Descripción de la sintaxis para la invocación por protocolo del Cliente @firma

# Parámetros generales

El protocolo tendrá siempre forma de URI, con la siguiente organización de componentes (según la sintaxis común [scheme://][authority][path][?query][#fragment]):

* [scheme://] Esquema del protocolo:
  + afirma
* [authority] Autoridad del protocolo:
  + Únicamente se admite la sección *host*, no puede contener ni puerto ni información de usuario
  + Autoridades válidas como host de la autoridad del protocolo:
    - sign
      * Indica una operación de firma.
    - cosign
      * Indica una operación de cofirma.
    - countersign
      * Indica una operación de contrafirma.
    - save
      * Indica una operación de guardado de datos.
* [path] Ruta del protocolo:
  + No se admite que figuren rutas en el protocolo.
* [?query] Consulta del protocolo:
  + Debe contener los parámetros necesarios para cada operación (indicada en la autoridad del protocolo).
* [#fragment] Fragmento del protocolo:
  + No se admite que la URI contenga fragmentos.

# Parámetros admitidos dentro de la consulta para las distintas operaciones

Estos son los parámetros que configuran cada una de las operaciones. Los parámetros siempre se proporcionarán en codificación de URL (“*URL Encoded*”).

Existe la posibilidad de que la longitud total de la URL sobrepase el límite establecido por el navegador desde el que se realice la invocación del Cliente. En ese caso, se

Los parámetros que pueden utilizarse catalogados por las operaciones disponibles son:

## save - Operación de guardado

Parámetros admitidos:

* id
  + Identificador de sesión aleatorio.
* dat
  + Datos a guardar, cifrados con el algoritmo DES/PADDING, usando la clave indicada en el parámetro key, y codificados en Base64 con las siguientes peculiaridades:
    - No contendrá espacios, ni tabuladores ni retorno de carro.
    - Se sustituirán los siguientes caracteres dentro del diccionario:
      * “+” pasa a ser “-“.
      * “/” pasa a ser “\_”.
  + PADDING: El relleno de los datos se realiza con 8 caracteres ‘\0’.
* key
  + Clave de cifrado de 8 caracteres.
* title
  + Título del diálogo de guardado.
* filename
  + Nombre propuesto para el fichero que contendrá los datos a guardar.
* exts
  + Lista de extensiones, separadas por coma, propuestas para el fichero.
* desc
  + Descripción del tipo de fichero.
* stservlet
  + URL del servicio de almacenamiento de datos.
* metro
  + Opcional. Booleano (“true” o “false”) usado en Windows 8 para indicar si estamos en la interfaz de nueva generación o no.

## sign - Operación de firma

Parámetros admitidos:

* id
  + Identificador de sesión aleatorio.
* dat
  + Datos a firmar, codificados en Base64 y a su vez con las siguientes peculiaridades:
    - No contendrá espacios, ni tabuladores ni retorno de carro.
    - Se sustituirán los siguientes caracteres dentro del diccionario:
      * “+” pasa a ser “-“.
      * “/” pasa a ser “\_”.
* key
  + Clave de cifrado de 8 caracteres.
* format
  + Formato de firma.
* algorithm
  + Algoritmo de firma.
* properties
  + Parámetros de configuración específicos del formato de firma en base 64.
* stservlet
  + URL del servicio de almacenamiento de datos.
* metro
  + Opcional. Booleano (“true” o “false”) usado en Windows 8 para indicar si estamos en la interfaz de nueva generación o no.

## cosign - Operación de cofirma o firma en paralelo

Parámetros admitidos:

* id
  + Identificador de sesión aleatorio.
* dat
  + Firma a cofirmar, codificada en Base64 con las siguientes peculiaridades:
    - No contendrá espacios, ni tabuladores ni retorno de carro.
    - Se sustituirán los siguientes caracteres dentro del diccionario:
      * “+” pasa a ser “-“.
      * “/” pasa a ser “\_”.
  + La firma debe ser implícita o estar realizada con el mismo algoritmo de firma que se desea utilizar con la cofirma.
* key
  + Clave de cifrado de 8 caracteres.
* format
  + Formato de firma.
* algorithm
  + Algoritmo de firma.
* properties
  + Parámetros de configuración específicos del formato de firma en base 64.
* stservlet
  + URL del servicio de almacenamiento de datos.
* metro
  + Opcional. Booleano (“true” o “false”) usado en Windows 8 para indicar si estamos en la interfaz de nueva generación o no.

## contersign - Operación de contrafirma o firma en cascada

Parámetros admitidos:

* id
  + Identificador de sesión aleatorio.
* dat
  + Firma a contrafirmar, codificada en Base64 con las siguientes peculiaridades:
    - No contendrá espacios, ni tabuladores ni retorno de carro.
    - Se sustituirán los siguientes caracteres dentro del diccionario:
      * “+” pasa a ser “-“.
      * “/” pasa a ser “\_”.
  + La firma debe ser implícita o estar realizada con el mismo algoritmo de firma que se desea utilizar con la contrafirma.
* key
  + Clave de cifrado de 8 caracteres.
* format
  + Formato de firma.
* algorithm
  + Algoritmo de firma.
* properties
  + Parámetros de configuración específicos del formato de firma en base 64.
* stservlet
  + URL del servicio de almacenamiento de datos.
* metro
  + Opcional. Booleano (“true” o “false”) usado en Windows 8 para indicar si estamos en la interfaz de nueva generación o no.

# Alternativa de invocación para datos pesados

Según el cliente o navegador usado para realizar la invocación al cliente móvil, es posible que la URL utilizada exceda la longitud máxima que permite. En ese caso, es posible enviar los datos a un servicio de almacenamiento temporal disponible con el propio cliente móvil. Al cliente sólo se le pasarán los datos necesarios para que los descargue y los procese.

El límite de los datos que se pueden enviar sin este mecanismo está determinado por el cliente que realiza la invocación, así que se deberán probarlos cliente concretos que harán uso del servicio o usar siempre este mecanismo de tratamiento de datos pesados.

En el caso de desear enviar datos pesados, los estructuraremos en forma de fichero XML con el formato:

<COD\_OPERACION>

<e k=”CLAVE1” v=”VALOR1” />

<e k=”CLAVE2” v=”VALOR2” />

…

<e k=”CLAVEn” v=”VALORn” />

</COD\_OPERACION>

En donde:

* *COD\_OPERACION*: Código de la operación que se configura.
* *CLAVEx*: Clave del listado de parámetros de configuración.
* *VALORx*: Valor correspondiente a la clave con el mismo número.

Un ejemplo de XML es:

<sign>

<e k="id" v="000987508929"/>

<e k="key" v="39772569"/>

<e k="stservlet" v="http://192.168.177.173:8080/afirma-signature-storage/StorageService"/>

<e k="format" v="CAdES"/>

<e k="algorithm" v="SHA1withRSA"/>

<e k="properties" v="c2VydmVyVXJsPWh0dHA6Ly8xOTIuMTY4LjE3Ny4xNzM6ODA4MC9hZmlybWEtc2VydmVyLXRyaXBoYXNlLXNpZ25lci9TaWduYXR1cmVTZXJ2aWNlCgkJIA%3D%3D"/>

<e k="dat" v="ZnVsYW5pdG9AYXRvcy5uZXQ%3D"/>

</sign>

Este XML se enviará, codificado en base 64 al mismo servicio de almacenamiento cuya ruta se establece a través del parámetro “stservlet”. Esta cadena a su vez estará cifrada y codificada en base 64.

Revise el apartado de almacenamiento de datos para conocer cómo realizar el envío del XML con la configuración de la operación.

Una vez enviado el XML se deberá invocar la app con los parámetros necesarios para que cargue la configuración de la operación. Los parámetros que deberán realizarse para la invocación son:

* fileId
  + Identificador asignado al fichero de configuración en el servicio remoto.
* rtservlet
  + URL del servicio para la recuperación del fichero de configuración.
* key
  + Clave DES para el descifrado del fichero de configuración.

Sintaxis del servicio de almacenamiento de datos

El servicio de almacenamiento de datos es uno de los 2 servicios web de tipo REST en los que se sustenta la arquitectura de uso del Cliente @firma móvil. Este servicio se encarga de almacenar datos temporales en un servidor remoto (proporcionado por el mismo dueño del servicio) que hace de pasarela de datos entre la app nativa y la página web en la que se integra el MiniApplet de @firma.

El almacenamiento de datos se realiza mediante la llamada a la URL del servicio junto con varios parámetros de configuración. Los parámetros necesarios son:

* op
  + Operación que se desea realizar. Se establecerá el valor: put
* v
  + Versión del servicio. Actualmente se debe configurar el valor: 1\_0
* id
  + Identificador aleatorio asignado a los datos que se almacenan.
* dat
  + Datos que se desean almacenar en base 64.

Por ejemplo:

http://miapp.com/SignatureStorageServer**/StorageService**?**op**=put&**v**=1\_0&**id**=000987508929&**dat**=ZnVsYW5pdG9AYXRvcy5uZXQ%3D

Sintaxis del servicio de recuperación de datos

El servicio de recuperación de datos es uno de los 2 servicios web de tipo REST en los que se sustenta la arquitectura de uso del Cliente @firma móvil. Este servicio permite recuperar datos temporales de un servidor remoto (proporcionado por el mismo dueño del servicio) que hace de pasarela de datos entre la app nativa y la página web en la que se integra el MiniApplet de @firma.

La recuperación de estos datos se realiza mediante la llamada a la URL del servicio junto con varios parámetros de configuración. Los parámetros necesarios son:

* op
  + Operación que se desea realizar. Se establecerá el valor: get
* v
  + Versión del servicio. Actualmente se debe configurar el valor: 1\_0
* id
  + Identificador que se utilizó para el almacenamiento de los datos.

Por ejemplo:

http://miapp.com/SignatureRetrieverServer**/RetrieveService**?**op**=get&**v**=1\_0&**id**=000987508929