

Package dev.heinzl.simplessoproxy.scripting.api

Interface ScriptingApi

public interface **ScriptingApi**

This class represents the scripting API contract. It is intended to be used in the UI's scripting editor. It serves the purpose for public documentation and interface for implementations. All functions here should be available as autocompletion in the UI's scripting editor. Current usual behavior is that runtime exception will be thrown in case of error to abort.

Author:

Johannes HEINZL, heinzl.dev

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type	Method	Description
void	addPermanentProxyRequestHeader (String key, String value)	This method adds a proxy request header to all http exchanges.
void	addPermanentProxyResponseHeader (String key, String value)	This method adds a proxy response header to all http exchanges.
void	addProxyRequestCookieIfNotPreset (org.springframework.web.server.Serve String name, String value)	This method adds a proxy request cookie if not present to a specified http exchange.
void	addProxyRequestHeaderIfNotPreset (org.springframework.web.server.Serve String key, String value)	This method adds a proxy request header if not present to a specified http exchange.
void	addProxyResponseHeaderIfNotPreset (org.springframework.web.server.Serve	This method adds a proxy response header if not present to a specified http

	String key, String value)	exchange.
void	addProxyResponseSetCookieIfNotPresent (org.springframework.web.server.Serve String name, String value)	This method adds a proxy response cookie if not present to a specified http exchange.
void	addProxyResponseSetCookieIfNotPresent (org.springframework.web.server.Serve String name, String value, String path)	This method adds a proxy response cookie if not present to a specified http exchange.
void	createGatewayFilter (groovy.lang.Closure closure)	Creates gatewayfilter with provided closure object which need to be created and provided in the scripting editor.
HttpResponse < String >	executeRequest (HttpRequest request)	Takes http request object which contains all information needed to execute the request.
void	executeScript (org.springframework.web.server.Serve ScriptType scriptType)	Executed script from internal script repository for script by providing a default type.
void	executeScript (org.springframework.web.server.Serve String scriptType)	Executed script from internal script repository for script by providing a custom type.
String	getAppBaseUrl()	Returns the app base url.
String	getAppSecret (org.springframework.web.server.Serve	Gets an app secret (e.g.
String	getBasicAuthenticationHeader (String username, String password)	Utility method to generate a basic authentication header.
org.json.JSONObject	getJSONObjectFromString (String body)	Utility method to convert a json string to a JSONObject

org.slf4j.Logger	getLogger()	Returns Logger instance.
String	getProxyPassword (org.springframework.web.server.ServerWebExchange exchange)	This method returns the password of the current logged in user of the proxy backend.
String	getProxyUsername (org.springframework.web.server.ServerWebExchange exchange)	This method returns current logged in user of the proxy backend.
dev.heinzl.simplessoproxy.scripting.RepositoryFacade	getRepositoryFacade()	Returns an instance of RepositoryFacade which provides bundled access to several repositories
void	logInfo (String text)	Logs provided text with info level in default logger.
void	setAppSecret (org.springframework.web.server.ServerWebExchange exchange, String secret)	Stores an app secret (e.g. "secret")
void	setScript (ScriptType scriptType, groovy.lang.Closure closure)	Set script in internal script repository for script with a default type.
void	setScript (String scriptType, groovy.lang.Closure closure)	Set script in internal script repository for script with a custom type.

Method Details

addProxyRequestHeaderIfNotPreset

```
void addProxyRequestHeaderIfNotPreset(org.springframework.web.server.ServerWebExchange exchange,
                                     String key,
                                     String value)
```

This method adds a proxy request header if not present to a specified http exchange.

Parameters:

exchange - represents the http communication object

key - is the header name

value - is the header value

addProxyResponseHeaderIfNotPreset

```
void addProxyResponseHeaderIfNotPreset(org.springframework.web.server.ServerWebExchange exchange,  
                                         String key,  
                                         String value)
```

This method adds a proxy response header if not present to a specified http exchange.

Parameters:

exchange - represents the http communication object

key - is the header name

value - is the header value

addProxyRequestCookieIfNotPreset

```
void addProxyRequestCookieIfNotPreset(org.springframework.web.server.ServerWebExchange exchange,  
                                         String name,  
                                         String value)
```

This method adds a proxy request cookie if not present to a specified http exchange.

Parameters:

exchange - represents the http communication object

name - is the cookie name

value - is the cookie value

addProxyResponseSetCookieIfNotPresentInRequest

```
void addProxyResponseSetCookieIfNotPresentInRequest  
(org.springframework.web.server.ServerWebExchange exchange,  
  String name,  
  String value)
```

This method adds a proxy response cookie if not present to a specified http exchange. The cookie path defaults here to "/".

Parameters:

exchange - represents the http communication object

name - is the cookie name

value - is the cookie value

addProxyResponseSetCookieIfNotPresentInRequest

```
void addProxyResponseSetCookieIfNotPresentInRequest  
(org.springframework.web.server.ServerWebExchange exchange,  
  String name,  
  String value,  
  String path)
```

This method adds a proxy response cookie if not present to a specified http exchange.

Parameters:

exchange - represents the http communication object

name - is the cookie name

value - is the cookie value

path - is the applicable path for the cookie

addPermanentProxyRequestHeader

```
void addPermanentProxyRequestHeader(String key,  
                                     String value)
```

This method adds a proxy request header to all http exchanges.

Parameters:

key - is the header name

value - is the header value

addPermanentProxyResponseHeader

```
void addPermanentProxyResponseHeader(String key,  
                                     String value)
```

This method adds a proxy response header to all http exchanges.

Parameters:

key - is the header name

value - is the header value

getProxyUsername

```
String getProxyUsername(org.springframework.web.server.ServerWebExchange exchange)
```

This method returns current logged in user of the proxy backend. It extracts the username from a JWT value provided by the proxy cookie set during authentication.

Parameters:

exchange - represents the http communication object

Returns:

the username

getProxyPassword

```
String↗ getProxyPassword(org.springframework.web.server.ServerWebExchange exchange)
```

This method returns the password of the current logged in user of the proxy backend. The password is normally stored temporarily and is retrieved with the username by utilizing `getProxyUsername(ServerWebExchange exchange)`. It returns null if either no password is available or the temporarily password store has it expired already

Parameters:

exchange - represents the http communication object

Returns:

the username or null

getAppSecret

```
String↗ getAppSecret(org.springframework.web.server.ServerWebExchange exchange)
```

Gets an app secret (e.g. session token) from a secret store. Http exchange is needed to determine the user.

Parameters:

exchange - represents the http communication object

Returns:

secret

setAppSecret

```
void setAppSecret(org.springframework.web.server.ServerWebExchange exchange,  
                  String↗ secret)
```

Stores an app secret (e.g. session token) in a secret store. Http exchange is needed to determine the user.

Parameters:

exchange - represents the http communication object

secret - to be stored

getAppBaseUrl

```
String ↗ getAppBaseUrl()
```

Returns the app base url.

Returns:

app base url

getLogger

```
org.slf4j.Logger getLogger()
```

Returns Logger instance.

Returns:

org.slf4j.Logger instance

logInfo

```
void logInfo(String ↗ text)
```

Logs provided text with info level in default logger.

Parameters:

text - to be logged

getRepositoryFacade

```
dev.heinzl.simplessoproxy.scripting.RepositoryFacade getRepositoryFacade()
```

Returns an instance of RepositoryFacade which provides bundled access to several repositories

Returns:

Instance of RepositoryFacade

executeRequest

```
HttpResponse<String> executeRequest(HttpRequest request)
```

Takes http request object which contains all information needed to execute the request. It executes the request in a blocking manner.

Parameters:

request - java.net.http.HttpRequest object

Returns:

java.net.http.HttpResponse object with body as string

getBasicAuthenticationHeader

```
String getBasicAuthenticationHeader(String username,  
                                     String password)
```

Utility method to generate a basic authentication header.

Parameters:

username - is the username as string

password - is the password as string

Returns:

basic auth header

getJSONObjectFromString

```
org.json.JSONObject getJSONObjectFromString(String body)
```

Utility method to convert a json string to a JSONObject.

Parameters:

body - as json string

Returns:

org.json.JSONObject on successful conversion

createGatewayFilter

```
void createGatewayFilter(groovy.lang.Closure closure)
```

Creates gatewayfilter with provided closure object which need to be created and provided in the scripting editor.

Parameters:

closure - groovy.lang.Closure as function with scripted custom logic

setScript

```
void setScript(ScriptType scriptType,  
              groovy.lang.Closure closure)
```

Set script in internal script repository for script with a default type.

Parameters:

scriptType - from enum of default script types

closure - groovy.lang.Closure as function with scripted custom logic

setScript

```
void setScript(String? scriptType,  
              groovy.lang.Closure closure)
```

Set script in internal script repository for script with a custom type.

Parameters:

scriptType - as custom script type

closure - groovy.lang.Closure as function with scripted custom logic

executeScript

```
void executeScript(org.springframework.web.server.ServerWebExchange exchange,  
                  ScriptType scriptType)
```

Executed script from internal script repository for script by providing a default type.

Parameters:

exchange - represents the http communication object which triggers the execution

scriptType - from enum of default script types

executeScript

```
void executeScript(org.springframework.web.server.ServerWebExchange exchange,  
                  String scriptType)
```

Executed script from internal script repository for script by providing a custom type.

Parameters:

exchange - represents the http communication object which triggers the execution

scriptType - as custom script type