

# Table Of Content

---

<a href="#">gemom.client.ui</a> .....	2
<a href="#">ClientUIAuthInterface</a> .....	2
<a href="#">Index</a> .....	4

# Package gemom.client.ui

## Class Summary

### [ClientUIAuthInterface](#)

The **ClientUIAuthInterface** class provides all functionalities required to perform the authentication involving an end-user and using a **Card Selector** (e.g.

gemom.client.ui

## Class ClientUIAuthInterface

```
java.lang.Object
|
+--gemom.client.ui.ClientUIAuthInterface
```

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

```
public class ClientUIAuthInterface
extends java.lang.Object
```

The **ClientUIAuthInterface** class provides all functionalities required to perform the authentication involving an end-user and using a **Card Selector** (e.g. Microsoft *CardSpace*). The *Card Selector* will present to the end-user the *iCards* that meet the *SP's Authentication Policy*, give him/her the opportunity to select one of them, check which info will be provided, manage the actual authentication process with the **STS** and return the *Authentication Token* to the calling client application. The *ClientUIAuthInterface* class requires the availability of additional software components on the client machine; specifically: the **Higgins Selector Selector** (*HSS*, aka **Higgins Selector Switch**) and of a **Card Selector** (e.g. Microsoft *CardSpace*). Under Microsoft Windows the *HSS* acts as a Windows service and it must be active in order to perform an authentication as described above. The returned *Authentication Token* must be provided by the client application to the SP which has to process it using the *GEMOM Java Server Authentication Library*.

**Version:**

1.0

**Author:**

Leonardo Straniero TXT e-solution Spa

## Constructors

### ClientUIAuthInterface

```
public ClientUIAuthInterface()
```

## 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 has not explicitly provided 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 a `SSLSession` instance from which the X.509 certificate chain must be acquired.

### Returns:

The string representing the certificate chain.

### Throws:

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

---

## openCardSelector

```
public java.lang.String openCardSelector(java.lang.String policy,
    java.lang.String certificate_chain)
```

This method is in charge of invoking the *Card Selector*. It creates the request to be sent to the *Higgins Selector Selector* service using the provided *SP's Authentication Policy* and *X.509 certificate chain*.

### Parameters:

`policy` - This string represents the *SP's authentication policy* as received from the *SP*.  
`certificate_chain` - This string contains the *SP's X.509 certificate chain* as received from the *SP* or acquired from an `SSLSession` instance

### Returns:

A string that contains the encrypted authentication token. The returned string will contain a value of *"NO TOKEN"* if the authentication process fails.

# INDEX

## C

[ClientUIAuthInterface](#) ... 2  
[ClientUIAuthInterface](#) ... 2

## G

[getCertsFromSecureConnection](#) ... 3

## O

[openCardSelector](#) ... 3