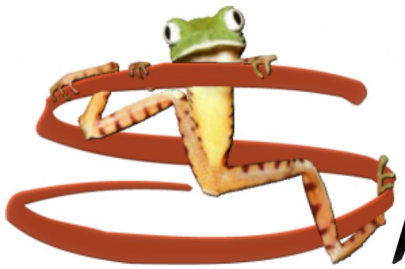


Skína IT Solutions

Cómo usar la librería Simple-SecREST para implementar una comunicación REST segura.

Ing. Ricardo Naranjo Faccini, M.Sc.

2020-09-10

https://www.skinait.com/simple-secrest-Escritos-60/https://www.skinait.com/simple-secrest-Escritos-60/

Agenda

- Licencia
- La librería
- Implementación del servidor
- Implementación del cliente

Licencia

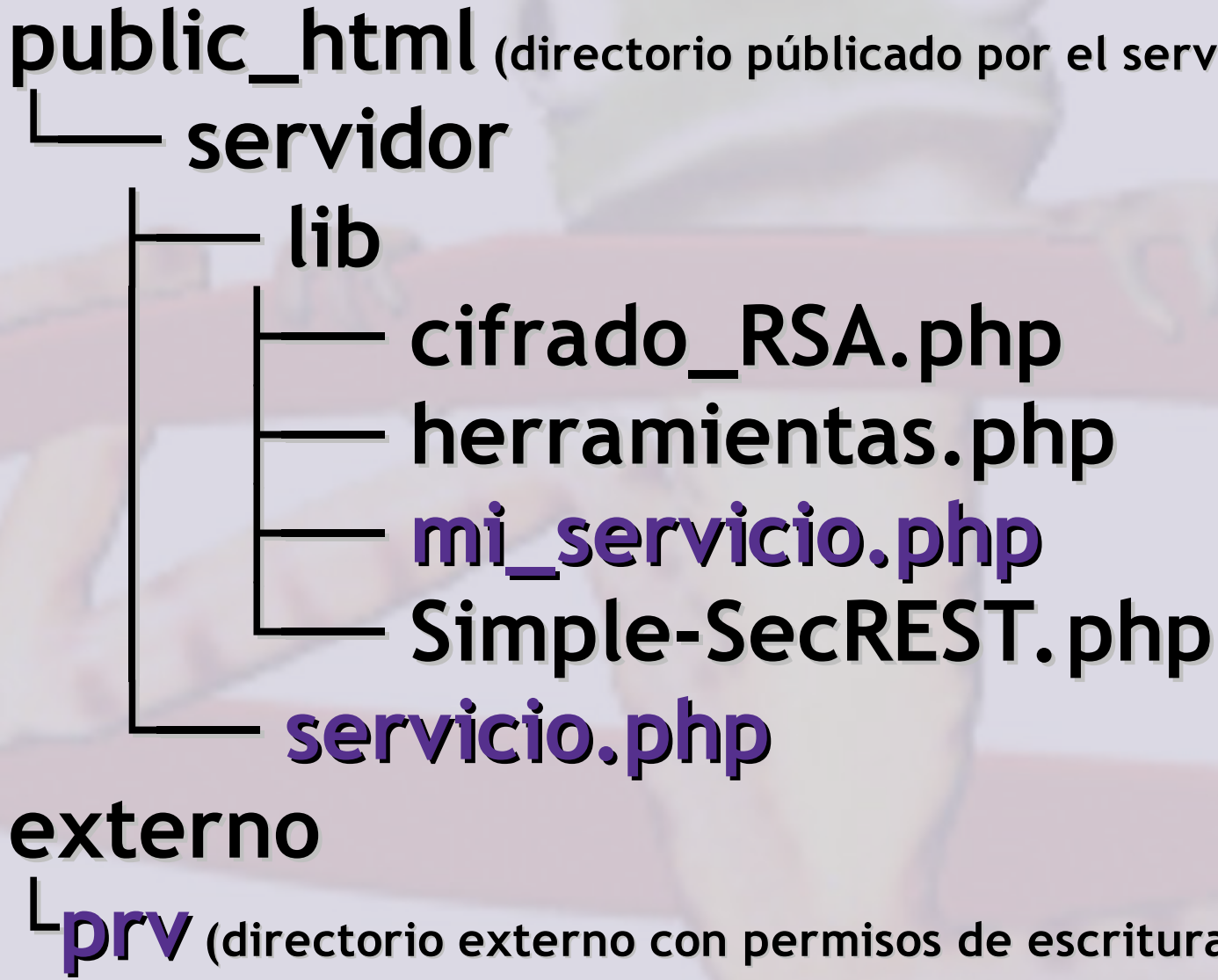
- Se distribuye como software libre.
- Protegido con la licencia LGPL Lesser General Public Licence.
- Desarrollado por Ing. Ricardo Naranjo Faccini, M.Sc.
- Liberado por Skina IT Solutions.
  - Fábrica de software radicada en Colombia

Sobre la librería

- Repositorio en GitHUB:
  - https://github.com/gramo44/Simple-SecREST
- Se encarga de mantener seguro el intercambio de datos en 4 momentos:
  - Saludo (Handshake):
    - Intercambio de llaves RSA de corta duración
  - Vencimiento de llaves:
    - Genera nueva pareja de claves y la firma con la que está por vencerse.
  - Envío de credenciales de acceso {login, clave}
    - Genera el hash de la clave y lo combina con la fecha y hora actual.
  - Intercambio de datos previa validación y cifrado RSA con las llaves de corta duración.

Implementación del servidor

Organizar el árbol de directorios adecuado.



Vincular servicio.php con el servicio que se brindará.

```
$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

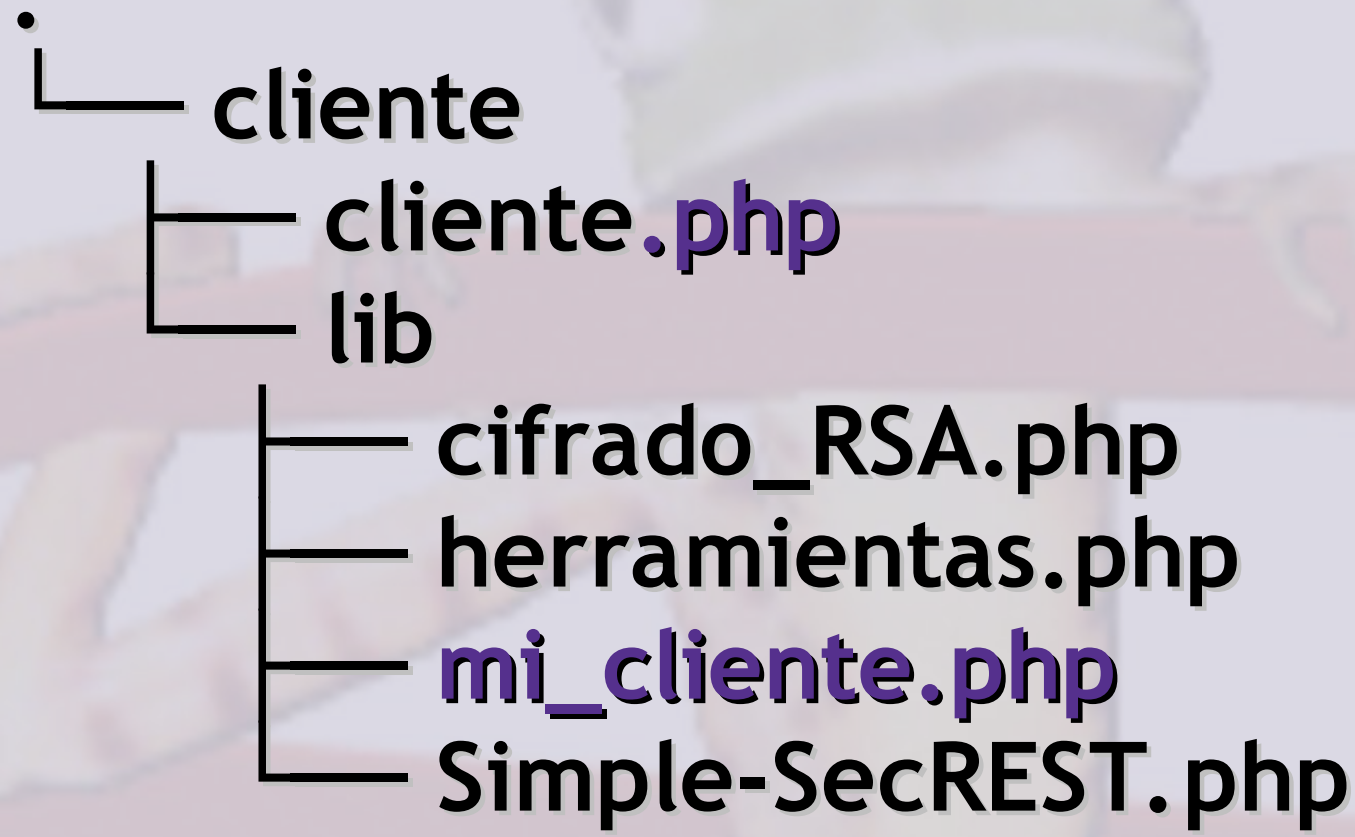
```
/*-----------------------------------------------------------------*/
function cargar_llave_publica_sesion($sesion)
/*******************************************************************
@brief Verifica si se tiene almacenada una llave pública del cliente
asociada con el identificador de sesión, almacenada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

Ésta 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.
*******************************************************************/
```

Implementación del cliente

Organizar el árbol de directorios adecuado.



Ajustar la clase lib/mi_cliente.php

```
class mi_cliente extends CL_REST {
    function guardar_sesion_servicio($url, $sesion, $pkey) {
    /*************************************************************
    @brief Almacena el id de sesión asociado con el servidor con una llave
    pública y su fecha de creación.
    ENTRADAS
    @param $url El localizador del servicio con el que se estableció conexión.
    @param $sesion El identificador de sesión que se va a asociar.
    @param $pkey La llave pública asociada con el URL.
    *************************************************************/
        Instrucciones;
    }
    public function cargar_sesion_servicio($url) {
    /*************************************************************
    @brief Recuperar el último id de sesión establecido con el servidor si
    se tiene disponible.
    ENTRADAS
    @param $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.
    *************************************************************/
        Instrucciones;
    }
}
```

Vincular cliente.php con la clase que consumirá el servicio.

```
require_once("lib/mi_cliente.php");
require_once("lib/herramientas.php");

$url = "http://url.del.servidor/ruta/al/servidor/servicio.php";
$login = "login";
$clave = hash('sha512', 'clave_de_ingreso');
$fecha = date("Y-m-d H:i:s");
$dir   = "/ruta/a/un/directorio/privado/con/permisos/de/escritura";
$depur = false; // o true si se quieren mensajes de depuración.
$cliente = new mi_cliente($url, $login, $clave, $fecha, $dir, $depur);

$parametros['xxxx'] = "123456789";
$respuesta = $cliente->solicitar("servicio_1", $parametros);
print var_export($respuesta, true);
```

Muchas Gracias

¿Preguntas?

ventas@skinait.com

http://www.skinait.com