



Cómo usar la librería Simple-SecREST para implementar un servidor REST seguro.

Ing. Ricardo Naranjo Faccini, M.Sc.

2020-09-10

Agenda

- **Licencia**
- **Descargar la librería**
- **Organizar el árbol de directorios adecuado.**
- **Ajustar servicio.php**
- **Ajustar la clase mi_servicio.php**
 - Métodos abstractos obligatorios.
 - Métodos del servicio.

Licencia

- Se distribuye como software libre.
- Protegido con la licencia LGPL Lesser General Public Licence.
- Desarrollado por Ing. Ricardo Naranjo Faccini, M.Sc.
- Liberado por Skina IT Solutions.
 - Fábrica de software radicada en Colombia

Descargar la librería

- Repositorio en GitHub:
 - <https://github.com/gramo44/Simple-SecREST>

Organizar el árbol de directorios adecuado.

public_html (directorio publicado por el servidor web)

└─ **servidor**

└─ **lib**

└─ **cifrado_RSA.php**

└─ **herramientas.php**

└─ **mi_servicio.php**

└─ **Simple-SecREST.php**

└─ **servicio.php**

externo

└─ **prv** (directorio externo con permisos de escritura para el servidor web)

Ajustar servicio.php

```
$depurando = false; // Si se desea recibir mensajes de depuración.  
  
$servidor = new mi_servicio("Ruta/al/directorio/externo"  
    , null  
    , 300 // Indica la vigencia en segundos  
        // de las llaves  
    , $depurando);  
  
$lista_blanca['metodo'] = array( "servicio_1" // Nombre de los  
    , "servicio_2" // servicios que se  
    , "servicio_3" // brindan.  
    , "servicio_4"  
    , "servicio_5"  
    );  
  
$servidor->establecer_lista_blanca($lista_blanca);  
  
$respuesta = $servidor->atender($_REQUEST);
```

Ajustar la clase lib/mi_servicio.php

```
class mi_servicio extends SR_REST {
    function M_REST_servicio_1($request)
    {
        $retorno = array( 'xxxx' => 0
                        , 'yyyy' => 0
                        , 'id_error' => 0
                        , 'error' => ""
                        );

        return $retorno;
    }
    // ...
    function M_REST_servicio_n($request)
    {
        $retorno = array( 'xxxx' => 0
                        , 'yyyy' => 0
                        , 'id_error' => 0
                        , 'error' => ""
                        );

        return $retorno;
    }
    // MÉTODOS ABSTRACTOS OBLIGATORIOS
}
```

Ajustar la clase lib/mi_servicio.php

```
/*-----*/  
function cargar_hash_de_clave($login) : string  
/*****  
    @brief Esta función debe entregar el SHA512 de la clave del usuario  
    con $login para ser utilizada durante la autenticación del cliente.  
  
    ENTRADAS:  
    @param $login el login del usuario  
    SALIDAS:  
    El hash de la clave del usuario  
*****/
```


Ajustar la clase lib/mi_servicio.php

```
/*-----*/
function guardar_llave_publica_sesion($sesion, $pkey)
/*****
@brief 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.

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

Ajustar la clase lib/mi_servicio.php

```
/*-----*/  
function cargar_llave_publica_sesion($sesion)  
/*****  
@brief 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.
```

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

ENTRADAS

@param \$sesion un código alfanumérico que identifica la sesión.

SALIDA

string : La llave pública del cliente asociada con el código de sesión.
*****/

A background image of a green frog with large eyes perched on a thick, reddish-brown branch.

Muchas Gracias

¿Preguntas?

ventas@skinait.com

<http://www.skinait.com>