SeguriSign versión 8.6

Manual de la API

Revisión 1.0

SeguriData

SeguriData Privada, S.A. de C.V.

Av. Insurgentes Sur #2375, 3er. Piso,
Col. Tizapán, Del. Alvaro Obregón,
CP. 01000, México, D.F.

Tel.: +52 (55) 3098.0700
Fax: +52 (55) 3098.0702

http://www.seguridata.com

Derechos Reservados © SeguriData IP S.A. de C.V., Av. Insurgentes Sur #2375, 3er. piso, Col. Tizapán, Del. Alvaro Obregón, C.P. 01000, México, D.F.. Derechos Reservados © SeguriData Privada S.A. de C.V., Av. Insurgentes Sur #2375, 3er. piso, Col. Tizapán, México, D.F., México, 1998. Este producto constituye una obra intelectual protegida por las leyes nacionales y tratados internacionales en materia de derechos de autor, y queda prohibida su reproducción o uso total o parcial, que no sean autorizadas por su titular.

Número de Parte: API8.6

Contenido

CAPÍTULO 1

¿Cómo utilizar este manual?

1.1 Simbología y convenciones

En todo el manual se hace uso de una simbología específica para hacer más sencilla la identificación del tipo de información que se expone, así como de convenciones tipográficas, para hacer más clara la documentación.

1.1.1 Recomendaciones y Advertencias

En los lugares que resulte mas oportuno, se insertarán comentarios sobre el contenido del texto.

Importante

Este tipo de anotaciones contienen sugerencias y aclaraciones que facilitan el uso de la aplicación.

Precaución

Este tipo de anotaciones advierten sobre posibles riesgos en las operaciones descritas en el texto y que pueden causar pérdida de funcionalidad o datos.

1.1.2 Tipografía

Las convenciones tipográficas utilizadas a lo largo del manual indican tanto la fuente como el uso de los datos que aparecen bajo ese estilo.

El texto que aparezca con el tipo de letra usuario, deberá de introducirse tal cual aparece en el manual, en la interfaz que se esté utilizando en ese momento.

Los textos que se muestren como:

comando <*parámetro 1*>,[,<*parámetro 2*>] <parámetro 3>

denotarán la sintaxis de un comando. La palabra **comando**, se deberá escribir tal como aparece, en tanto que <*parámetro 1*> y <*parámetro 2*> se deberán de sustituir por valores que dependen del proceso que se realiza. El <parámetro 3> corresponde a un texto que se debe usar literalmente como aparece en la sintaxis.

Los símbolos < y > delimitarán el nombre del parámetro a que se haga referencia, requerida en el comando.

El uso de los paréntesis [y] se usarán para denotar la parte del comando que es opcional.

La sintaxis

[, <*parámetro 2*> …]

en un comando indica que se podrán agregar tantos parámetros como sea necesario y

{ <*valor 1*> | <*valor 2*> | ... | <*valor n*> }

indicará que se deberá usar solamente uno de los *n* valores listados entre las llaves.

Finalmente, para indicar los términos que se usan por primera vez o que están en otro idioma, se usará letra *cursiva*. Estos términos aparecerán en el Glosario de Términos en el Apéndice de este manual.

1.1.3 Acciones bajo la interfaz de usuario

Al interactuar con la interfaz de usuario, el acceso a los diversos menues se denotará como una secuencia de las opciones que se deben seleccionar, separadas por un símbolo triangular (>) que indica el paso al siguiente nivel de submenú. Por ejemplo, para indicar que se debe seleccionar la opción Page Size … que se encuentra en el submenú Page Layout y éste a su vez se encuentra en el menú Format, se indicará como:

Format > Page Layout > Page Size …

El acceso a los menues se puede realizar con combinaciones de teclas y seleccionando la opción con la tecla [Return] o [Enter], o bien puede seleccionar con ayuda del puntero del ratón, en cuyo caso la selección se hará oprimiendo el botón izquierdo del ratón cuando el puntero se encuentre sobre la opción deseada.

A esta última forma de seleccionar una opción se le referirá como hacer *clic* sobre la opción.

1.2 Objetivo de este manual

Debido a que SeguriSign es una aplicación que proporciona servicios de autenticación, almacenamiento y administración de documentos firmados, la mayoría de las instrucciones específicas se encuentra en los siguientes capítulos de este manual.

Este manual está orientado al personal que actuará como administrador de SeguriSign, a cargo de labores de instalación y configuración de SeguriSign.

CAPÍTULO 2

La API de SeguriSign para Java.

2.1 En qué consiste la API de SeguriSign para Java.

Dentro de las capacidades de SeguriSign, se encuentra la de proporcionar un medio de acceso a ciertas rutinas que pueden invocarse desde aplicaciones Java a fin de gestionar procesos de firma unilateral y multilateral.

El archivo SSignv8.6.jar permitirá la incorporación de estos procesos a cualquier aplicación desarrollada en Java 6 SE/EE.

La API proporciona las siguientes funcionalidades para procesos de firma:

UNILATERAL

- Autenticar mensajes firmados y/o ensobretados.
- Obtener los detalles de un mensaje firmado y su firmante
- Recuperación del mensaje original asegurado
- Solicitud de las evidencias criptográficas asociadas a cierto proceso: mensaje firmado, estampa asociada y respuesta OCSP.

MULTILATERAL

- Iniciar procesos de firma
- Autenticar mensajes firmados para un determinado proceso
- Finalizar procesos
- Obtener el estatus de un proceso
- Solicitar se origine un mensaje criptográfico para un proceso de firma en proceso o terminado
- Obtener los detalles de un CMS: características de las firmas, certificados de firmantes y autoridades, evidencias asociadas como pueden ser estampas de tiempo y respuestas OCSP.

Además habilita:

- Decodificación de estampas de tiempo TSP
- Obtener los detalles de un certificado digital

2.2 Funciones de la API

2.2.1 getOcspResponses

Función: Obtiene las respuestas OCSP encontradas en el CMS

Sintaxis Java: public java.util.ArrayList<SSignEvidence> getOcspResponses()

Parámetros de Salida: Arreglo de objetos con las respuestas OCSP encontradas en el CMS

2.2.2 setOcspResponses

Función: Asigna las respuestas OCSP del mensaje .

Sintaxis Java:public void setOcspResponses(java.util.ArrayList<SSignEvidence> ocspResponses) GetOriginalDocument

2.2.3 toString

Sintaxis Java:public java.lang.String toString()

2.2.4 getCertificates

Función: Obtiene los certificados codificados en el CMS .

Sintaxis Java:public java.util.HashMap<java.lang.String,SSignCertificate> getCertificates()

Regresa: Arreglo de objetos de tipo certificado

2.2.5 setCertificates

Función: Asigna el arreglo de objetos del CMS .

Sintaxis Java:public void setCertificates(java.util.HashMap<java.lang.String,SSignCertificate> certificates)

2.2.6 getData

Sintaxis Java:public byte[] getData()Obtiene los bytes que conforman el CMS

Regresa: Bytes que conforman el CMS

2.2.7 setData

Sintaxis Java:public void setData(byte[] data)Asigna los bytes que conforman el CMS

Parametros:

data - Conjunto de bytes que conforman el CMS

2.2.8 getSignerInfos

Sintaxis Java:public SignerInfos getSignerInfos()Obtiene el signer infos del CMS

Regresa: Signer infos codificados en el CMS

2.2.9 setSignerInfos

Sintaxis Java:public void setSignerInfos(SignerInfos signerInfos)Asigna los signer infos contenidos en el mensaje

Parametros:

signerInfos - Signer infos del CMS

2.2.10toString

Sintaxis Java:public java.lang.String toString()Overrides:

toString in class java.lang.Object

2.2.11getIssuer

Sintaxis Java:public java.lang.String getIssuer()Obtiene los nombres del emisor codificados en el signer info

Regresa: Nombres del emisor codificados en el signer info

2.2.12setIssuer

Sintaxis Java:public void setIssuer(java.lang.String issuer)Dispone los nombres del emisor codificados en el signer info

Parametros:

issuer - Nombres del emisor codificados en el signer info

2.2.13getSerialNumber

Sintaxis Java:public java.lang.String getSerialNumber()Devuelve la serie del certificado contenida en el signer info

Regresa: Serie del certificado codificada en el signer info

2.2.14setSerialNumber

Sintaxis Java:public void setSerialNumber(java.lang.String serialNumber)Asigna la serie del certificado contenida en el signer info

Parametros:

serialNumber - Serie del certificado contenida en el signer info

2.2.15getIssuerBlock

Sintaxis Java:public byte[] getIssuerBlock()Retorna el bloque DER de la codificación de los nombres de emisor

Regresa: Bloque DER de la codificación de los nombres de emisor

2.2.16setIssuerBlock

Sintaxis Java:public void setIssuerBlock(byte[] issuerBlock)Asigna el bloque DER de la codificación de los nombres de emisor

Parametros:

issuerBlock - Bloque DER de la codificación de los nombres de emisor

2.2.17toString

Sintaxis Java:public java.lang.String toString()Overrides:

toString in class java.lang.Object

2.2.18-MismatchedDocumentException

Sintaxis Java:public MismatchedDocumentException()

2.2.19MismatchedDocumentException

Sintaxis Java:public MismatchedDocumentException(java.lang.String message,

java.lang.Throwable arg1)

2.2.20MismatchedDocumentException

Sintaxis Java:public MismatchedDocumentException(java.lang.String message)

2.2.21MismatchedDocumentException

Sintaxis Java:public MismatchedDocumentException(java.lang.Throwable arg0)

2.2.22setId

Sintaxis Java:public void setId(java.lang.String id)Especifica el identificador para la transacción de firma multilateral

Parametros:

id - Valor del identificador para la transacción

2.2.23getId

Sintaxis Java:public java.lang.String getId()Retorna el valor del identificador de la transacción de firma multilateral

Regresa: Valor del identificador de la transacción de firma multilateral

2.2.24setThumbPrint

Sintaxis Java:public void setThumbPrint(java.lang.String thumbPrint)Especifica la huella digital del proceso de firma multilateral

2.2.25getThumbPrint

Sintaxis Java:public java.lang.String getThumbPrint()Retorna la huella digital del proceso de firma multilateral

Regresa: Huella digital del proceso de firma multilateral

2.2.26setInfo

Sintaxis Java:public void setInfo(java.lang.String info)Asigna la información adicional asociada al proceso de firma multilateral

2.2.27getInfo

Sintaxis Java:public java.lang.String getInfo()Retorna la información adicional asociada al proceso de firma multilateral

Regresa: Información adicional asociada al proceso de firma multilateral

2.2.28setTimeStamp

Sintaxis Java:public void setTimeStamp(SSignEvidence timeStamp)Asocia la estampa de tiempo asociada al proceso de firma multilateral

Parametros:

timeStamp - Estampa de tiempo asociada a la creación del CMS

2.2.29getTimeStamp

Sintaxis Java:public SSignEvidence getTimeStamp()Devuelve la estampa de tiempo asociada a la creación del CMS

Regresa: Estampa de tiempo asociada a la creación del CMS

2.2.30setCms

Sintaxis Java:public void setCms(SSignEvidence cms)Asigna el cms del proceso de firma multilateral

Parametros:

cms - CMS del proceso de firma multilateral

2.2.31getCms

Sintaxis Java:public SSignEvidence getCms()Retorna el cms del proceso de firma multilateral

Regresa:CMS para el proceso de firma multilateral

2.2.32toString

Sintaxis Java:public java.lang.String toString()Overrides:

toString in class java.lang.Object

2.2.33getPort

Sintaxis Java:public int getPort()Obtiene el valor asignado al puerto del servidor SeguriSign

Regresa: Puerto del servidor SeguriSign

2.2.34setPort

Sintaxis Java:public void setPort(int port)Asigna el valor del puerto del servidor con el que se manejarán las transacciones

Parametros:

port - Puerto del servidor SeguriSign

2.2.35getServer

Sintaxis Java:public java.lang.String getServer()Obtiene el nombre o ip del servidor SeguriSign

Regresa: Nombre o ip del servidor SeguriSign

2.2.36setServer

Sintaxis Java:public void setServer(java.lang.String server)Asigna el nombre o ip del servidor SeguriSign con el que se gestionarán los procesos de firma

Parametros:

server - Nombre o ip del servidor SeguriSign

2.2.37getTimeOut

Sintaxis Java:public int getTimeOut()Obtiene el tiempo definido en milisegundos para espera en procesos de entrada y salida de datos en comunicaciones con el servidor

Regresa: Tiempo en milisegundos

2.2.38setTimeOut

Sintaxis Java:public void setTimeOut(int timeOut)Contiene la cantidad de tiempo en milisegundos para espera de entrada de datos para comunicaciones con el servidor

Parametros:

timeOut - Tiempo en milisegundos

2.2.39init

Sintaxis Java:public MultilateralProcess init(byte[] data,

Process.DataType dataType,

ParameterTypes.DigestAlgorithm digestAlgorithm,

java.lang.String dataInfo,

Process.ProcessType processType)

java.lang.IllegalArgumentException,

java.lang.ExceptionInicia un proceso de firma multilateral

Parametros:

data - Información a asociar al proceso de firma multilateral

dataType - Especifica cómo interpretar los datos parametrizados

digestAlgorithm - Define el algoritmo de digestión involucrado en el proceso de firma (SHA1)

dataInfo - Metadato para el proceso de firma

processType - Indica si se genera un paquete firmado CMS o XML

Regresa: Detalles del proceso multilateral iniciado

Arroja:

java.lang.IllegalArgumentException - Alguno de los argumentos es nulo o no tiene valor

java.lang.Exception - Error durante el procesamiento del mensaje

2.2.40getTimeOut

Sintaxis Java:public int getTimeOut()Obtiene el tiempo definido en milisegundos para espera en procesos de entrada y salida de datos en comunicaciones con el servidor

Regresa: Tiempo en milisegundos

2.2.41setTimeOut

Sintaxis Java:public void setTimeOut(int timeOut)Contiene la cantidad de tiempo en milisegundos para espera de entrada de datos para comunicaciones con el servidor

Parametros:

timeOut - Tiempo en milisegundos

2.2.42init

Sintaxis Java:public MultilateralProcess init(byte[] data,

Process.DataType dataType,

ParameterTypes.DigestAlgorithm digestAlgorithm,

java.lang.String dataInfo,

Process.ProcessType processType)

java.lang.IllegalArgumentException,

java.lang.ExceptionInicia un proceso de firma multilateral

Parametros:

data - Información a asociar al proceso de firma multilateral

dataType - Especifica cómo interpretar los datos parametrizados

digestAlgorithm - Define el algoritmo de digestión involucrado en el proceso de firma (SHA1)

dataInfo - Metadato para el proceso de firma

processType - Indica si se genera un paquete firmado CMS o XML

Regresa: Detalles del proceso multilateral iniciado

Arroja:

java.lang.IllegalArgumentException - Alguno de los argumentos es nulo o no tiene valor

java.lang.Exception - Error durante el procesamiento del mensaje

2.2.43getStatus

Sintaxis Java:public ParameterTypes.ProcessStatus getStatus(java.lang.String id)

java.lang.IllegalArgumentException,

java.lang.ExceptionSolicita el estatus para un proceso de firma multilateral existente

Parametros:

id - Identificador del proceso de firma multilateral del que se solicita el estatus

Regresa:Estatus actual del proceso

Arroja:

java.lang.IllegalArgumentException - Alguno de los argumentos es nulo o sin valor

java.lang.Exception - Se presentaron errores de comunicación con el servidor, disponibilidad o existencia del proceso

2.2.44getMultilateralSignedMessageData

Sintaxis Java:public CMSSignedMessage getMultilateralSignedMessageData(byte[] cms,

java.lang.String id,

ParameterTypes.DigestAlgorithm digestAlgorithm)

java.lang.IllegalArgumentException,

java.lang.Exception

Obtiene los certificados contenidos en un mensaje criptográfico firmado unilateral o multilateralmente

Parametros:

cms - Mensaje Criptográfico a decodificar

id - Identificador del proceso de firma multilateral al que corresponde el CMS parametrizado

digestAlgorithm - Algoritmo de digestión para generar huellas digitales (SHA1)

Regresa: Regresa objeto con evidencias encontradas en CMS

Arroja:

java.lang.IllegalArgumentException - Los argumentos recibidos no fueron satisfactorios

java.lang.Exception

2.2.45 getRDN

Sintaxis Java: public java.util.HashMap<java.lang.String,java.lang.String> getRDN(java.lang.String subjectNames)

java.lang.Exception

Obtiene un mapa de datos para los nombres distinguidos de un certificado

Parametros:

subjectNames - Cadena con los nombres distinguidos obtenida de un objeto SSignCertificate

Regresa: Objeto con los datos de la cadena de nombres y que tienen como llave para localizar al objeto las etiquetas de los nombres distinguidos

Arroja:

java.lang.Exception - Los datos no pudieron ser procesados correctamente

2.2.46ParameterTypes

Sintaxis Java:public ParameterTypes()

2.2.47valueOf

Sintaxis Java: public static ParameterTypes.CMSType valueOf(java.lang.String name) Regresa la constante con el nombre de enumeración. La cadena debe coincidir exactamente con un identificador utilizado para declarar una constante en este tipo de enumeración. (Espacios en blanco no están permitidos.)

Parametros:

name - el nombre de la constante de enumeración que se devuelve.

Regresa: la constante de enumeración con el nombre especificado.

Arroja:

java.lang.IllegalArgumentException - si este tipo de enumeración no es constante con el nombre especificado

java.lang.NullPointerException - si el argumento es nulo

2.2.48getValue

Sintaxis Java:public int getValue()

2.2.49values

Sintaxis Java:public static ParameterTypes.DataType[] values() Regresa una matriz que contiene las constantes de este tipo de enumeración, en el orden en que se declaró. Este método se puede utilizar para iterar a través de las constantes de la siguiente manera:

```
for (ParameterTypes.DataType c : ParameterTypes.DataType.values())
   System.out.println(c);
```

Regresa: una matriz que contiene las constantes de este tipo de enumeración, en el orden en que se declaran.

2.2.50valueOf

Sintaxis Java:public static ParameterTypes.DataType valueOf(java.lang.String name) Regresa la constante de este tipo con el nombre de enumeración. La cadena debe coincidir exactamente con un identificador utilizado para declarar una constante en este tipo de enumeración. (Espacios en blanco no están permitidos.)

Parametros:

name - el nombre de la constante de enumeración que se devuelve.

Regresa: la constante de enumeración con el nombre especificado.

Arroja:

java.lang.IllegalArgumentException - si este tipo de enumeración no es constante con el nombre especificado.

java.lang.NullPointerException - si el argumento es nulo.

2.2.51getValue

Sintaxis Java:public int getValue()

2.2.52Process

Sintaxis Java:public Process()

2.2.53values

Sintaxis Java:public static Process.CMSType[] values() Regresa una matriz que contiene las constantes de este tipo de enumeración, en el orden en que se declaró. Este método se puede utilizar para iterar a través de las constantes de la siguiente manera:

```
for (Process.CMSType c : Process.CMSType.values())
   System.out.println(c);
```

Regresa: una matriz que contiene las constantes de este tipo de enumeración, en el orden en que se declaran.

2.2.54valueOf

Sintaxis Java:public static Process.CMSType valueOf(java.lang.String name) Regresa la constante de este tipo con el nombre de enumeración. La cadena debe coincidir exactamente con un identificador utilizado para declarar una constante en este tipo de enumeración. (Espacios en blanco no están permitidos.)

Parametros:

name - el nombre de la constante de enumeración que se devuelve.

Regresa: la constante de enumeración con el nombre especificado.

Arroja:

java.lang.IllegalArgumentException - si este tipo de enumeración no es constante con el nombre especificado.

java.lang.NullPointerException - si el argumento es nulo.

2.2.55getValue

Sintaxis Java: public int getValue()

2.2.56 values

Sintaxis Java:public static Process.DataType[] values() Regresa una matriz que contiene las constantes de este tipo de enumeración, en el orden en que se declaró. Este método se puede utilizar para iterar a través de las constantes de la siguiente manera:

for (Process.DataType c : Process.DataType.values())

System.out.println(c);

Regresa:una matriz que contiene las constantes de este tipo de enumeración, en el orden en que se declaran.

2.2.57valueOf

Sintaxis Java:public static Process.DataType valueOf(java.lang.String name)Regresa la constante de este tipo con el nombre de enumeración. La cadena debe coincidir exactamente con un identificador utilizado para declarar una constante en este tipo de enumeración. (Espacios en blanco no están permitidos.)

Parametros:

name - el nombre de la constante de enumeración que se devuelve.

Regresa: la constante de enumeración con el nombre especificado.

Arroja:

java.lang.IllegalArgumentException -si este tipo de enumeración no es constante con el nombre especificado.

java.lang.NullPointerException - si el argumento es nulo

2.2.58getValue

Sintaxis Java:public int getValue()

2.2.59 values

Sintaxis Java:public static Process.EvidenceType[] values() Regresa una matriz que contiene las constantes de este tipo de enumeración, en el orden en que se declaró. Este método se puede utilizar para iterar a través de las constantes de la siguiente manera:

for (Process.EvidenceType c : Process.EvidenceType.values())

System.out.println(c);

Regresa: una matriz que contiene las constantes de este tipo de enumeración, en el orden en que se declaran.

2.2.60valueOf

Sintaxis Java:public static Process.EvidenceType valueOf(java.lang.String name) Regresa la constante de este tipo con el nombre de enumeración. La cadena debe coincidir exactamente con un identificador utilizado para declarar una constante en este tipo de enumeración. (Espacios en blanco no están permitidos.)

Parametros:

name - el nombre de la constante de enumeración que se devuelve.

Regresa: la constante de enumeración con el nombre especificado.

Arroja:

java.lang.IllegalArgumentException - si este tipo de enumeración no es constante con el nombre especificado.

java.lang.NullPointerException - si el argumento es nulo

2.2.61-getValue

Sintaxis Java:public int getValue()

2.2.62 values

Sintaxis Java:public static Process.ProcessType[] values() Regresa una matriz que contiene las constantes de este tipo de enumeración, en el orden en que se declaró. Este método se puede utilizar para iterar a través de las constantes de la siguiente manera:

for (Process.ProcessType c : Process.ProcessType.values())

System.out.println(c);

Regresa: una matriz que contiene las constantes de este tipo de enumeración, en el orden en que se declaran.

2.2.63valueOf

Sintaxis Java:public static Process.ProcessType valueOf(java.lang.String name) Regresa la constante de este tipo con el nombre de enumeración. La cadena debe coincidir exactamente con un identificador utilizado para declarar una constante en este tipo de enumeración. (Espacios en blanco no están permitidos.)

Parametros:

name - el nombre de la constante de enumeración que se devuelve.

Regresa: la constante de enumeración con el nombre especificado.

Arroja:

java.lang.IllegalArgumentException - si este tipo de enumeración no es constante con el nombre especificado.

java.lang.NullPointerException - si el argumento es nulo.

2.2.64getValue

Sintaxis Java:public int getValue()

2.2.65- getDigestAlgorithm

Sintaxis Java:public java.lang.String getDigestAlgorithm() Obtiene nombre del algoritmo de digestión codificado en el signer info

Regresa: Nombre del algoritmo de digestión codificado en el signer info

2.2.66setDigestAlgorithm

Sintaxis Java:public void setDigestAlgorithm(java.lang.String digestAlgorithm) Asigna el nombre del algoritmo de digestión codificado en el signer info

Parametros:

digestAlgorithm - Algoritmo de digestión codificado en el signer info

2.2.67getDigestEncryptionAlgorithm

Sintaxis Java:public java.lang.String getDigestEncryptionAlgorithm() Obtiene el nombre del algoritmo utilizado en el proceso de firma

Regresa:Nombre del algoritmo utilizado en el proceso de firma

2.2.68 setDigestEncryptionAlgorithm

Sintaxis Java:public void setDigestEncryptionAlgorithm(java.lang.String digestEncryptionAlgorithm) Asigna el nombre del algoritmo utilizado en el proceso de firma

Parametros:

digestEncryptionAlgorithm - Nombre del algoritmo utilizado en el proceso de firma

2.2.69getEncryptedDigest

Sintaxis Java:public java.lang.String getEncryptedDigest() Obtiene la cadena que representa la firma del mensaje (base 64)

Regresa:Cadena que representa la firma del mensaje

2.2.70setEncryptedDigest

Sintaxis Java:public void setEncryptedDigest(java.lang.String encryptedDigest)Asigna la cadena que representa la firma del mensaje

Parametros:

encryptedDigest - Cadena que representa la firma del mensaje

2.2.71getEncryptedDigestBlock

Sintaxis Java:public byte[] getEncryptedDigestBlock()Obtiene el arreglo de bytes que conforman la firma

Regresa:Arreglo de bytes de la firma codificada en el signer info

2.2.72 setEncryptedDigestBlock

Sintaxis Java:public void setEncryptedDigestBlock(byte[] encryptedDigestBlock)Asigna el bloque binario de la firma codificada en el signer info

Parametros:

encryptedDigestBlock - Arreglo de bytes de la firma codificada en el signer info

2.2.73getIssuerAndSerialNumber

Sintaxis Java:public IssuerAndSerialNumber getIssuerAndSerialNumber()Obtiene el número de serie y nombres distinguidos del emisor, codificados en el signer info

Regresa:Objeto que contiene los nombres distinguidos y número de serie codificados en el signer info

2.2.74setIssuerAndSerialNumber

Sintaxis Java:public void setIssuerAndSerialNumber(IssuerAndSerialNumber issuerAndSerialNumber)Asigna los nombres distinguidos del emisor y el número de serie contenidos en el signer info

Parametros:

issuerAndSerialNumber - Instancia del objeto que contiene los datos contenidos en el bloque de emisor y número de serie

2.2.75getMessageDigest

Sintaxis Java:public java.lang.String getMessageDigest()Obtiene la huella digital del mensaje firmado

Regresa: Cadena hexadecimal que representa la huella digital del mensaje firmado o asegurado

2.2.76setEncryptedDigestBlock**Sintaxis Java:**

Sintaxis Java:public void setEncryptedDigestBlock(byte[] encryptedDigestBlock)Asigna el bloque binario de la firma codificada en el signer info

Parametros:

encryptedDigestBlock - Arreglo de bytes de la firma codificada en el signer info

2.2.77getIssuerAndSerialNumber

Sintaxis Java:public IssuerAndSerialNumber getIssuerAndSerialNumber()Obtiene el número de serie y nombres distinguidos del emisor, codificados en el signer info

Regresa:Objeto que contiene los nombres distinguidos y número de serie codificados en el signer info

2.2.78setIssuerAndSerialNumber

Sintaxis Java:public void setIssuerAndSerialNumber(IssuerAndSerialNumber issuerAndSerialNumber)Asigna los nombres distinguidos del emisor y el número de serie contenidos en el signer info

Parametros:

issuerAndSerialNumber - Instancia del objeto que contiene los datos contenidos en el bloque de emisor y número de serie

2.2.79getMessageDigest

Sintaxis Java:public java.lang.String getMessageDigest()Obtiene la huella digital del mensaje firmado

Regresa: Cadena hexadecimal que representa la huella digital del mensaje firmado o asegurado

2.2.80setVersion

Sintaxis Java:public void setVersion(int version)Asigna el valor de la versión codificada del signer info

Parametros:

version - Versión codificada del signer info

2.2.81toString

Sintaxis Java:public java.lang.String toString()Overrides:

toString in class java.lang.Object

2.2.82-getSignerInfoSet

Sintaxis Java:public java.util.ArrayList<SignerInfo> getSignerInfoSet()Retorna una colección de objetos que contienen detalles de los signer infos codificados en el mensaje

Regresa: Objetos que contienen los detalles de los signer info individuales

2.2.83setSignerInfoSet

public void setSignerInfoSet(java.util.ArrayList<SignerInfo> signerInfoSet)Asigna los signer infos del mensaje

Parametros:

signerInfoSet - Colección de objetos signer info

2.2.84toString

Sintaxis Java:public java.lang.String toString()Overrides:

toString in class java.lang.Object

2.2.85-AuthenticatePKCS7

Sintaxis Java:public int AuthenticatePKCS7(java.lang.String IP,

int Port,

java.lang.String Folio,

int FolioLen,

java.lang.String Serial,

int SerialLen,

java.lang.String FileName,

java.lang.String B64,

java.lang.String PKCS7,

int PKCS7Len,

java.lang.String ExternContent,

int ExternContentLen,

char GetReceipt)Deprecated.

Función: Conforma y procesa una petición por parte del cliente con el propósito de conocer el estatus de autenticación de un mensaje criptográfico PKCS7, en este caso el PKCS7 se pasa como un objeto de tipo String

Parametros:

IP - La dirección IP del servidor SeguriSign

Port - El puerto de comunicación que emplea el servidor SeguriSign para atender las peticiones. El valor por omisión es 7920

Folio - Número de Folio para la transacción y que puede ser relevante para el aplicativo que integra Firma Digital

FolioLen - Longitud del Folio del Aplicativo

Serial - Número de Serie del Certificado que se espera utilice el Firmante

SerialLen - Longitud del Número de Serie esperado

FileName - Nombre del archivo firmado procedente de la ejecución de algún módulo de Cliente de Firma (Applet o ActiveX). En caso que no se firmen archivos sino transacciones de texto, se recomienda ampliamente enviar un valor de nombre de archivo arbitrario con extensián txt.

B64 - Indica si la información firmada se encuentra en Base 64 ("TRUE" o "FALSE")

PKCS7 - Mensaje Criptográfico resultante del proceso de Firma (Applet o ActiveX)

PKCS7Len - Longitud del Mensaje Criptográfico anterior

ExternContent - Información Firmada para el caso de transacciones procedentes de un navegador Netscape 4.x, en los casos en que la firma provenga de otro navegador, su valor será "NONE"

ExternContentLen - Longitud de la información firmada indicada en el parámetro anterior

GetReceipt - Indica si se desea obtener una copia del Recibo Criptográfico asignado a la transacción (Y ó N)

Regresa: En una ejecucián exitosa el valor de retorno es 1, en caso contrario 0

2.2.86AuthenticatePKCS7

Sintaxis Java:public int AuthenticatePKCS7(java.lang.String IP,

int Port,

java.lang.String Folio,

int FolioLen,

java.lang.String Serial,

int SerialLen,

java.lang.String FileName,

java.lang.String B64,

byte[] PKCS7,

int PKCS7Len,

byte[] ExternContent,

int ExternContentLen,

char GetReceipt)Deprecated.

Función: Conforma y procesa una petición por parte del cliente con el propósito de conocer el estatus de autenticación de un mensaje criptográfico PKCS7, en este caso el PKCS7 se pasa como un arreglo de bytes

Parametros:

IP - Nombre o dirección IP del servidor SeguriSign

Port - El puerto de comunicación que emplea el servidor SeguriSign para atender las peticiones. El valor por omisión es 7920

Folio - Número de Folio para la transacción y que puede ser relevante para el aplicativo que integra Firma Digital

FolioLen - Longitud del Folio del Aplicativo

Serial - Número de Serie del Certificado que se espera utilice el Firmante

SerialLen - Longitud del Número de Serie esperado

FileName - Nombre del archivo firmado procedente de la ejecución de algún módulo de Cliente de Firma (Applet o ActiveX). En caso que no se firmen archivos sino transacciones de texto, se recomienda ampliamente enviar un valor de nombre de archivo arbitrario con extensián txt.

B64 - Indica si la información firmada se encuentra en Base 64 ("TRUE" o "FALSE")

PKCS7 - Mensaje Criptográfico resultante del proceso de Firma, en formato binario (Applet o ActiveX)

PKCS7Len - Longitud del Mensaje Criptográfico anterior

ExternContent - Información Firmada para el caso de transacciones procedentes de un navegador Netscape 4.x, en los casos en que la firma provenga de otro navegador, su valor será "NONE"

ExternContentLen - Longitud de la información firmada indicada en el parámetro anterior

GetReceipt - Indica si se desea obtener una copia del Recibo Criptográfico asignado a la transacción (Y ó N)

Regresa: En una ejecución exitosa el valor de retorno es 1, en caso contrario 0

2.2.87GetOriginalDocument

Sintaxis Java:public int GetOriginalDocument(java.lang.String IP,

int Port,

java.lang.String Sequence,

java.lang.String B64,

java.lang.String Directory)Deprecated.

Función: Solicita al servidor el documento original de la transacción que corresponde con el número de secuencia (Sequence) proporcionado como parámetro. El documento en caso de existir será depositado en el directorio indicado (Directory). Asimismo, será almacenado en memoria, y será posible recuperarlo según el método GetOriginalDocument(), el nombre del documento se conocerá por el método GetFileName() y su longitud en GetOriginalDocumentLen()

Parametros:

IP - La dirección IP del servidor SeguriSign

Port - El puerto de comunicación que emplea el servidor SeguriSign para atender las peticiones. El valor por omisión es 7920

Sequence - Secuencia de la Transacción de la que se obtendrá la información original asegurada

B64 - Indica si la información se obtendrá en binario o Base 64 ("FALSE" o "TRUE")

Directory - Directorio local al que se grabará el contenido del Mensaje Criptográfico según el nombre de Archivo registrado en el Servidor SeguriSign e indicado por el método GetFileName()

Regresa:En una ejecución exitosa el valor de retorno es 1, en caso contrario 0

2.2.88GetOriginalDocument

Sintaxis Java:public int GetOriginalDocument(java.lang.String IP,

int Port,

java.lang.String Sequence,

java.lang.String B64)Deprecated.

Función: Solicita al servidor el documento original de la transacción que corresponde con el número de secuencia (Sequence) proporcionado como parámetro. El documento en caso de existir será almacenado en memoria y se recuperará por el método GetOriginalDocument(), el nombre del documento se conocerá al ejecutar el método GetFileName() y su longitud de acuerdo a GetOriginalDocumentLen()

Parametros:

IP - La dirección IP del servidor SeguriSign

Port - El puerto de comunicación que emplea el servidor SeguriSign para atender las peticiones. El valor por omisión es 7920

Sequence - Secuencia de la Transacción de la que se obtendrá la información original asegurada

B64 - Indica si la información se obtendrá en binario o Base 64 ("FALSE" o "TRUE")

Regresa: En una ejecución exitosa el valor de retorno es 1, en caso contrario 0

2.2.89GetCryptographicData4Sequence

Sintaxis Java:public byte[] GetCryptographicData4Sequence(java.lang.String ip,
int port,
java.lang.String sequence,
char requestedCryptoData)Deprecated.

Función: Solicita una evidencia criptográfica asociada a la Secuencia SeguriSign especificada

Parametros:

ip - Dirección IP del Servidor SeguriSign al que se hará la solicitud

port - Puerto SeguriSign para peticiones

sequence - Secuencia SeguriSign de la que se solicita una evidencia criptográfica en particular

requestedCryptoData - Evidencia a solicitar al Servidor (1=recibo criptográfico; 2=respuesta ocsp; 3=mensaje criptográfico)

Regresa: (not null) Evidencia Criptográfica asociada a la Secuencia SeguriSign especificada, (null) Error

2.2.90getCryptographicData4Sequence

Sintaxis Java:public byte[] getCryptographicData4Sequence(java.lang.String server,
int port,
java.lang.String sequence,
char requestedCryptoData)Deprecated.

Función: Solicita una evidencia criptográfica asociada a la Secuencia SeguriSign especificada

Parametros:

server - Dirección IP o nombre del Servidor SeguriSign al que se hará la solicitud

port - Puerto SeguriSign para peticiones

sequence - Secuencia SeguriSign de la que se solicita una evidencia criptográfica en particular

requestedCryptoData - Evidencia a solicitar al Servidor (1=recibo criptográfico; 2=respuesta ocsp; 3=mensaje criptográfico)

Regresa: (not null) Evidencia Criptográfica asociada a la Secuencia SeguriSign especificada, (null) Error

2.2.91getCryptographicData4Sequence

Sintaxis Java:public byte[] getCryptographicData4Sequence(java.lang.String server,
int port,
java.lang.String sequence,
Process.EvidenceType requestedCryptoData)Deprecated.

Función: Solicita una evidencia criptográfica asociada a la Secuencia SeguriSign especificada

Parametros:

server - Dirección IP o nombre del Servidor SeguriSign al que se hará la solicitud

port - Puerto SeguriSign para peticiones

sequence - Secuencia SeguriSign de la que se solicita una evidencia criptográfica en particular

requestedCryptoData - Evidencia a solicitar al Servidor

Regresa: (not null) Evidencia Criptográfica asociada a la Secuencia SeguriSign especificada, (null) Error

2.2.92getSignerData

Sintaxis Java:public int getSignerData(byte[] mc)
java.lang.ExceptionDeprecated.

Función: Obtiene los Datos del Certificado del Firmante a partir de un Mensaje Criptográfico con firma unilateral generado con SeguriSign ActiveX o SeguriSign Applet

Parametros:

mc - Mensaje Criptográfico a decodificar

Regresa: Regresa 0 en caso de error y 1 en caso de decodificar con éxito el mensaje

Arroja:

java.lang.Exception - Error de lectura o parseo de certificado digital

2.2.93getSignerData

Sintaxis Java:public int getSignerData(byte[] mc,
java.lang.String hashAlg)
java.lang.ExceptionDeprecated.

Función: Obtiene los Datos del Certificado del Firmante a partir de un Mensaje Criptográfico con firma unilateral generado con SeguriSign ActiveX o SeguriSign Applet

Parametros:

mc - Mensaje Criptográfico a decodificar

hashAlg - Algoritmo de digestión para calcular la huella digital del certificado del firmante

Regresa: Regresa 0 en caso de error y 1 en caso de decodificar con éxito el Mensaje

Arroja:

java.lang.Exception - Error de lectura o parseo de certificado digital

2.2.94getCMCertificates

Sintaxis Java:public byte[][] getCMCertificates(byte[] mc,

java.lang.String hashAlg)Deprecated.

Función: Obtiene los certificados contenidos en un Mensaje Criptográfico Firmado unilateral o multilateralmente

Parametros:

mc - Mensaje Criptográfico a decodificar

hashAlg - Algoritmo de digestión para calcular la huella digital del certificado

Regresa: Regresa null en caso de error y un arreglo de arreglos de bytes conteniendo cada certificado, en caso de decodificar con éxito el Mensaje

2.2.95getCertData

Sintaxis Java:public int getCertData(byte[] certificate,

java.lang.String hashAlg)

java.lang.ExceptionDeprecated.

Función: Obtiene los Datos de un certificado Digital

Parametros:

certificate - Certificado Digital a decodificar

hashAlg - Algoritmo de digestión para calcular la huella digital del certificado

Regresa: Regresa 0 en caso de error y 1 en caso de decodificar con éxito el Certificado

Arroja:

java.lang.Exception - Error de parseo o lectura de certificado digital al procesarlo

2.2.96decodeTS

Sintaxis Java:public int decodeTS(byte[] ts)

java.lang.ExceptionDeprecated.

Función: Obtiene los Datos de una estampilla de Tiempo

Parametros:

ts - Estampa de Tiempo a decodificar

Regresa: Regresa 0 en caso de error y 1 en caso de decodificar con éxito la Estampa

Arroja:

java.lang.Exception - Error de parseo o lectura de estampa o certificado digital

2.2.97decodeTS

Sintaxis Java:public int decodeTS(byte[] ts,

java.lang.String hashAlg)

java.lang.ExceptionDeprecated.

Función: Obtiene los Datos de una estampilla de Tiempo

Parametros:

ts - Estampa de Tiempo a decodificar

hashAlg - Algoritmo de Digestión para obtener la huella digital del certificado

Regresa: Regresa 0 en caso de error y 1 en caso de decodificar con éxito la Estampa

Arroja:

java.lang.Exception - Error de parseo o lectura de estampa o certificado digital

2.2.98Hash

Sintaxis Javapublic java.lang.String Hash(java.lang.String toHash,

int toHashLen,

java.lang.String alg)Deprecated.

Función: Calcula una digestión de un archivo o de una cadena

Parametros:

toHash - Cadena a digerir o path del archivo a procesar

toHashLen - Longitud de la cadena a digerir, si su valor es cero indica que toHash es un path de archivo

alg - Algoritmo de digestión a utilizar para realizar el cálculo (MD2/MD4/MD5/SHA1)

Regresa: Cadena hexadecimal que representa la digestión calculada

2.2.99Hash

Sintaxis Javapublic java.lang.String Hash(byte[] toHash,

java.lang.String alg)Deprecated.

Función: Calcula una digestión de un arreglo de bytes

Parametros:

toHash - Arreglo de Bytes a procesar

alg - Algoritmo de digestión a utilizar para realizar el cálculo (MD2/MD4/MD5/SHA1)

Regresa: Cadena hexadecimal que representa la digestión calculada

2.2.100binaryHash

Sintaxis Java:public byte[] binaryHash(java.lang.String toHash,
int toHashLen,
java.lang.String alg)Deprecated.

Función: Calcula una digestión de un archivo o de una cadena

Parametros:

toHash - Cadena a digerir o path del archivo a procesar

toHashLen - Longitud de la cadena a digerir, si su valor es cero indica que toHash es un path de archivo

alg - Algoritmo de digestión a utilizar para realizar el cálculo (MD2/MD4/MD5/SHA1)

Regresa: Digestión calculada

2.2.101binaryHash

Sintaxis Java:public byte[] binaryHash(byte[] toHash,
java.lang.String alg)Deprecated.

Función: Calcula una digestión de un arreglo de bytes

Parametros:

toHash - Arreglo de Bytes a procesar

alg - Algoritmo de Digestión a utilizar para realizar el cálculo (MD2/MD4/MD5/SHA1)

Regresa: Digestión calculada

2.2.102multiSignedMessage_Init

Sintaxis Javapublic java.lang.String multiSignedMessage_Init(java.lang.String server,
int port,
byte[] data,
char dataType,

java.lang.String info)

java.io.IOException,

java.security.NoSuchAlgorithmExceptionDeprecated.

Función: Solicita al Servidor SeguriSign iniciar un proceso de firma multilateral. Como primer punto debe darse de alta la información a firmar digitalmente y a la que harán referencia las firmas digitales generadas

Parametros:

server - Dirección IP o nombre del servidor SeguriSign

port - Puerto de comunicación que emplea el servidor SeguriSign para atender las peticiones.

data - Corresponde a la información a dar de alta en el servidor como punto inicial de un proceso de firma multilateral

dataType - Indica si la información está contenida en el parámetro data, si se refiere a la ruta y nombre de archivo que corresponde a la información o es la huella digital de la información a procesar. 0 - Información a registrar; 1 - Ruta del archivo a registrar; 2 - Digestión del archivo (SHA1)

info - Información adjunta al archivo, pudiera utilizarse para registrar el nombre del archivo o el tipo de información de que se trata: imagen gif, archivo pdf, etc.

Regresa:

En una ejecución exitosa el valor de retorno corresponde al ID asociado a la informacián asegurada y por el que serán referenciadas las firmas digitales generadas como parte del proceso de firma multilateral. Si se presenta un error de procesamiento, el ID obtenido será una cadena vacía

Arroja:

java.io.IOException - Error de lectura de los datos proporcionados

java.security.NoSuchAlgorithmException - El algoritmo de digestión SHA1 no está disponible

2.2.103multiSignedMessage_Init

Sintaxis Java:public java.lang.String multiSignedMessage_Init(java.lang.String ip,

int port,

byte[] data,

char dataType,

java.lang.String info,

byte flags)

java.io.IOException,

java.security.NoSuchAlgorithmExceptionDeprecated.

Función: Solicita al Servidor SeguriSign iniciar un proceso de firma multilateral. Como primer punto debe darse de alta la información a firmar digitalmente y a la que harán referencia las firmas digitales generadas

Parametros:

ip - Nombre de servidor o dirección IP o nombre del servidor SeguriSign

port - Puerto de comunicación que emplea el servidor SeguriSign para atender las peticiones.

data - Corresponde a la información a dar de alta en el servidor como punto inicial de un proceso de firma multilateral

dataType - Indica si la información está contenida en el parámetro data, si se refiere a la ruta y nombre de archivo que corresponde a la informacián o es la huella digital de la información a procesar. 0 - Información a registrar; 1 - Ruta del archivo a registrar; 2 - Digestión del archivo (SHA1)

info - Información adjunta al archivo, pudiera utilizarse para registrar el nombre del archivo o el tipo de información de que se trata: imagen gif, archivo pdf, etc.

flags - Indica el tipo de proceso de firma a inicializar 0 - Proceso de firma multilateral estándar 1 - Proceso de firma XML

Regresa:

En una ejecución exitosa el valor de retorno corresponde al ID asociado a la informacián asegurada y por el que serán referenciadas las firmas digitales generadas como parte del proceso de firma multilateral. Si se presenta un error de procesamiento, el ID obtenido será una cadena vacía

Arroja:

java.io.IOException - Error de lectura de los datos proporcionados

java.security.NoSuchAlgorithmException - El algoritmo de digestión SHA1 no está disponible

2.2.104multiSignedMessage_Init

Sintaxis Java:public java.lang.String multiSignedMessage_Init(java.lang.String server,

int port,

byte[] data,

Process.DataType dataType,

java.lang.String info,

Process.ProcessType flags)

java.io.IOException,

java.security.NoSuchAlgorithmExceptionDeprecated.

Función: Solicita al Servidor SeguriSign iniciar un proceso de firma multilateral. Como primer punto debe darse de alta la información a firmar digitalmente y a la que harán referencia las firmas digitales generadas

Parametros:

server - Dirección IP o nombre del servidor SeguriSign

port - Puerto de comunicación que emplea el servidor SeguriSign para atender las peticiones.

data - Corresponde a la información a dar de alta en el servidor como punto inicial de un proceso de firma multilateral

dataType - Indica si la información está contenida en el parámetro data, si se refiere a la ruta y nombre de archivo que corresponde a la informacián o es la huella digital de la información a procesar.

info - Información adjunta al archivo, pudiera utilizarse para registrar el nombre del archivo o el tipo de información de que se trata: imagen gif, archivo pdf, etc.

flags - Indica el tipo de proceso de firma a inicializar

Regresa:

En una ejecución exitosa el valor de retorno corresponde al ID asociado a la información asegurada y por el que serán referenciadas las firmas digitales generadas como parte del proceso de firma multilateral. Si se presenta un error de procesamiento, el ID obtenido será una cadena vacía

Arroja:

java.io.IOException - Error de lectura de los datos proporcionados

java.security.NoSuchAlgorithmException - El algoritmo de digestión SHA1 no está disponible

2.2.105multiSignedMessage_Init

Sintaxis Java:public java.lang.String multiSignedMessage_Init(java.lang.String server,

int port,

byte[] data,

char dataType,

java.lang.String digAlg,

java.lang.String info,

byte flags)

java.io.IOException,

java.lang.IllegalArgumentException,

java.security.NoSuchAlgorithmExceptionDeprecated.

Función: Solicita al Servidor SeguriSign iniciar un proceso de firma multilateral. Como primer punto debe darse de alta la información a firmar digitalmente y a la que harán referencia las firmas digitales generadas

Parametros:

server - Dirección IP o nombre del servidor SeguriSign

port - Puerto de comunicación que emplea el servidor SeguriSign para atender las peticiones

data - Corresponde a la información a dar de alta en el servidor como punto inicial de un proceso de firma multilateral

dataType - Indica si la información fue parametrizada, o si este argumento se refiere a la ruta y nombre de archivo que corresponde a la información o es la huella digital del mensaje a asegurar 0 - Información a registrar; 1 - Ruta del archivo a registrar; 2 - Digestión del archivo

digAlg - Indica el algoritmo de digestión a utilizar para el proceso de firma (SHA1/SHA-256/SHA-384/SHA-512)

info - Información adjunta al archivo. Puede utilizarse para asociar al proceso de firma el nombre del archivo o el tipo de información de que se trata: imagen gif, archivo pdf, etc.

flags - Indica el tipo de proceso de firma a inicializar 0 - Proceso de firma multilateral estándar 1 - Proceso de firma XML

Regresa:

En una ejecución exitosa el valor de retorno corresponde a un objeto MultilateralInitTransaction con los detalles obtenidos para la transacción

Arroja:

java.io.IOException - Error de lectura de los datos proporcionados

java.lang.IllegalArgumentException - El argumento es nulo, vacío o su valor no es correcto

java.security.NoSuchAlgorithmException - El algoritmo especificado no se encuentra disponible

java.lang.Exception - Error de procesamiento o comunicación con el servidor especificado

2.2.106multiSignedMessage_Init

Sintaxis Java:public java.lang.String multiSignedMessage_Init(java.lang.String server,
int port,
byte[] data,
Process.DataType dataType,
java.lang.String digAlg,
java.lang.String info,
Process.ProcessType flags)

java.io.IOException,

java.lang.IllegalArgumentException,

java.security.NoSuchAlgorithmExceptionDeprecated.

Función: Solicita al Servidor SeguriSign iniciar un proceso de firma multilateral. Como primer punto debe darse de alta la información a firmar digitalmente y a la que harán referencia las firmas digitales generadas

Parametros:

server - Dirección IP o nombre del servidor SeguriSign

port - Puerto de comunicación que emplea el servidor SeguriSign para atender las peticiones

data - Corresponde a la información a dar de alta en el servidor como punto inicial de un proceso de firma multilateral

dataType - Indica si la información fue parametrizada, o si este argumento se refiere a la ruta y nombre de archivo que corresponde a la información o es la huella digital del mensaje a asegurar

digAlg - Indica el algoritmo de digestión a utilizar para el proceso de firma (SHA1/SHA-256/SHA-384/SHA-512)

info - Información adjunta al archivo. Puede utilizarse para asociar al proceso de firma el nombre del archivo o el tipo de información de que se trata: imagen gif, archivo pdf, etc.

flags - Indica el tipo de proceso de firma a inicializar

Regresa:

En una ejecución exitosa el valor de retorno corresponde a un objeto MultilateralInitTransaction con los detalles obtenidos para la transacción

Arroja:

java.io.IOException - Error de lectura de los datos proporcionados

java.lang.IllegalArgumentException - El argumento es nulo, vacío o su valor no es correcto

java.security.NoSuchAlgorithmException - El algoritmo especificado no se encuentra disponible

java.lang.Exception - Error de procesamiento o comunicación con el servidor especificado

2.2.107multiSignedMessage_Update

Sintaxis Java:public java.lang.String multiSignedMessage_Update(java.lang.String ip,

int port,

byte[] pkcs7,

java.lang.String id,

java.lang.String serial)Deprecated.

Función: Solicita al Servidor SeguriSign la autenticación de un mensaje Criptográfico para firma multilateral. El mensaje a autenticar debe ser sin contenido y estará relacionado con el punto anterior del proceso de firma

Parametros:

ip - Dirección IP o nombre del servidor SeguriSign

port - Puerto de comunicación que emplea el servidor SeguriSign para atender las peticiones.

pkcs7 - Mensaje criptográfico firmado y sin contenido que va a ser autenticado y asociado como parte del proceso de firma multilateral

id - Información original referida por medio del id asignado en el primer punto del proceso de firma

serial - Número de serie del certificado de firmante que se espera está presente en el mensaje criptográfico

Regresa:

En una ejecución exitosa el valor de retorno es la secuencia SeguriSign asignada a la autenticación de la firma actual, en caso de error en el procesamiento el valor de retorno será una cadena vacía

2.2.108multiSignedMessage_Final

Sintaxis Java:public byte[][] multiSignedMessage_Final(java.lang.String ip,

int port,

java.lang.String id,

char type)Deprecated.

Función: Solicita el Servidor SeguriSign el cierre de un proceso de firma multilateral, o que sea conformado un CMS con las firmas actuales

Parametros:

ip - Dirección IP o nombre del Servidor SeguriSign

port - Puerto de comunicación que emplea el Servidor

id - Identificador del proceso de firma multilateral a concluir

type - Tipo de operación solicitada. 0 - CMS con todas las firmas; no se cierra el proceso de firma multilateral y se solicita una estampa que no es conservada; 1 - CMS con todas las firmas; se cierra el proceso de firma multilateral y se solicita una estampa que es conservada y devuelta para el caso de conformación de CMS final; 2 - CMS con las firmas del entorno SeguriSign; se genera una estampa nueva; 3 - CMS con las firmas foráneas a SeguriSign; se genera una estampa nueva;

Regresa:

byte[][] Arreglo bidimensional que contiene: Mensaje firmado multilateralmente; Estampa de tiempo asociada; Dato adjunto que hace referencia del contenido asegurado

2.2.109multiSignedMessage_Final

Sintaxis Java:public byte[][] multiSignedMessage_Final(java.lang.String ip,

int port,

java.lang.String id,

Process.CMSType type)Deprecated.

Función: Solicita el Servidor SeguriSign el cierre de un proceso de firma multilateral, o que sea conformado un CMS con las firmas actuales

Parametros:

ip - Dirección IP o nombre del Servidor SeguriSign

port - Puerto de comunicación que emplea el Servidor

id - Identificador del proceso de firma multilateral a concluir

type - Tipo de operación solicitada

Regresa:

byte[][] Arreglo bidimensional que contiene: Mensaje firmado multilateralmente; Estampa de tiempo asociada; Dato adjunto que hace referencia del contenido asegurado

2.2.110multiSignedMessage_Status

Sintaxis Java:public char multiSignedMessage_Status(java.lang.String ip,

int port,

java.lang.String id)Deprecated.

Función: Solicita el Servidor SeguriSign el estatus de un proceso de firma multilateral.

Parametros:

ip - Dirección IP o nombre del Servidor SeguriSign

port - Puerto de comunicación que emplea el Servidor

id - Identificador del proceso de firma multilateral a reportar el estatus

Regresa:

status: 0 El método no se ejecutó correctamente 1 El proceso fue creado y se encuentra en espera de firmas 2 El proceso se encuentra abierto 3 El proceso se encuentra cerrado

2.2.111multiSignedMessage_AddSignature

Sintaxis Java:public java.lang.String multiSignedMessage_AddSignature(java.lang.String ip,

int port,

byte[] p7m,

byte[] tsp,

java.lang.String id)Deprecated.

Función: Asocia a un proceso de firma multilateral existente, una firma y sus evidencias adjuntas procedentes de un entorno distinto.

Parametros:

ip - La dirección IP del Servidor SeguriSign

port - El puerto de comunicación que empleas el Servidor

p7m - Mensaje criptográfico previemante autenticado y que debe formar parte de un proceso de firma multilateral.

tsp - Estampa de tiempo asociada al mensaje parametrizado

id - Identificador del proceso de firma al que se refiere el mensaje ingresado y sus evidencias asociadas.

Regresa:

(not null) Secuencia SeguriSign asignada al mensaje autenticado y agregado a un proceso de firma multilateral, (null) Error

2.2.112toLocalDate

Sintaxis Java:public java.util.Date toLocalDate(java.lang.String theDate,

java.lang.String theFormat)

java.lang.ExceptionDeprecated.

Función: Convierte una cadena de fecha UTC a un tipo de dato Date en tiempo local

Parametros:

theDate - Cadena con la fecha UTC a convertir a tiempo local

theFormat - Formato de fecha en el que se encuentra codificada la fecha parametrizada

Regresa: Fecha local

Arroja:

java.lang.Exception - Error al parsear la fecha parametrizada

2.2.113GetError

Sintaxis Java:public java.lang.String GetError()Deprecated.

Función: Regresa el mensaje que describe el error ocurrido

Regresa: Mensaje descriptivo del error ocurrido en la ejecución no exitosa del más reciente método ejecutado

2.2.114GetSequence

Sintaxis Java:public java.lang.String GetSequence()Deprecated.

Función: Regresa el número de secuencia que el servidor asignó a una transacción de autenticación

Regresa: Número de Secuencia asociada a la transacción de autenticación procesada

2.2.115GetReceipt

Sintaxis Java:public byte[] GetReceipt()Deprecated.

Función: Devuelve el recibo criptográfico solicitado según el último argumento del método AuthenticatePKCS7.

Regresa: Regresa la cadena que representa el Recibo Criptográfico solicitado según el último argumento del método AuthenticatePKCS7.

2.2.116GetFileName

Sintaxis Java:public java.lang.String GetFileName()Deprecated.

Función: Devuelve el Nombre de Archivo asociado a una transacción de Autenticación de Mensaje Criptográfico recuperada por el método GetOriginalDocument

Regresa: Nombre de Archivo asociado a la transacción solicitada

2.2.117GetOriginalDocument

Sintaxis Java:public byte[] GetOriginalDocument()Deprecated.

Función: Devuelve el contenido del Documento Original solicitado al Servidor

Regresa: Documento Original recuperado

2.2.118GetOriginalDocumentLen

Sintaxis Java:public int GetOriginalDocumentLen()Deprecated.

Función: Devuelve la longitud del Documento Original solicitado al Servidor

Regresa: Longitud del la información original recuperada

2.2.119SetCharSet

Sintaxis Java:public void SetCharSet(java.lang.String val)Deprecated.

Función: Asigna el valor parametrizado para controlar el CharSet para la codificación de Cadenas de Error

Parametros:

val - CharSet a utilizar en el manejo de cadenas de error

2.2.120SetTimeOut

Sintaxis Java:public void SetTimeOut(int t)Deprecated.

Función: Asigna la cantidad de tiempo en milisegundos que el socket permancerá esperando leer datos del canal

Parametros:

t - Tiempo en milisegundos definido como máximo a esperar antes de descartar la comunicación con el Servidor

2.2.121getNotAfter

Sintaxis Java:public java.util.Date getNotAfter()Deprecated.

Función: Obtiene la fecha de inicio del periodo de validez del certificado digital

Regresa:Fecha de inicio de validez del certificado digital

2.2.122getNotBefore

Sintaxis Java:public java.util.Date getNotBefore()Deprecated.

Función: Obtiene la fecha de expiración del periodo de validez del Certificado Digital

Regresa:Fecha de fin de validez del certificado digital

2.2.123getSignerNames

Sintaxis Java:public java.lang.String getSignerNames()Deprecated.

Función: Obtiene los nombres distinguidos del Firmante del Mensaje o Estampilla evaluados

Regresa: Cadena con los nombres distinguidos del sujeto del certificado de usuario

2.2.124getSignerIssuerNames

Sintaxis Java:public java.lang.String getSignerIssuerNames()Deprecated.

Función: Obtiene los nombres distinguidos del Emisor del Firmante del Mensaje o Estampilla evaluados

Regresa: Cadena con los nombres distinguidos del emisor del certificado de usuario

2.2.125getCertSerialNumber

Sintaxis Java:public java.lang.String getCertSerialNumber()Deprecated.

Función: Obtiene el Número de Serie del Certificado del Firmante

Regresa: Cadena que representa al número de serie del certificado de usuario

2.2.126getError

Sintaxis Java:public java.lang.String getError()Deprecated.

Función: Obtiene el enunciado que describe el mensaje de error ocurrido

Regresa: Enunciado descriptivo del error más reciente ocurrido para todo método ejecutado con estatus no exitoso

2.2.127getCertHash

Sintaxis Java:public java.lang.String getCertHash()Deprecated.

Función: Obtiene la huella digital del Certificado del Firmante

Regresa:Cadena que representa la huella digital del certificado del firmante

2.2.128gettspData

Sintaxis Java:public java.lang.String gettspData()Deprecated.

Función: Devuelve los datos obtenidos de la decodificación de la Estampilla de Tiempo

Regresa: Datos contenidos en la estampa decodificada

2.2.129getAdditionalData

Sintaxis Java:public java.lang.String getAdditionalData()Deprecated.

Función: Indica el emisor de la evidencia criptográfica solicitada al Servidor SeguriSign, en caso de que ésta sea un recibo criptográfico. Si la evidencia criptográfica solicitada es un mensaje criptográfico, indicará el nombre del archivo asegurado. Ver getCryptographicData4Sequence

Regresa: additionalData

2.2.130getMessageDigest

Sintaxis Java::public java.lang.String getMessageDigest()Deprecated.

Función: Digestión de la información dada de alta para un proceso de firma multilateral

Regresa: messageDigest;

2.2.131isEmptycryptographicMessage

Sintaxis Java:public boolean isEmptycryptographicMessage(byte[] mc)

java.lang.IllegalArgumentException,

java.lang.ExceptionDeprecated.

Función: Indica si el mensaje criptográfico parametrizado se encuentra en modo detached

Parametros:

mc - Mensaje criptográfico del que se determinará si posee o no contenido asegurado

Regresa: Verdadero si el mensaje firmado no posee el contenido asegurado

Arroja: java.lang.IllegalArgumentException - Los argumentos no fueron proporcionados como se esperaba

java.lang.Exception - El mensaje parametrizado no es parseable

2.2.132decodeTSP

Sintaxis Java:public TSPStamp decodeTSP(SSignEvidence evidence)

java.lang.ExceptionDeprecated.

Función: Obtiene los detalles de una estampa de tiempo

Parametros:

evidence - Evidencia que corresponde a una estampa TSP

Regresa:Datos de la estampa parametrizada

Arroja: java.lang.Exception - La evidencia parametrizada no es una estampa o no es parseable

2.2.133getMultilateralSignedMessageData

Sintaxis Java:public CMSSignedMessage getMultilateralSignedMessageData(byte[] cms,

java.lang.String id)

java.lang.IllegalArgumentException,

java.lang.ExceptionDeprecated.

Función: Obtiene los certificados contenidos en un Mensaje Criptográfico Firmado unilateral o multilateralmente

Parametros:

cms - Mensaje Criptográfico a decodificar

id - Identificador del proceso de firma multilateral al que corresponde el CMS parametrizado

Regresa: Regresa objeto con evidencias encontradas en CMS

Arroja:

java.lang.IllegalArgumentException - Los argumentos recibidos no fueron satisfactorios

java.lang.Exception

2.2.134getRDN

Sintaxis Java:public java.util.HashMap<java.lang.String,java.lang.String> getRDN(java.lang.String subjectNames)

java.lang.ExceptionDeprecated.

Función: Obtiene un mapa de datos para los nombres distinguidos de un certificado

Parametros:

subjectNames - Cadena con los nombres distinguidos obtenida de un objeto SSignCertificate

Regresa:Objeto con los datos de la cadena de nombres y que tienen como llave para localizar al objeto las etiquetas de los nombres distinguidos

Arroja:

java.lang.Exception - Los datos no pudieron ser procesados correctamente

2.2.135multiSignedMessage_Verify

Sintaxis Java:public boolean multiSignedMessage_Verify(java.lang.String server,

int port,

java.lang.String id,

byte[] data,

Process.DataType dataType)

java.io.IOException,

java.lang.IllegalArgumentException,

java.security.NoSuchAlgorithmException,

MismatchedDocumentException,

java.lang.ExceptionDeprecated.

Función: Solicita se verifique que la información parametrizada forma parte de un proceso de firma multilateral

Parametros:

server - Servidor SeguriSign

port - Puerto al servidor SeguriSign

id - Identificador del proceso de firma multilateral

data - Información a contrastar

dataType - Indica si la información está contenida en el parámetro data, si se refiere a la ruta y nombre de archivo que corresponde a la información o es la huella digital de la información a procesar.

Regresa: El documento corresponde al proceso indicado (true)

Arroja:

java.io.IOException - Error de lectura en los datos de entrada

java.lang.IllegalArgumentException - Se recibió un argumento nulo, vacío o incorrecto

java.security.NoSuchAlgorithmException - El algoritmo de digestión del proceso no está disponible

MismatchedDocumentException - El documento no corresponde al proceso indicado

java.lang.Exception - Error de comunicaciones, de proceso no existente o procesamiento

2.2.136setSerialNumber

Sintaxis Java:public void setSerialNumber(java.lang.String serialNumber)Asigna la representación en cadena del número de serie

Parametros:

serialNumber - Cadena hexadecimal para el valor en bytes del número de serie

2.2.137getSerialNumber

Sintaxis Java:public java.lang.String getSerialNumber()Obtiene el valor de la cadena que representa el número de serie

Regresa: Cadena hexadecimal que representa el número de serie del certificado

2.2.138setSubjectNames

Sintaxis Java:public void setSubjectNames(java.lang.String subjectNames)Asigna la cadena de los nombres distinguidos del sujeto

Parametros:

subjectNames - Cadena que representa los nombres distinguidos del sujeto

2.2.139getSubjectNames

Sintaxis Java:public java.lang.String getSubjectNames()Retorna los nombres distinguidos del sujeto

Regresa:Cadena que representa los nombres distinguidos del sujeto

2.2.140setIssuerNames

Sintaxis Java:public void setIssuerNames(java.lang.String issuerNames)Asigna los nombres distinguidos del emisor

Parametros:

issuerNames - Cadena que representa los nombres distinguidos del emisor

2.2.141getIssuerNames

Sintaxis Java:public java.lang.String getIssuerNames()Obtiene la cadena que representa los nombres distinguidos del emisor

Regresa:Cadena que representa los nombres distinguidos del emisor

2.2.142setThumbPrint

Sintaxis Java:public void setThumbPrint(java.lang.String thumbPrint)Asigna la cadena hexadecimal que representa la huella digital SHA1 del certificado

Parametros:

thumbPrint - Cadena hexadecimal que representa la huella digital SHA1 del certificado

2.2.143getThumbPrint

Sintaxis Java:public java.lang.String getThumbPrint()Retorna la huella digital del certificado

Regresa:Cadena hexadecimal para la huella digital del certificado

2.2.144setEncoding

Sintaxis Java:public void setEncoding(byte[] encoding)Asigna los bytes del certificado digital

Parametros:

encoding - Conjunto de bytes que conforman el certificado digital

2.2.145getEncoding

Sintaxis Java:public byte[] getEncoding()Obtiene el conjunto de bytes que conforman el certificado digital

Regresa: Arreglo de bytes que conforman el certificado digital

2.2.146setNotBefore

Sintaxis Java:public void setNotBefore(java.util.Date notBefore)Asigna el inicio de validez del certificado

Parametros:

notBefore - Fecha de inicio de validez

2.2.147getNotBefore

Sintaxis Java:public java.util.Date getNotBefore()Obtiene el inicio de validez del certificado

Regresa: Fecha de inicio de validez del certificado

2.2.148setNotAfter

Sintaxis Java:public void setNotAfter(java.util.Date notAfter)Asigna el fin de validez del certificado

Parametros:

notAfter - Fecha de fin de validez del certificado digital

2.2.149getNotAfter

Sintaxis Java:public java.util.Date getNotAfter()Obtiene el fin de validez del certificado

Regresa: Fecha de fin de validez del certificado

2.2.150toString

Sintaxis Java:public java.lang.String toString()Overrides:

toString in class java.lang.Object

2.2.151setData

Sintaxis Java:public void setData(byte[] data)Habilita el especificar el conjunto de bytes que conforman el documento

Parametros:

data - Arreglo de bytes del contenido del documento

2.2.152getData

Sintaxis Java:public byte[] getData()Retorna los bytes del contenido del documento

Regresa:Arreglo de bytes del contenido del documento

2.2.153setBase64

Sintaxis Java:public void setBase64(boolean base64)Especifica si el contenido se ha codificado como base 64

Parametros:

base64 - Especifica que los bytes de contenido se encuentran codificados como base 64 (true)

2.2.154isBase64

Sintaxis Java:public boolean isBase64()Retorna verdadero si el documento es base 64, falso en caso contrario

Regresa:Valor que especifica si el documento se definiï¿½ como base 64

2.2.155setName

Sintaxis Java:public void setName(java.lang.String name)Define un nombre de archivo para el documento y su contenido

Parametros:

name - Nombre asociado al contenido, puede referirse al contenido firmado

2.2.156getName

Sintaxis Java:public java.lang.String getName()Retorna el nombre de archivo asociado al documento y/o a su contenido

Regresa:Nombre asociado al documento y su contenido

2.2.157toString

Sintaxis Java:public java.lang.String toString()Overrides:

toString in class java.lang.Object

2.2.158setSequence

Sintaxis Java:public void setSequence(java.lang.String sequence)Asigna el valor de la secuencia SeguriSign

Parametros:

sequence - Secuencia SeguriSign a la que pertenece la evidencia

2.2.159getSequence

Sintaxis Java:public java.lang.String getSequence()Retorna el valor de la secuencia SeguriSign

Regresa: Valor de la secuencia SeguriSign

2.2.160setData

Sintaxis Java:public void setData(byte[] data)Asigna el conjunto de bytes que conforman la evidencia

Parametros:

data - Arreglo de bytes que conforman la evidencia

2.2.161getData

Sintaxis Java:public byte[] getData()Retorna el conjunto de bytes que conforman la evidencia

Regresa:Conjunto de bytes que conforman la evidencia

2.2.162setEvidenceType

Sintaxis Java:public void setEvidenceType(java.lang.String evidenceType)Asigna el tipo de evidencia de que se trata: "TSP", "OCSP", "PKCS7"

Parametros:

evidenceType - Tipo de evidencia: "TSP", "OCSP", "PKCS7"

2.2.163getEvidenceType

Sintaxis Java:public java.lang.String getEvidenceType()Retorna el tipo de evidencia de que se trata

Regresa: Tipo de evidencia asignado: "TSP", "OCSP", "PKCS7"

2.2.164setBase64

Sintaxis Java:public void setBase64(boolean base64)Asigna el valor que determina si la evidencia se encuentra codificada en base 64

Parametros:

base64 - (true) Si la evidencia se encuentra codificada en base64; (false) Cuando la evidencia es binaria

2.2.165isBase64

Sintaxis Java:public boolean isBase64()Retorna el valor que indica si la evidencia se encuentra o no codificada en base 64

Regresa: (true) Si la evidencia se encuentra codificada en base64; (false) Cuando la evidencia es binaria

2.2.166getInfo

Sintaxis Java:public java.lang.String getInfo()Obtiene la información asociada a la evidencia (cuando aplica)

Regresa: Información asociada a la evidencia

2.2.167setInfo

Sintaxis Java:public void setInfo(java.lang.String info)Asigna la información asociada a la evidencia

Parametros:

info - Información asociada a la evidencia (PKCS#7)

2.2.168toString

Sintaxis Java:public java.lang.String toString()Overrides:

toString in class java.lang.Object

2.2.169setSequence

Sintaxis Java:public void setSequence(java.lang.String sequence)Asigna el secuencial para la estampa de tiempo

Parametros:

sequence - Identificador de la estampa

2.2.170getSequence

Sintaxis Java:public java.lang.String getSequence()Retorna el identificador de la estampa

Regresa:Identificador de la estampa

2.2.171setPolicy

Sintaxis Java:public void setPolicy(java.lang.String policy)Asigna el valor de la política asociada a la estampa

Parametros:

policy - Valor de la política asociada a la estampa

2.2.172getPolicy

Sintaxis Java:public java.lang.String getPolicy()Devuelve el valor de la política asociada a la estampa

Regresa:Valor de la política asociada a la estampa

2.2.173setHash

Sintaxis Java:public void setHash(java.lang.String hash)Asigna la huella digital asociada a la estampa

Parametros:

hash - Huela digital asociada a la estampa

2.2.174getHash

Sintaxis Java:public java.lang.String getHash()Retorna el valor de la huella digital asociada a la estampa

Regresa:Valor de la huella digital asociada a la estampa

2.2.175setSigner

Sintaxis Java:public void setSigner(SSignCertificate signer)Asigna el certificado del emisor de la estampa

Parametros:

signer - Retorna el certificado del emisor de la estampa

2.2.176getSigner

Sintaxis Java:public SSignCertificate getSigner()Retorna el certificado de firmante de la estampa

Regresa: Certificado de firmante de la estampa

2.2.177setIssuingDate

Sintaxis Java:public void setIssuingDate(java.util.Date issuingDate)Asigna la fecha de emisión de la estampa

Parametros:

issuingDate - Fecha de emisión de la estampa

2.2.178getIssuingDate

Sintaxis Java:public java.util.Date getIssuingDate()Retorna el valor de la fecha en que fue emitida la estampa

Regresa: Valor de la fecha en que fue emitida la estampa

2.2.179setEncoding

Sintaxis Java:public void setEncoding(byte[] encoding)Asigna el conjunto de bytes que conforman la estampa

Parametros:

encoding - Contenido binario de la estampa

2.2.180getEncoding

Sintaxis Java:public byte[] getEncoding()Retorna el conjunto de bytes que conforman la estampa

Regresa:Contenido binario de la estampa

2.2.181toString

Sintaxis Java:public java.lang.String toString()Overrides:

toString in class java.lang.Object

2.2.182decode

Sintaxis Java:public TSPStamp decode(SSignEvidence tspStamp,

ParameterTypes.DigestAlgorithm digestAlgorithm)

java.lang.IllegalArgumentException,

java.lang.ExceptionObtiene los detalles de una estampilla de tiempo TSP

Parametros:

tspStamp - Evidencia de tipo TSP

digestAlgorithm - Algoritmo de digestión para generar huellas digitales

Regresa: Objeto con detalles de la estampa decodificada

Arroja: java.lang.IllegalArgumentException - Alguno de los argumentos recibidos es nulo o sin valor

java.lang.Exception - Error de procesamiento de la estampa parametrizada

2.2.183decode

Sintaxis Java:public TSPStamp decode(byte[] tspStamp,

ParameterTypes.DigestAlgorithm digestAlgorithm)

java.lang.IllegalArgumentException,

java.lang.ExceptionObtiene los detalles de una estampilla de tiempo TSP

Parametros:

tspStamp - Conjunto de bytes que conforman la estampilla de tiempo TSP

digestAlgorithm - Algoritmo de digestión para generar huellas digitales

Regresa:Objeto con detalles de la estampa decodificada

Arroja: java.lang.IllegalArgumentException - Alguno de los argumentos recibidos es nulo o sin valor

java.lang.Exception - Error de procesamiento de la estampa parametrizada

2.2.184authenticateCryptographicMessage

Sintaxis Java public SSignEvidence authenticateCryptographicMessage(SSignDocument cryptographicMessage,

SSignDocument externContent,

java.lang.String folio,

java.lang.String serial)

Arroja: java.lang.IllegalArgumentException,

java.lang.ExceptionAuxilia en el proceso de autenticación de mensaje de proceso de firma unilateral

Parametros:

cryptographicMessage - Mensaje asegurado para proceso de firma unilateral

externContent - Contenido externo para mensaje firmado en modo 'detached'

folio - Cadena que vincula el aplicativo con una firma unilateral

serial - Habilita el validar que un certificado de firmante con cierto nÃºmero de serie es el utilizado para el proceso de firma

Regresa:Detalles de transacción de autenticación de mensaje

Arroja: java.lang.IllegalArgumentException - Alguno de los argumentos es nulo o no tiene valor

java.lang.Exception - Error durante el procesamiento del mensaje

2.2.185getCryptographicEvidence

Sintaxis Java:public SSignEvidence getCryptographicEvidence(java.lang.String sequence,

ParameterTypes.EvidenceType requestedEvidence)

java.lang.IllegalArgumentException,

java.lang.ExceptionSolicita una evidencia asociada a un proceso de firma

Parametros:

sequence - Identificador de la firma autenticada

requestedEvidence - Evidencia solicitada

Regresa: Objeto con los bytes que conforman la evidencia solicitada

Arroja: java.lang.IllegalArgumentException - Algún parámetro es nulo o sin valor

java.lang.Exception - Error de procesamiento del mensaje

2.2.186getOriginalDocument

Sintaxis Java:public SSignDocument getOriginalDocument(java.lang.String sequence)

java.lang.IllegalArgumentException,

java.lang.ExceptionAuxilia en el proceso de recuperación del mensaje original asegurado

Parametros:

sequence - Identificador de la transacción de autenticación a recuperar su contenido original

Regresa:Documento original asegurado

Arroja: java.lang.IllegalArgumentException - Alguno de los argumentos es nulo o no tiene valor

java.lang.Exception - Error durante el procesamiento del mensaje

2.2.187getOriginalDocument

Sintaxis Java:public boolean getOriginalDocument(java.lang.String sequence,

java.lang.String directory)

java.lang.IllegalArgumentException,

java.lang.ExceptionRecupera el contenido de un mensaje asegurado y lo salva en el directorio especificado

Parametros:

sequence - Identificador de la transacción de autenticación a recuperar su contenido original

directory - Directorio para recuperar el mensaje original

Regresa:true cuando el mensaje fue recuperado de manera correcta

Arroja: java.lang.IllegalArgumentException - Alguno de los argumentos es nulo o vacío

java.lang.Exception - Se presentaron errores de comunicaciones con el servidor, existencia del proceso o procesamiento de la información original recuperada

2.2.188getPort

Sintaxis Java:public int getPort()Obtiene el valor asignado al puerto del servidor SeguriSign

Regresa: Puerto del servidor SeguriSign

2.2.189getServer

Sintaxis Java:public java.lang.String getServer()Obtiene el nombre o ip del servidor SeguriSign

Regresa: Nombre o ip del servidor SeguriSign

2.2.190getSignedMessageData

Sintaxis Java:public com.seguridata.segurisign.beans.PKCS7Message getSignedMessageData(SSignEvidence signedMessage)

java.lang.IllegalArgumentException,

java.lang.ExceptionObtiene los detalles de un mensaje de firma unilateral

Parametros:

signedMessage - Evidencia correspondiente a un mensaje criptográfico

Regresa:Detalles obtenidos para el mensaje parametrizado

Arroja: java.lang.IllegalArgumentException - Alguno de los argumentos es nulo o sin valor

java.lang.Exception - Error al procesar el mensaje criptográfico

2.2.191getTimeOut

Sintaxis Java:public int getTimeOut()Obtiene el tiempo definido en milisegundos para espera en procesos de entrada y salida de datos en comunicaciones con el servidor

Regresa: Tiempo en milisegundos

2.2.192setPort

Sintaxis Java:public void setPort(int port)Asigna el valor del puerto del servidor con el que se manejarán las transacciones

Parametros: port - Puerto del servidor SeguriSign

2.2.193setServer

Sintaxis Java:public void setServer(java.lang.String server)Asigna el nombre o ip del servidor SeguriSign con el que se gestionarán los procesos de firma

Parametros: server - Nombre o ip del servidor SeguriSign

2.2.194setTimeOut

Sintaxis Java:public void setTimeOut(int timeOut)Contiene la cantidad de tiempo en milisegundos para espera de entrada de datos para comunicaciones con el servidor

Parametros: timeOut - Tiempo en milisegundos