maxAge, secure) Add cookie to the this client.
void	setTimeout(msTimeout) Sets a timeout in milliseconds for retrieving of data (when 0 there is no timeout).

Method Details

createDeleteRequest

[DeleteRequest](#) [createDeleteRequest](#) (url)

Creates a new delete request (a request to delete a resource on server).

Parameters

{[String](#)} url

Returns

[DeleteRequest](#)

Sample

```
var client = plugins.http.createNewHttpClient();
var request = client.createDeleteRequest('http://www.servoy.com/delete.me');
var response = request.executeRequest();
var httpCode = response.getStatusCode(); // httpCode 200 is ok"
var content = response.getResponseBody();
```

createGetRequest

[GetRequest](#) [createGetRequest](#) (url)

Creates a new get request (retrieves whatever information is stored on specified url).

If this url is a https ssl encrypted url which certificates are not in the java certificate store.

(Like a self signed certificate or a none existing root certificate)

Then for a smart client a dialog will be given, to give the user the ability to accept this certificate for the next time.

For a Web or Headless client the system administrator does have to add that certificate (chain) to the java install on the server.

See <http://wiki.servoy.com/display/tutorials/Import+a+%28Root%29+certificate+in+the+java+cacerts+file>

Parameters

[{String}](#) url

Returns

[GetRequest](#)

Sample

```
var client = plugins.http.createNewHttpClient();
var request = client.createGetRequest('http://www.servoy.com');
var response = request.executeRequest();
var httpCode = response.getStatusCode(); // httpCode 200 is ok"
var content = response.getResponseBody();
```

createHeadRequest

[HeadRequest](#) **createHeadRequest** (url)

Creates a new head request (similar to get request, must not contain body content).

Parameters

[{String}](#) url

Returns

[HeadRequest](#)

Sample

```
var client = plugins.http.createNewHttpClient();
var request = client.createHeadRequest('http://www.servoy.com');
var response = request.executeRequest();
var httpCode = response.getStatusCode(); // httpCode 200 is ok
var header = response.getResponseHeaders('last-modified');
```

createOptionsRequest

[OptionsRequest](#) **createOptionsRequest** (url)

Creates a new options request (a request for information about communication options).

Parameters

[{String}](#) url

Returns

[OptionsRequest](#)

Sample

```
var client = plugins.http.createNewHttpClient();
var request = client.createOptionsRequest('http://www.servoy.com');
var methods = request.getAllowedMethods(request.executeRequest());
```

createPostRequest

[PostRequest](#) **createPostRequest** (url)

Create a new post request (Origin server should accept/process the submitted data.)

If this url is a https ssl encrypted url which certificates are not in the java certificate store.

(Like a self signed certificate or a none existing root certificate)

Then for a smart client a dialog will be given, to give the user the ability to accept this certificate for the next time.

For a Web or Headless client the system administrator does have to add that certificate (chain) to the java install on the server.

See <http://wiki.servoy.com/display/tutorials/Import+a+%28Root%29+certificate+in+the+java+cacerts+file>

Parameters

[{String}](#) url

Returns

[PostRequest](#)

Sample

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

createPutRequest

[PutRequest](#) **createPutRequest** (url)

Creates a new put request (similar to post request, contains information to be submitted).

Parameters

[{String}](#) url

Returns

[PutRequest](#)

Sample

```
var client = plugins.http.createNewHttpClient();
var request = client.createPutRequest('http://jakarta.apache.org');
request.setFile('UploadMe.gif');
var httpCode = putRequest.executeRequest().getStatusCode() // httpCode 200 is ok
```

createTraceRequest

[TraceRequest](#) **createTraceRequest** (url)

Creates a new trace request (debug request, server will just echo back).

Parameters

[{String}](#) url

Returns

[TraceRequest](#)

Sample

```
var client = plugins.http.createNewHttpClient();
var response = request.executeRequest();
var httpCode = response.getStatusCode(); // httpCode 200 is ok"
var content = response.getResponseBody();
```

getCookie

[Cookie](#) **getCookie** (cookieName)

Get a cookie by name.

Parameters

[{String}](#) cookieName

Returns

[Cookie](#)

Sample

```
var cookie = client.getCookie('JSESSIONID');
if (cookie != null)
{
    // do something
}
else
    client.setCookie('JSESSIONID', 'abc', 'localhost', '/', -1, false)
```

getCookies

[Cookie\[\]](#) **getCookies** ()

Get all cookies from this client.

Returns

[Cookie\[\]](#)

Sample

```
var cookies = client.getHttpClientCookies()
```

setClientProxyCredentials

void **setClientProxyCredentials** (userName, password)

Set proxy credentials.

Parameters

[{String}](#) userName
[{String}](#) password

Returns

void

Sample

```
client.setClientProxyCredentials('my_proxy_username','my_proxy_password');
```

setCookie

[Boolean](#) **setCookie** (cookieName, cookieValue)

Add cookie to the this client.

Parameters

[{String}](#) cookieName - the name of the cookie
[{String}](#) cookieValue - the value of the cookie

Returns

[Boolean](#)

Sample

```
var cookieSet = client.setCookie('JSESSIONID', 'abc', 'localhost', '/', -1, false)
if (cookieSet)
{
    //do something
}
```

setCookie

[Boolean](#) **setCookie** (cookieName, cookieValue, domain)

Add cookie to the this client.

Parameters

[{String}](#) cookieName - the name of the cookie
[{String}](#) cookieValue - the value of the cookie
[{String}](#) domain - the domain

Returns

[Boolean](#)

Sample

```
var cookieSet = client.setCookie('JSESSIONID', 'abc', 'localhost', '/', -1, false)
if (cookieSet)
{
    //do something
}
```

setCookie

[Boolean](#) **setCookie** (cookieName, cookieValue, domain, path)

Add cookie to the this client.

Parameters

`{String}` cookieName - the name of the cookie
`{String}` cookieValue - the value of the cookie
`{String}` domain - the domain
`{String}` path - the path

Returns

`Boolean`

Sample

```
var cookieSet = client.setCookie('JSESSIONID', 'abc', 'localhost', '/', -1, false)
if (cookieSet)
{
    //do something
}
```

setCookie

`Boolean` **setCookie** (cookieName, cookieValue, domain, path, maxAge)

Add cookie to the this client.

Parameters

`{String}` cookieName - the name of the cookie
`{String}` cookieValue - the value of the cookie
`{String}` domain - the domain
`{String}` path - the path
`{Number}` maxAge - maximum age of cookie

Returns

`Boolean`

Sample

```
var cookieSet = client.setCookie('JSESSIONID', 'abc', 'localhost', '/', -1, false)
if (cookieSet)
{
    //do something
}
```

setCookie

`Boolean` **setCookie** (cookieName, cookieValue, domain, path, maxAge, secure)

Add cookie to the this client.

Parameters

`{String}` cookieName - the name of the cookie
`{String}` cookieValue - the value of the cookie
`{String}` domain - the domain
`{String}` path - the path
`{Number}` maxAge - maximum age of cookie
`{Boolean}` secure - true if it is a secure cookie, false otherwise

Returns

`Boolean`

Sample

```
var cookieSet = client.setCookie('JSESSIONID', 'abc', 'localhost', '/', -1, false)
if (cookieSet)
{
    //do something
}
```

setTimeout

void **setTimeout** (msTimeout)

Sets a timeout in milliseconds for retrieving of data (when 0 there is no timeout).

Parameters

msTimeout

Returns

void

Sample

```
client.setTimeout(1000)
```