HttpClient

Feb 27, 2024 07:18

Supported Clients

SmartClient WebClient NGClient

Methods Summary

void	close()	releases all resources that this client has, should be called after usage.
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).
PatchRequest	createPatchRequest(url)	Creates a new patch request (used for granular updates).
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.
Array	getCookies()	Get all cookies from this client.
void	setClientProxyCredentials(userName, password)	Set proxy credentials.
void	setClientProxyServer(hostname, port)	Set proxy server.
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).

Methods Details

close()

```
releases all resources that this client has, should be called after usage.
```

Supported Clients

SmartClient,WebClient,NGClient

Sample

createDeleteRequest(url)

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

Parameters

String url ;

Returns

DeleteRequest

Supported Clients

SmartClient,WebClient,NGClient

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(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

Supported Clients

SmartClient,WebClient,NGClient

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(url)

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

Parameters

String url ;

Returns

HeadRequest

Supported Clients

SmartClient,WebClient,NGClient

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(url)

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

Parameters

String url ;

Returns

OptionsRequest

Supported Clients

SmartClient,WebClient,NGClient

Sample

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

createPatchRequest(url)

Creates a new patch request (used for granular updates).

Parameters

String url ;

Returns

PatchRequest

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var client = plugins.http.createNewHttpClient();
var request = client.createPatchRequest('http://jakarta.apache.org');
request.setBodyContent('{"email": "newemail@newdomain.com"}','application/json');
var httpCode = request.executeRequest().getStatusCode() // httpCode 200 is ok
```

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

Supported Clients

SmartClient,WebClient,NGClient

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(url)

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

Parameters

String url ;

Returns

PutRequest

Supported Clients

SmartClient,WebClient,NGClient

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(url)

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

Parameters

String url ;

Returns

TraceRequest

Supported Clients

SmartClient,WebClient,NGClient

Sample

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

getCookie(cookieName)

Get a cookie by name.

Parameters

String cookieName ;

Returns

Cookie

Supported Clients

SmartClient,WebClient,NGClient

Sample

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

getCookies()

Get all cookies from this client.

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient

Sample

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

setClientProxyCredentials(userName, password)

Set proxy credentials.

Parameters

String userName ;
String password ;

Supported Clients

SmartClient,WebClient,NGClient

Sample

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

setClientProxyServer(hostname, port)

Set proxy server.

Parameters

String hostname - proxy host // null value will clear proxyHost settings;
Number port - proxy port //null value will clear proxyHost settings;

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
client.setClientProxyServer('server',port);
```

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

Supported Clients

SmartClient,WebClient,NGClient

Sample

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

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

Supported Clients

SmartClient,WebClient,NGClient

Sample

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

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

Supported Clients

SmartClient,WebClient,NGClient

Sample

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

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

Supported Clients

SmartClient,WebClient,NGClient

Sample

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

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

Supported Clients

SmartClient,WebClient,NGClient

Sample

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

setTimeout(msTimeout)

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

Parameters

Object msTimeout ;

Supported Clients

SmartClient,WebClient,NGClient

Sample

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