

Table Of Content

ClientAuthInterface	2
Index	5

Class ClientAuthInterface

```
java.lang.Object
|
+--gemom.client.ClientAuthInterface
```

All Implemented Interfaces:

org.eclipse.higgins.crpps.service.TokenListener

< [Constructors](#) > < [Methods](#) >

```
public class ClientAuthInterface
extends java.lang.Object
implements org.eclipse.higgins.crpps.service.TokenListener
```

The **ClientAuthInterface** class allows a client application to manage the different phases of the authentication process. In particular it provides support to process the data received from the *Service Provider (SP)*, to request an authentication token to the the *Security Token Service (STS)*, token to be forwarded to the SP. It is assumed that the client application has an associate *iCard* (in particular a *managed iCard*) provided by the same *IdP* to be used during the authentication process with its related STS. The class provides a method to configure the library with the location path of the *iCard*, its associated password and the path of the XML configuration file of the client. The class provides also a method to read and process this XML configuration file. The class is in charge of managing the whole interaction process with the STS. The security token request process begins with a *WS-Trust* binding between the client and the STS: to this end the *AXIS Framework* is used. Afterwards the class creates and submits an STS Token Request and receives the security token. Please refer to the installation and configuration manual for more details.

Version:

1.1

Author:

Leonardo Straniero TXT e-solution Spa

Constructors

ClientAuthInterface

```
public ClientAuthInterface()
```

Methods

getCertsFromSecureConnection

```
public java.lang.String getCertsFromSecureConnection(javax.net.ssl.SSLSession  
secureSession)  
throws javax.net.ssl.SSLPeerUnverifiedException,  
javax.security.cert.CertificateEncodingException
```

This method **must be used** to acquire the *SP's* X.509 certificate chain from an SSLSession instance when the server does not explicitly provide its certificate or when the client logic does not want to use the provided certificate in favour of the one used, for example, to set up an SSL connection.

Parameters:

secureSession - is the SSLSession instance from which the X.509 certificate chain must be acquired.

Returns:

The string representing the certificates chain.

Throws:

javax.net.ssl.SSLPeerUnverifiedException -
javax.security.cert.CertificateEncodingException -

notifyCanceled

```
public void notifyCanceled()
```

notifySecurityToken

```
public void notifySecurityToken(java.lang.String token)
```

This method simply performs the log, using log4j, of the received encrypted authentication token.

Parameters:

token - A string containing the security token to be logged.

parseInput

```
public void parseInput(java.lang.String rp_policy,  
                       java.lang.String certificate_chain)
```

This method initializes the authentication library on the client side by reading the configuration parameters from a configuration file. It is responsible to acquire the iCard to be used for the authentication process and the related credentials to be used toward the STS. Additionally, this method acquires, from the configuration file, the location of an additional configuration file that specifies the classes the client library has to use to manage the client-STs binding operation, the messages' encryption, the SecuritySTSToken service and the XMLSecurity service. For more details please refer the installation and configuration guide of the library.

Parameters:

rp_policy - The string that represents the *SP*'s authentication policy, as received from the *SP*.

certificate_chain - The string containing the *SP*'s X.509 certificate chain, as received from the *SP* or acquired from an SSLSession instance

requestToken

```
public java.lang.String requestToken()
```

This method is used to simply request a security token to the STS. It creates a request based on the authentication context that has been set using the `parseInput()` method.

Returns:

A string that contains the encrypted security token. The security token will contain an un-encrypted string with a value of "NO TOKEN" if the authentication process fails.

requestToken

```
public java.lang.String requestToken(java.lang.StringBuffer display_token)
```

This method is used to request to the STS both a security token (which is always encrypted) and a clear-text version of the token (*display_token*) the client application can use for its own purposes (e.g. display token claims to an end-user). The method creates a request based on the authentication context that has been set using the `parseInput()` method.

Parameters:

display_token - The object to be set with a clear-text version of the token to be made available to the calling method.

Returns:

A string that contains the encrypted security token. The security token will contains an un-encrypted string with a value of "NO TOKEN" if the authentication process fails.

INDEX

C

[ClientAuthInterface](#) ... 2

[ClientAuthInterface](#) ... 2

G

[getCertsFromSecureConnection](#) ... 3

N

[notifyCanceled](#) ... 3

[notifySecurityToken](#) ... 3

P

[parseInput](#) ... 4

R

[requestToken](#) ... 4

[requestToken](#) ... 4