Simple-SecREST

Ing. Ricardo Naranjo Faccini, M.Sc.

Para Skina IT Solutions

Versión 1.0

Miércoles, 20 de Octubre de 2021

Licensing

Tabla de contenidos

Indice de namespaces....1
Indice jerárquico....2
Índice de estructura de datos....3
Indice de archivos....4
skinaitCodificador....5
skinaitREST....6
Documentación de las estructuras de datos....7
skinaitRESTclienteREST....7
skinaitCodificadorcodificador....11
skinaitRESTconexionREST....16
mi_cliente....17
mi_servicio....19
skinaitRESTservidorREST....22
Documentación de archivos....27
cifrado_RSA.php....27
cliente.php....28
vista del cliente REST....28
herramientas.php....30
REST segura....30
mi_cliente.php....32
mi_servicio.php....33
REST.php....34
servicio.php....35
vista del servidor REST....35

Indice de namespaces

Lista de 'namespaces'

Lista de los 'namespaces', con una breve descripción:

skinait\Codificador 5
skinait\REST 6

Indice jerárquico

Jerarquía de la clase

Esta lista de herencias esta ordenada aproximadamente por orden alfabético:

- skinait\Codificador\codificador 11
- skinait\REST\conexionREST 16
  - skinait\REST\clienteREST 7
    - mi_cliente 17
- skinait\REST\servidorREST 22
  - mi_servicio 19

Índice de estructura de datos

Estructura de datos

Lista de estructuras con una breve descripción:

skinait\REST\clienteREST 7
skinait\Codificador\codificador 11
skinait\REST\conexionREST 16
mi_cliente 17
mi_servicio 19
skinait\REST\servidorREST 22

Indice de archivos

Lista de archivos

Lista de todos los archivos con descripciones breves:

cifrado_RSA.php 27
cliente.php 28
herramientas.php 30
mi_cliente.php 32
mi_servicio.php 33
REST.php 34
servicio.php 35

Documentación de namespaces

Referencia del Namespace skinait\Codificador

Estructuras de datos

class codificador

Referencia del Namespace skinait\REST

Estructuras de datos

class conexionREST
class clienteREST
class servidorREST

Documentación de las estructuras de datos

Referencia de la Clase skinait\REST\clienteREST

Diagrama de herencias de skinait\REST\clienteREST

Métodos públicos

__construct ($url, $login, $hclave, $fecha, $directorio, $depurando=false, $duracion=300)
Constructora, inicializa los campos del objeto a sus valores por defecto.

guardar_sesion_servicio ($url, $sesion, $pkey)
Esta función debe entregar el último id de sesión establecido con el servidor si se tiene disponible.

solicitar ($metodo, $parametros=array())
Solicita la ejecución de un método al servidor enviándo los parámetros indicados.

Campos de datos

$url
El localizador del servicio con el que se establecerá conexión.

$login
El login.

$hclave
El sha512 de la clave.

$fecha
Fecha y hora del momento en que se estableció la conexión.

Métodos protegidos

cargar_sesion_servicio ($url)
actualizar_llave_publica_de_cliente ($url)
Esta función debe almacenar el id de sesión indicado asociado con el servidor.

actualizar_ll_srv ($url)
Solicita al servidor la llave pública de cifrado que actualmente está utilizando.

get_directorio ()
Entrega la ruta al directorio donde se almacenan las llaves.

get_duracion ()
Entrega la duración en segundos de las llaves.

Otros miembros heredados

Descripción detallada

Clase para el cliente de integración de componentes via REST.

Definición en la línea 67 del archivo REST.php.

Documentación del constructor y destructor

skinait\REST\clienteREST::__construct ($url, $login, $hclave, $fecha, $directorio, $depurando = false, $duracion = 300)

Constructora, inicializa los campos del objeto a sus valores por defecto.

Parámetros

$url	El localizador del servicio con el que se establecerá conexión.
$login	El login de acceso.
$hclave	La huella SHA512 de la clave de acceso.
$fecha	La fecha y hora de cuando se estableció la conexión.
$directorio	El directorio donde se pueden almacenar las llaves.
$duracion	La cantidad de segundos que pueden transcurrir sin que se deban refrescar las llaves de cifrado.

Definición en la línea 80 del archivo REST.php.

Documentación de las funciones miembro

skinait\REST\clienteREST::actualizar_ll_srv ($url)[protected]

Solicita al servidor la llave pública de cifrado que actualmente está utilizando.

ENTRADAS:

Parámetros

$url	El localizador del servicio con el que se establecerá conexión.

Definición en la línea 217 del archivo REST.php.

skinait\REST\clienteREST::actualizar_llave_publica_de_cliente ($url)[protected]

Esta función debe almacenar el id de sesión indicado asociado con el servidor.

Ésta función existe para minimizar el tráfico asociado con el envío de llaves públicas entre cliente y servidor.

Debe ser implementada de acuerdo con la lógica particular de cada sistema de información.

Parámetros

$url	El localizador del servicio con el que se estableció conexión.
$sesion	El identificador de sesión que se va a asociar.
$pkey	La llave pública del servidor con la cual se cifrarán los mensajes.

Se envía al servidor la llave pública actualizada firmada con la antigua llave privada.

Parámetros

$url	El localizador del servicio con el que se establecerá conexión.

Definición en la línea 193 del archivo REST.php.

skinait\REST\clienteREST::cargar_sesion_servicio ($url)[abstract], [protected]

Reimplementado en mi_cliente (p.17).

skinait\REST\clienteREST::get_directorio ()[protected]

Entrega la ruta al directorio donde se almacenan las llaves.

Definición en la línea 257 del archivo REST.php.

skinait\REST\clienteREST::get_duracion ()[protected]

Entrega la duración en segundos de las llaves.

Definición en la línea 267 del archivo REST.php.

skinait\REST\clienteREST::guardar_sesion_servicio ($url, $sesion, $pkey)[abstract]

Esta función debe entregar el último id de sesión establecido con el servidor si se tiene disponible.

Ésta función existe para minimizar el tráfico asociado con el envío de llaves públicas entre cliente y servidor.

Debe ser implementada de acuerdo con la lógica particular de cada sistema de información.

Parámetros

$url	El localizador del servicio con el que se está estableciendo conexión. SALIDA El último id de sesión o null si no se tiene. La llave pública con la que se están cifrando los mensajes.

Reimplementado en mi_cliente (p.18).

skinait\REST\clienteREST::solicitar ($metodo, $parametros = array())

Solicita la ejecución de un método al servidor enviándo los parámetros indicados.

ENTRADAS

Parámetros

$metodo	El método que se solicitará al servidor.
$parametros	Arreglo asociativo con los parámetros requeridos.

Definición en la línea 277 del archivo REST.php.

Documentación de los campos

skinait\REST\clienteREST::$fecha

Fecha y hora del momento en que se estableció la conexión.

Definición en la línea 71 del archivo REST.php.

skinait\REST\clienteREST::$hclave

El sha512 de la clave.

Definición en la línea 70 del archivo REST.php.

skinait\REST\clienteREST::$login

El login.

Definición en la línea 69 del archivo REST.php.

skinait\REST\clienteREST::$url

El localizador del servicio con el que se establecerá conexión.

Definición en la línea 68 del archivo REST.php.

La documentación para esta clase fue generada a partir del siguiente fichero:

REST.php

Referencia de la Clase skinait\Codificador\codificador

Métodos públicos

__construct ($duracion=300, $directorio=null, $nombre_base="rsa_key", $tipo_huella="sha512", $depurando=false)
Constructora, inicializa los campos del objeto a sus valores por defecto.

get_duracion ()
Entregar la duración establecida de las llaves en segundos.

get_ll_publica ()
Entregar la llave pública vigente.

get_ll_pub_old ()
Entregar la llave pública que pronto expirará.

get_refrescar ()
Informa el valor del centinela que sugiere refrescar las llaves dado que se detectó que se está utilizando una próxima a expirar.

cargar_llaves_RSA ()
Si las llaves están almacenadas en los archivos correspondientes las carga desde los archivos.

firmar ($mensaje, $sellado=false, $usar_old=false)
Se utiliza para cifrar con la llave privada un mensaje que cualquier persona podrá descifrar con la llave pública que se compartirá.

descifrar ($mensaje)
Se utiliza cuando llega un mensaje de un emisor externo que tomó la llave pública de éste objeto para cifrar un mensaje.

borrarLlavesViejas ()
Revisa la antiguedad de las llaves almacenadas en el directorio con las llaves pública y privada, si son más viejas que lo indicado en "duración" se eliminan los archivos.

Métodos públicos estáticos

static validar_firma ($mensaje, $firma, $ll_publica, $tamanho=2048, $tipo_huella="sha512")
Se utiliza cuando llega un mensaje de un emisor externo del cual se conoce la llave pública para descifrarlo.

static cifrar ($mensaje, $ll_publica, $tamanho=2048)
Se utiliza cuando se quiere enviar un mensaje confidencial a un receptor.

Métodos protegidos

crear_llaves_RSA ()
Generacion de llaves RSA en php.

Descripción detallada

Clase para cifrar, descifrar, firmar, sellar y validar mensajes con RSA. Cifrar = codificar con una llave pública, sólo el emisor de la llave podrá conocer su contenido. Descifrar = decodificar con la llave privada del codificador mensajes que fueron enviados confidencialmente a éste codificador. Firmar = generar un hash de un mensaje y cifrarlo con la llave privada, para certificar que el emisor del mensaje fue éste codificador. Sellar = codificar un mensaje con la llave privada para certificar que el emisor del mensaje fue éste codificador. Validar = Verificar que un mensaje sellado o firmado fue emitido por quien generó la llave pública asociada. METODOS function __construct($duracion = 300, $directorio = null , $nombre_base = "rsa_key", $tipo_huella = "sha512") function get_ll_publica() function get_ll_pub_old() function get_refrescar() function cargar_llaves_RSA() final protected function crear_llaves_RSA() function firmar($mensaje, $sellado = false, $usar_old = false) public static function validar_firma($mensaje, $firma, $ll_publica , $tamanho = 2048, $tipo_huella = "sha512") public static function cifrar($mensaje, $ll_publica, $tamanho = 2048) function descifrar($mensaje) function borrarLlavesViejas()

Definición en la línea 67 del archivo cifrado_RSA.php.

Documentación del constructor y destructor

skinait\Codificador\codificador::__construct ($duracion = 300, $directorio = null, $nombre_base = "rsa_key", $tipo_huella = "sha512", $depurando = false)

Constructora, inicializa los campos del objeto a sus valores por defecto.

ENTRADAS:

Parámetros

$duracion	Tiempo máximo de duración de las llaves por defecto 5 minutos (300 segs = 5 min)
$directorio	Ruta a directorio donde se almacenarrán las llaves. El directorio debería tener permisos de lecto-escritura para php.
$nombre_base	Nombre de los archivos con las llaves, uno con extensión .pem y el otro, para la llave privada, con extensión .crt
$tipo_huella	Algoritmo hash que se utilizará para las huellas digitales debería estar entre los valores descritos en hash_algos().

Definición en la línea 91 del archivo cifrado_RSA.php.

Documentación de las funciones miembro

skinait\Codificador\codificador::borrarLlavesViejas ()

Revisa la antiguedad de las llaves almacenadas en el directorio con las llaves pública y privada, si son más viejas que lo indicado en "duración" se eliminan los archivos.

Deja una copia de la anterior pareja de llaves para brindar continuidad durante un intercambio de datos recurrente y prolongado (una sesión que esté intercambiando mensajes con frecuencia menor a la duracion)

Ésta clase se asegura de utilizar llaves recientes (de acuerdo con $this->duracion) Si la generación del mensaje de retorno se demora más que 2 veces la duración establecida en el objeto la conexión fallará. Es decir que hay que asegurarse que si la generación de un mensaje es demorado (i.e. la búsqueda en una base de datos gigante), debe tomarse ésto en cuenta para el establecimiento de la propiedad duración del objeto.

Definición en la línea 550 del archivo cifrado_RSA.php.

skinait\Codificador\codificador::cargar_llaves_RSA ()

Si las llaves están almacenadas en los archivos correspondientes las carga desde los archivos.

Si no existen los archivos de las llaves, genera un nuevo par de llaves. Si existe el par de llaves antiguas, las carga también, de lo contrario las deja en null.

Definición en la línea 184 del archivo cifrado_RSA.php.

static skinait\Codificador\codificador::cifrar ($mensaje, $ll_publica, $tamanho = 2048)[static]

Se utiliza cuando se quiere enviar un mensaje confidencial a un receptor.

Dado que el receptor es el único que conoce la llave privada correspondiente para descifrarlo, nadie más podrá conocer su contenido.

ENTRADAS:

Parámetros

$mensaje	El mensaje a cifrar.
$ll_publica	La llave pública del receptor a quien se va a enviar el mensaje.
$tamanho	El tamaño de las llaves.

Definición en la línea 430 del archivo cifrado_RSA.php.

skinait\Codificador\codificador::crear_llaves_RSA ()[final], [protected]

Generacion de llaves RSA en php.

Si existen llaves previas almacenadas, las pasa a los archivos con sufijo _old para guardar registro de su existencia previa.

SALIDA: Se generarán dos archivos, uno con la llave privada con extensión .crt, el otro con llave pública con extensión .pub; la función retorna tanto la llave pública como la privada en un arreglo.

PRE: $this->directorio es un directorio con permisos de escritura para Php Los archivos .pub y .crt almacenados allí son parejas de llaves pública y privada válidos.

Definición en la línea 238 del archivo cifrado_RSA.php.

skinait\Codificador\codificador::descifrar ($mensaje)

Se utiliza cuando llega un mensaje de un emisor externo que tomó la llave pública de éste objeto para cifrar un mensaje.

Se considera que éste mensaje es confidencial dado que éste objeto es el único que conoce la llave privada.

Si no es posible descifrar el mensaje con la llave pública se intenta con la llave pública antigua. El retorno indicará si debe enviarse la actual llave pública al receptor.

Dado que el algoritmo refresca las llaves RSA, sólamente se podrán descifrar mensajes recientes (de acuerdo con $this->duración).

ENTRADAS:

Parámetros

$mensaje	Mensaje a descifrar, lo cifró el emisor utilizando la llave pública que previamente le compartimos. SALIDA: Arreglo con El mensaje descifrado (o null si no se pudo descifrar). Indicador de si hay que indicar al emisor que actualice la llave pública.

Definición en la línea 470 del archivo cifrado_RSA.php.

skinait\Codificador\codificador::firmar ($mensaje, $sellado = false, $usar_old = false)

Se utiliza para cifrar con la llave privada un mensaje que cualquier persona podrá descifrar con la llave pública que se compartirá.

Equivale a un mensaje firmado, puesto que cualquiera puede conocer su contenido, pero se garantiza que el único que pudo firmarlo fue éste objeto dado que nadie más conoce la llave privada.

El mensaje no es confidencial dado que cualquiera puede conocer la llave pública para descifrarlo.

ENTRADAS:

Parámetros

$mensaje	El mensaje que se va a firmar.
$sellado	true indica que se cifrará todo el mensaje con la llave privada para su firma. false indica que se sacará la huella digital y se firmará únicamente la huella.
$usar_old	true indica que debe utilizarse la pareja de llaves antiguas y no las más recientes. SALIDA: Arreglo con la firma resultante o el mensaje sellado resultante. la llave pública para verificar la firma. El tipo de hash utilizado para la huella digital, si se utilizó.

Definición en la línea 306 del archivo cifrado_RSA.php.

skinait\Codificador\codificador::get_duracion ()

Entregar la duración establecida de las llaves en segundos.

Definición en la línea 143 del archivo cifrado_RSA.php.

skinait\Codificador\codificador::get_ll_pub_old ()

Entregar la llave pública que pronto expirará.

Definición en la línea 163 del archivo cifrado_RSA.php.

skinait\Codificador\codificador::get_ll_publica ()

Entregar la llave pública vigente.

Definición en la línea 153 del archivo cifrado_RSA.php.

skinait\Codificador\codificador::get_refrescar ()

Informa el valor del centinela que sugiere refrescar las llaves dado que se detectó que se está utilizando una próxima a expirar.

Definición en la línea 173 del archivo cifrado_RSA.php.

static skinait\Codificador\codificador::validar_firma ($mensaje, $firma, $ll_publica, $tamanho = 2048, $tipo_huella = "sha512")[static]

Se utiliza cuando llega un mensaje de un emisor externo del cual se conoce la llave pública para descifrarlo.

Dado que el emisor es el único que pudo cifrar el mensaje con la llave privada correspondiente con la pública, se considera firmado o sellado.

Dado que cualquiera puede conocer la llave pública, se considera que el mensaje no está oculto y no es confidencial.

ENTRADAS:

Parámetros

$mensaje	El mensaje al que se le quiere validar la firma.
$firma	La firma correspondiente al mensaje cuando se haya utilizado un algoritmo de hash, de lo contrario tendrá el mensaje completo que fue cifrado con la llave pública.
$ll_publica	La llave pública con la que se puede descifrar el mensaje.
$tamanho	El tamaño de las llaves
$tipo_huella	El algoritmo hash utilizado para generar la huella digital del mensaje. SALIDA: arreglo con: El mensaje descifrado. booleano que indica si la firma se validó exitosamente o no.

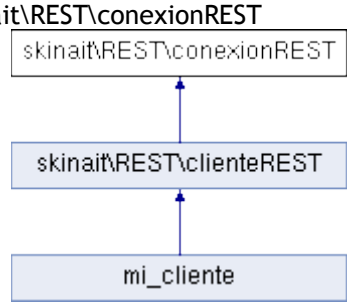
Definición en la línea 369 del archivo cifrado_RSA.php.

La documentación para esta clase fue generada a partir del siguiente fichero:

cifrado_RSA.php

Referencia de la Clase skinait\REST\conexionREST

Diagrama de herencias de skinait\REST\conexionREST



Métodos públicos estáticos

static solicitar_url ($url)
Descarga el contenido de un URL.

Descripción detallada

Clase para la descarga del contenido de un url. Generalmente será símplemente un file_get_contents, pero es posible que algún sitio requiera algo más elaborado mediante el uso de CURL.

OJO: Revisar la documentación de: https://reqbin.com/req/php/v0crmky0/rest-api-post-example

Definición en la línea 49 del archivo REST.php.

Documentación de las funciones miembro

static skinait\REST\conexionREST::solicitar_url ($url)[static]

Descarga el contenido de un URL.

ENTRADAS:

Parámetros

$url	La ruta al recurso que se va a descargar. SALIDAS: string: contenido de la página

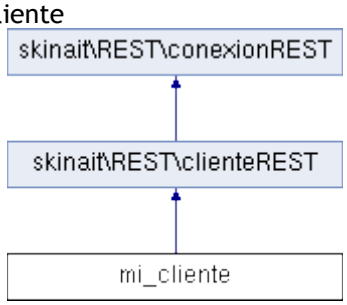
Definición en la línea 51 del archivo REST.php.

La documentación para esta clase fue generada a partir del siguiente fichero:

REST.php

Referencia de la Clase mi_cliente

Diagrama de herencias de mi_cliente



Métodos públicos

guardar_sesion_servicio ($url, $sesion, $pkey)

Almacena en forma persistente (archivo o base de datos) el id de sesión indicado asociado con el servidor con una llave pública y su fecha de creación.

cargar_sesion_servicio ($url)

Esta función debe recuperar el último id de sesión establecido con el servidor si se tiene disponible.

Otros miembros heredados

Descripción detallada

Definición en la línea 46 del archivo mi_cliente.php.

Documentación de las funciones miembro

mi_cliente::cargar_sesion_servicio ($url)

Esta función debe recuperar el último id de sesión establecido con el servidor si se tiene disponible.

Debe ser implementada de acuerdo con la lógica particular de cada sistema de información.

NOTA PARA ÉSTE EJEMPLO En éste ejemplo se almacena en un archivo en formato json con hashtables. Pero es ideal que se almacene en una base de datos como postgresql o mariadb.

ENTRADAS

Parámetros

$url	El localizador del servicio con el que se está estableciendo conexión. SALIDA Arreglo con: El último id de sesión o null si no se tiene La llave pública asociada con el id de sesión.

Reimplementado de skinait\REST\clienteREST (p.9).

Definición en la línea 123 del archivo mi_cliente.php.

mi_cliente::guardar_sesion_servicio ($url, $sesion, $pkey)

Almacena en forma persistente (archivo o base de datos) el id de sesión indicado asociado con el servidor con una llave pública y su fecha de creación.

Ésta función existe para minimizar el tráfico asociado con el envío de llaves públicas entre cliente y servidor.

NOTA PARA ÉSTE EJEMPLO Debe ser implementada de acuerdo con la lógica particular de cada sistema de información. En éste ejemplo se almacena en un archivo en formato json con hashtables. Pero es ideal que se almacene en una base de datos como postgresql o mariadb.

ENTRADAS

Parámetros

$url	El localizador del servicio con el que se estableció conexión.
$sesion	El identificador de sesión que se va a asociar.
$pkey	La llave pública asociada con el URL.

Reimplementado de skinait\REST\clienteREST (p.9).

Definición en la línea 55 del archivo mi_cliente.php.

La documentación para esta clase fue generada a partir del siguiente fichero:

mi_cliente.php

Referencia de la Clase mi_servicio

Diagrama de herencias de mi_servicio

Métodos públicos

M_REST_existen_procesos_asociados ($request)
Verifica en la base de datos si existen procesos de cobro activos asociados con un deudor.

cargar_hash_de_clave ($login)
guardar_llave_publica_sesion ($sesion, $pkey)
Almacena en forma persistente (archivo o base de datos) un código de sesión establecido asociado con una llave pública y su fecha de creación.

cargar_llave_publica_sesion ($sesion)
Verifica si se tiene almacenada una llave pública del cliente asociada con el identificador de sesión, almacenada en un momento más reciente que $this->duracion.

Otros miembros heredados

Descripción detallada

Definición en la línea 61 del archivo mi_servicio.php.

Documentación de las funciones miembro

mi_servicio::cargar_hash_de_clave ($login)

Reimplementado de skinait\REST\servidorREST (p.24).

Definición en la línea 151 del archivo mi_servicio.php.

mi_servicio::cargar_llave_publica_sesion ($sesion)

Verifica si se tiene almacenada una llave pública del cliente asociada con el identificador de sesión, almacenada en un momento más reciente que $this->duracion.

Ésta llave pública se utiliza para cifrar la información que se va a devolver al cliente.

NOTA PARA ÉSTE EJEMPLO En éste ejemplo se almacena en un archivo en formato json con hashtables. Pero es ideal que se almacene en una base de datos como postgresql o mariadb.

ENTRADAS

Parámetros

$sesion	un código alfanumérico que identifica la sesión. SALIDA string : La llave pública del cliente asociada con el código de sesión.

Reimplementado de skinait\REST\servidorREST (p.24).

Definición en la línea 245 del archivo mi_servicio.php.

mi_servicio::guardar_llave_publica_sesion ($sesion, $pkey)

Almacena en forma persistente (archivo o base de datos) un código de sesión establecido asociado con una llave pública y su fecha de creación.

Llaves más antiguas que la duración establecida en $this->duracion deberán ser eliminadas del llavero.

NOTA PARA ÉSTE EJEMPLO En éste ejemplo se almacena en un archivo en formato json con hashtables. Pero es ideal que se almacene en una base de datos como postgresql o mariadb.

ENTRADAS

Parámetros

$sesion	un código alfanumérico de longitud 13 con el que se identifica la sesión.
$pkey	la llave pública asociada con el código de sesión. SALIDA bool: true -> almacenada exitosamente

Reimplementado de skinait\REST\servidorREST (p.25).

Definición en la línea 181 del archivo mi_servicio.php.

mi_servicio::M_REST_existen_procesos_asociados ($request)

Verifica en la base de datos si existen procesos de cobro activos asociados con un deudor.

NOTA PARA ÉSTE EJEMPLO Todos los metodos REST del servicio deben seguir el siguiente formato: El nombre del método debe tener el prefimo M_REST_ La función debe retornar un arreglo asociativo en el cual se deben incluir los campos id_error y error como se describen en SALIDAS.

ENTRADAS:

Parámetros

$request['id_deudor']	Documento de identidad del deudor [opcional].
$request['email_deudor']	Correo electrónico del deudor [opcional]. SALIDAS: Arreglo con los siguientes campos: $retorno['Qprocesos'] Cantidad de procesos de cobro asociados con la identidad. $retorno['id_error'] Código del error generado si lo hubiera. $retorno['error'] Descripción del error generado si lo hubiera.

Definición en la línea 94 del archivo mi_servicio.php.

La documentación para esta clase fue generada a partir del siguiente fichero:
mi_servicio.php

Referencia de la Clase skinait\REST\servidorREST

Diagrama de herencias de skinait\REST\servidorREST

Métodos públicos

__construct ($directorio, $sesion=null, $duracion=300, $depurando=false)
Constructora, inicializa los campos del objeto a sus valores por defecto.

establecer_lista_blanca ($lista_blanca)
Verifica que lista_blanca['metodo'] tenga únicamente nombres de métodos válidos.

atender ($request)
Atiende la petición de un cliente, no sin antes verificar que los datos que provienen del cliente sean seguros y estén cifrados.

asegurar_requerimiento ($arreglo)
DESCRIPCION: Verifica que los elementos del arreglo sean seguros.

get_duracion ()
Entregar la duración establecida de las llaves en segundos.

M_REST_saludo ($request)
Entrega la llave pública de cifrado al cliente.

M_REST_autenticar ($request)
Realiza el proceso de autenticación del cliente que intenta establecer conección con el servicio.

cargar_hash_de_clave ($login)
guardar_llave_publica_sesion ($sesion, $pkey)
Esta función debe entregar el SHA512 de la clave del usuario con $login para ser utilizada durante la autenticación del cliente.

cargar_llave_publica_sesion ($sesion)
Almacena en forma persistente (archivo o base de datos) un código de sesión establecido asociado con una llave pública.

__call ($nombre, $parametros)
Verifica si se tiene almacenada una llave pública del cliente asociada con el identificador de sesión.

Métodos públicos estáticos

static __callStatic ($nombre, $parametros)
DESCRIPCION: ENTRADAS: SALIDAS: PRE: POST:

Campos de datos

$directorio
Directorio con permisos de lecto-escritura donde se podrán almacenar las llaves de cifrado.

Descripción detallada

Clase para el servidor de integración de componentes via REST .

Definición en la línea 305 del archivo REST.php.

Documentación del constructor y destructor

skinait\REST\servidorREST::__construct ($directorio, $sesion = null, $duracion = 300, $depurando = false)

Constructora, inicializa los campos del objeto a sus valores por defecto.

ENTRADAS:

Parámetros

$directorio	Directorio con permisos de lecto-escritura donde se podrán almacenar las llaves de cifrado.
$sesion	Cadena alfanumérica de longitud 13 que identifica una sesión previamente establecida.
$duracion	La cantidad de segundos que pueden transcurrir sin que se deban refrescar las llaves de cifrado.

Definición en la línea 317 del archivo REST.php.

Documentación de las funciones miembro

skinait\REST\servidorREST::__call ($nombre, $parametros)

Verifica si se tiene almacenada una llave pública del cliente asociada con el identificador de sesión.

Ésta llave pública se utiliza para cifrar la información que se va a devolver al cliente.

ENTRADAS

Parámetros

$sesion	un código alfanumérico que identifica la sesión. SALIDA string : La llave pública del cliente asociada con el código de sesión. DESCRIPCION: ENTRADAS: SALIDAS: PRE: POST:

Definición en la línea 684 del archivo REST.php.

static skinait\REST\servidorREST::__callStatic ($nombre, $parametros)[static]

DESCRIPCION: ENTRADAS: SALIDAS: PRE: POST:

Definición en la línea 699 del archivo REST.php.

skinait\REST\servidorREST::asegurar_requerimiento ($arreglo)

DESCRIPCION: Verifica que los elementos del arreglo sean seguros.

Coteja contra la lista blanca de valores los campos libres. Todo dato que no sea libre lo descifra con la llave privada para garantizar que llegó de forma segura. SALIDAS El arreglo asegurado y, en caso que el cliente esté utilizando la antigua llave pública para cifrar sus datos, un arreglo $refrescar con la llave que hay que refrescar, firmada con la antigua llave privada.

Definición en la línea 480 del archivo REST.php.

skinait\REST\servidorREST::atender ($request)

Atiende la petición de un cliente, no sin antes verificar que los datos que provienen del cliente sean seguros y estén cifrados.

ENTRADAS:

Parámetros

$request	arreglo asociativo con la información de la requisicion. en particular deberá contar con el campo "metodo" que indicará el método que el usuario desea que se procese. SALIDAS: string: arreglo codificado en json con los datos de salida del método si el cliente se autenticó y registró su llave pública la salida estará cifrada sólo para el cliente.

Definición en la línea 417 del archivo REST.php.

skinait\REST\servidorREST::cargar_hash_de_clave ($login)[abstract]

Reimplementado en mi_servicio (p.19).

skinait\REST\servidorREST::cargar_llave_publica_sesion ($sesion)[abstract]

Almacena en forma persistente (archivo o base de datos) un código de sesión establecido asociado con una llave pública.

ENTRADAS

Parámetros

$sesion	un código alfanumérico de longitud 13 con el que se identifica la sesión.
$pkey	la llave pública asociada con el código de sesión. SALIDA bool: true -> almacenada exitosamente

Reimplementado en mi_servicio (p.19).

skinait\REST\servidorREST::establecer_lista_blanca ($lista_blanca)

Verifica que lista_blanca['metodo'] tenga únicamente nombres de métodos válidos.

ENTRADAS:

Parámetros

$lista_blanca	Arreglo asociativo indicando los valores válidos para cada una de las variables que vangan en el $_REQUEST. En particular ['metodo'] que trae la llamada a los diferentes servicios provistos por el servicio. SALIDAS: En $this->lista_blanca quedarán los métodos válidos.

Definición en la línea 372 del archivo REST.php.

skinait\REST\servidorREST::get_duracion ()

Entregar la duración establecida de las llaves en segundos.

Definición en la línea 536 del archivo REST.php.

skinait\REST\servidorREST::guardar_llave_publica_sesion ($sesion, $pkey)[abstract]

Esta función debe entregar el SHA512 de la clave del usuario con $login para ser utilizada durante la autenticación del cliente.

Debe ser implementada de acuerdo con la lógica particular de cada sistema de información.

ENTRADAS

Parámetros

$login	el identificador del usuario. SALIDA El hash SHA512 que se tiene almacenado de la clave del usuario, null si no se tiene.

Reimplementado en mi_servicio (p.20).

skinait\REST\servidorREST::M_REST_autenticar ($request)

Realiza el proceso de autenticación del cliente que intenta establecer conección con el servicio.

Requiere que se implementen los métodos abstracto llamados cargar_hash_de_clave, guardar_llave_publica_sesion y cargar_llave_publica_sesion.

El proceso revisa que la comunicación no exceda el tiempo de timeout, verifica la clave del usuario y, si todo está correcto, genera un identificador único de sesión y guarda la llave pública del cliente para futuras comunicaciones.

ENTRADAS:

Parámetros

$request	arreglo asociativo con la información de la requisicion. Particularmente debe tener los siguientes campos:
login	el login de identificación del usuario
hclave	el hash SHA512 de la clave de acceso
fecha	la fecha en que el cliente envía los datos de conexión
ll_cliente	la llave pública de cifrado del cliente SALIDAS: Arreglo asociativo con los campos: autenticado booleano que indica el estado de autenticación. mensaje descripción de la finalización del proceso [sesion] cadena alfanumérica con el código único de sesión para ser utilizado en futuras sesiones.

Definición en la línea 574 del archivo REST.php.

skinait\REST\servidorREST::M_REST_saludo ($request)

Entrega la llave pública de cifrado al cliente.

ENTRADAS:

Parámetros

$request	arreglo asociativo con la información de la requisicion. Este método no la utiliza, llega únicamente por estandarización. SALIDAS: La llave pública del servicio.

Definición en la línea 546 del archivo REST.php.

Documentación de los campos

skinait\REST\servidorREST::$directorio

Directorio con permisos de lecto-escritura donde se podrán almacenar las llaves de cifrado.

Definición en la línea 307 del archivo REST.php.

La documentación para esta clase fue generada a partir del siguiente fichero:

REST.php

Documentación de archivos

Referencia del Archivo cifrado_RSA.php

Estructuras de datos

class skinait\Codificador\codificador

Namespaces

skinait\Codificador

Referencia del Archivo cliente.php

Variables

$depurando = true
$url
$login = "skinait"
$clave = hash('sha512', '3uv5nqc!"')
$fecha = date("Y-m-d H:i:s")
$cliente = new mi_cliente($url, $login, $clave, $fecha, __DIR__, $depurando)
$parametros ['id_deudor'] = "123456789"
$respuesta = $cliente->solicitar("existen_procesos_asociados", $parametros)
$parametros ['email_deudor'] = "soporte@skinait.com"
$parametros ['email_deudor'] = "soporteskinait.com"

Descripción detallada

Este software fué realizado por el Ing. Ricardo Naranjo Faccini, M.Sc. para Skina IT Solutions E.U. fábrica de software colombiana radicada en Bogotá.

Skina IT Solutions E.U. https://www.skinait.com soporte@skinait.com

Copyright 2021

Este archivo es parte de la librería SimpleSecREST. Contiene el ejemplo de uso de la librería desde el punto de

vista del cliente REST.

This file is part of Simple-SecREST.

Simple-SecREST is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Simple-SecREST is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with Simple-SecREST. If not, see https://www.gnu.org/licenses/.

Documentación de las variables

$clave = hash('sha512', '3uv5nqc!"')

Definición en la línea 55 del archivo cliente.php.

$cliente = new mi_cliente($url, $login, $clave, $fecha, __DIR__, $depurando)

Definición en la línea 58 del archivo cliente.php.

$depurando = true

Definición en la línea 43 del archivo cliente.php.

$fecha = date("Y-m-d H:i:s")

Definición en la línea 56 del archivo cliente.php.

$login = "skinait"

Definición en la línea 54 del archivo cliente.php.

$parametros['email_deudor'] = "soporte@skinait.com"

Definición en la línea 65 del archivo cliente.php.

$parametros['email_deudor'] = "soporteskinait.com"

Definición en la línea 70 del archivo cliente.php.

$parametros['id_deudor'] = "123456789"

Definición en la línea 60 del archivo cliente.php.

print $respuesta = $cliente->solicitar("existen_procesos_asociados", $parametros)

Definición en la línea 61 del archivo cliente.php.

$url

```
Valor inicial:= "http://"
       . $_SERVER['SERVER_NAME']
       . preg_split("#/cliente#", $_SERVER['REQUEST_URI'])[0]
       . "/servidor/servicio.php"
```

Definición en la línea 49 del archivo cliente.php.

Referencia del Archivo herramientas.php

Funciones

Mostrar ($dato, $texto="")
periodo2unidades ($intervalo, $tipo)
Cambia el formato de un intervalo a las unidades expresadas en tipo.

traza_de_funciones ()

Descripción detallada

Este software fué realizado por el Ing. Ricardo Naranjo Faccini, M.Sc. para Skina IT Solutions E.U. fábrica de software colombiana radicada en Bogotá.

Skina IT Solutions E.U. https://www.skinait.com soporte@skinait.com

Copyright 2021

Este archivo es parte de la librería SimpleSecREST. Contiene funciones utilitarias varias que se requieren para el buen funcionamiento del sistema pero no están necesariamente involucradas con la transmisión

REST segura.

This file is part of Simple-SecREST.

Simple-SecREST is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Simple-SecREST is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with Simple-SecREST. If not, see https://www.gnu.org/licenses/.

Documentación de las funciones

Mostrar ($dato, $texto = "")

Definición en la línea 37 del archivo herramientas.php.

periodo2unidades ($intervalo, $tipo)

Cambia el formato de un intervalo a las unidades expresadas en tipo.

ENTRADAS:

Parámetros

$intervalo	El intervalo a formatear
$tipo	Podrá ser years, months, days, hours, minutes, seconds o miliseconds

	SALIDAS: Intervalo reformateado

Definición en la línea 45 del archivo herramientas.php.

traza_de_funciones ()

Definición en la línea 128 del archivo herramientas.php.

Referencia del Archivo mi_cliente.php

Estructuras de datos

class mi_cliente

Referencia del Archivo mi_servicio.php

Estructuras de datos

class mi_servicio

Referencia del Archivo REST.php

Estructuras de datos

class skinait\REST\conexionREST
class skinait\REST\clienteREST
class skinait\REST\servidorREST

Namespaces

skinait\REST

Referencia del Archivo servicio.php

Variables

$depurando = false
$servidor = new mi_servicio(__DIR__, null, 300, $depurando)
$lista_blanca ['metodo']
$respuesta = $servidor->atender($_REQUEST)

Descripción detallada

Este software fué realizado por el Ing. Ricardo Naranjo Faccini, M.Sc. para Skina IT Solutions E.U. fábrica de software colombiana radicada en Bogotá.

Skina IT Solutions E.U. https://www.skinait.com soporte@skinait.com

Copyright 2021

Este archivo es parte de la librería SimpleSecREST. Contiene el ejemplo de uso de la librería desde el punto de

vista del servidor REST.

This file is part of Simple-SecREST.

Simple-SecREST is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Simple-SecREST is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with Simple-SecREST. If not, see https://www.gnu.org/licenses/.

Documentación de las variables

$depurando = false

Definición en la línea 39 del archivo servicio.php.

$lista_blanca['metodo']

```
Valor inicial:= array( "existen_procesos_asociados"


                                                    )
```

Definición en la línea 46 del archivo servicio.php.

print $respuesta = $servidor->atender($_REQUEST)

Definición en la línea 54 del archivo servicio.php.

$servidor = new mi_servicio(__DIR__, null, 300, $depurando)

Definición en la línea 44 del archivo servicio.php.

Índice

actualizar_ll_srvskinaitRESTclienteREST..... 8
actualizar_llave_publica_de_clienteskinaitRESTclienteREST..... 8
asegurar_requerimientoskinaitRESTservidorREST..... 24
atenderskinaitRESTservidorREST..... 24
borrarLlavesViejasskinaitCodificadorcodificador..... 12
cargar_hash_de_clavemi_servicio..... 19
cargar_hash_de_claveskinaitRESTservidorREST..... 24
cargar_llave_publica_sesionmi_servicio..... 19
cargar_llave_publica_sesionskinaitRESTservidorREST..... 24
cargar_llaves_RSAskinaitCodificadorcodificador..... 13
cargar_sesion_serviciomi_cliente..... 17
cargar_sesion_servicioskinaitRESTclienteREST..... 9
cifrado_RSA.php..... 27
cifrarskinaitCodificadorcodificador..... 13
cliente.php..... 28
cliente.php$clave..... 28
cliente.php$cliente..... 28
cliente.php$depurando..... 29
cliente.php$fecha..... 29
cliente.php$login..... 29
cliente.php$parametros..... 29
..... 29
..... 29
cliente.php$respuesta..... 29
cliente.php$url..... 29
crear_llaves_RSAskinaitCodificadorcodificador..... 13
descifrarskinaitCodificadorcodificador..... 13
establecer_lista_blancaskinaitRESTservidorREST..... 24
firmarskinaitCodificadorcodificador..... 14
get_directorioskinaitRESTclienteREST..... 9
get_duracionskinaitCodificadorcodificador..... 14
get_duracionskinaitRESTclienteREST..... 9
get_duracionskinaitRESTservidorREST..... 25
get_ll_pub_oldskinaitCodificadorcodificador..... 14
get_ll_publicaskinaitCodificadorcodificador..... 14
get_refrescarskinaitCodificadorcodificador.....

15
guardar_llave_publica_sesionmi_servicio
20
guardar_llave_publica_sesionskinaitRESTservidorREST
25
guardar_sesion_serviciomi_cliente
18
guardar_sesion_servicioskinaitRESTclienteREST
9
herramientas.php
30
herramientas.phpMostrar
30
herramientas.phpperiodo2unidades
30
herramientas.phptraza_de_funciones
31
M_REST_autenticarskinaitRESTservidorREST
25
M_REST_existen_procesos_asociadosmi_servicio
20
M_REST_saludoskinaitRESTservidorREST
25
mi_cliente
17
mi_cliente.php
32
mi_clientecargar_sesion_servicio
17
mi_clienteguardar_sesion_servicio
18
mi_servicio
19
mi_servicio.php
33
mi_serviciocargar_hash_de_clave
19
mi_serviciocargar_llave_publica_sesion
19
mi_servicioguardar_llave_publica_sesion
20
mi_servicioM_REST_existen_procesos_asociados
20
Mostrarherramientas.php
30
periodo2unidadesherramientas.php
30
REST.php
34
servicio.php
35
servicio.php$depurando
35
servicio.php$lista_blanca
35
servicio.php$respuesta
35
servicio.php$servidor
36
skinaitCodificador
5
skinaitCodificadorcodificador
11
skinaitCodificadorcodificador__construct
12
skinaitCodificadorcodificadorborrarLlavesViejas
12
skinaitCodificadorcodificadorcargar_llaves_RSA
13
skinaitCodificadorcodificadorcifrar
13
skinaitCodificadorcodificadorcrear_llaves_RSA
13
skinaitCodificadorcodificadordescifrar
13

skinaitCodificadorcodificadorfirmar.....14
skinaitCodificadorcodificadorget_duracion.....14
skinaitCodificadorcodificadorget_ll_pub_old.....14
skinaitCodificadorcodificadorget_ll_publica.....14
skinaitCodificadorcodificadorget_refrescar.....15
skinaitCodificadorcodificadorvalidar_firma.....15
skinaitREST.....6
skinaitRESTclienteREST.....7
skinaitRESTclienteREST__construct.....8
skinaitRESTclienteREST$fecha.....10
skinaitRESTclienteREST$hclave.....10
skinaitRESTclienteREST$login.....10
skinaitRESTclienteREST$url.....10
skinaitRESTclienteRESTactualizar_ll_srv.....8
skinaitRESTclienteRESTactualizar_llave_publica_de_cliente.....8
skinaitRESTclienteRESTcargar_sesion_servicio.....9
skinaitRESTclienteRESTget_directorio.....9
skinaitRESTclienteRESTget_duracion.....9
skinaitRESTclienteRESTguardar_sesion_servicio.....9
skinaitRESTclienteRESTsolicitar.....9
skinaitRESTconexionREST.....16
skinaitRESTconexionRESTsolicitar_url.....16
skinaitRESTservidorREST.....22
skinaitRESTservidorREST__call.....23
skinaitRESTservidorREST__callStatic.....24
skinaitRESTservidorREST__construct.....23
skinaitRESTservidorREST$directorio.....26
skinaitRESTservidorRESTasegurar_requerimiento.....24
skinaitRESTservidorRESTatender.....24
skinaitRESTservidorRESTcargar_hash_de_clave.....24
skinaitRESTservidorRESTcargar_llave_publica_sesion.....24
skinaitRESTservidorRESTestablecer_lista_blanca.....24
skinaitRESTservidorRESTget_duracion.....25
skinaitRESTservidorRESTguardar_llave_publica_sesion.....25
skinaitRESTservidorRESTM_REST_autenticar.....25
skinaitRESTservidorRESTM_REST_saludo.....25
solicitar_urlskinaitRESTconexionREST.....16
solicitarskinaitRESTclienteREST.....

.....9
traza_de_funcionesherramientas.php.....
.....31
validar_firmaskinaitCodificadorcodificador.....
.....15
__callskinaitRESTservidorREST.....
.....23
__callStaticskinaitRESTservidorREST.....
.....24
__constructskinaitCodificadorcodificador.....
.....12
__constructskinaitRESTclienteREST.....
.....8
__constructskinaitRESTservidorREST.....
.....23
$clavecliente.php.....
.....28
$clientecliente.php.....
.....28
$depurandocliente.php.....
.....29
$depurandoservicio.php.....
.....35
$directorioskinaitRESTservidorREST.....
.....26
$fechacliente.php.....
.....29
$fechaskinaitRESTclienteREST.....
.....10
$hclaveskinaitRESTclienteREST.....
.....10
$lista_blancaservicio.php.....
.....35
$logincliente.php.....
.....29
$loginskinaitRESTclienteREST.....
.....10
$parametroscliente.php.....
.....29
.....29
.....29
$respuestacliente.php.....
.....29
$respuestaservicio.php.....
.....35
$servidorservicio.php.....
.....36
$urlcliente.php.....
.....29
$urlskinaitRESTclienteREST.....
.....10