Servicio de WebSSO

Manual de integración de aplicaciones PHP con openAM

Versión: 0100

[Versión del Producto]

HOJA DE CONTROL

Organismo	Consejería de Hacienda y Administración Pública		
Proyecto	Servicio de WebSSO		
Entregable	Manual de Integración		
Autor	Francisco Rodríguez Corredor		
Aprobado por		**Fecha Aprobación**	20/05/2015
		Nº Total de Páginas	13

REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
0100	Versión inicial	Francisco Rodríguez Corredor	20/05/2014

CONTROL DE DISTRIBUCIÓN

Nombre y Apellidos
<Nombre Apellido1 Apellido2>

ÍNDICE

1 OBJETIVO

El objetivo del presente documento es servir de guía para la integración de aplicaciones PHP que quieran hacer uso de las funcionalidades disponibles en el sistema de Single Sign On Web openAM, en adelante SSOWeb.

1.1 Glosario y definiciones

- SSO (Single Sign On). Es un mecanismo de autenticación mediante el cuál el usuario se autentica una vez propagando la identidad a las aplicaciones
- SAML. Es un estándar basado en XML para el intercambio de mensajes de autenticación y autorización entre dominios de seguridad.
- Federación de Identidades. La identidad federada es una de las soluciones para abordar la gestión de identidad en los sistemas de información. Su objetivo es obtener una gestión de usuarios eficiente, la sincronización de los datos identificativos, gestión de acceso, servicios de agrupación, servicios de directorio, auditoría e informes
- SP (Service Provider). Es el elemento que consume la información de autenticación y autorización en la relación federada. Se puede equiparar a la aplicación de negocio a integrar.
- IDP (IDentity Provider). Es el elemento que contiene la información de origen de la identidad en una relación federada.

2 PROCESO DE INTEGRACIÓN DE APLICACIONES PHP

2.1 Prerequisitos

Es necesario tener instalado el servidor Apache2, con el módulo de PHP5.

2.2 Procedimiento de configuración e integración

El proceso de integración de una aplicación PHP con el SSOWEB se compone de los siguientes pasos:

- Descarga e instalación del software SimpleSAMLPHP preconfigurado para el SSOWEB de la Junta de Andalucía. Este paquete se encuentra en la siguiente ruta DEFINIR
- Registro de nuestra aplicación en el entorno operativo del SSOWEB que se desea utilizar en la integración. Esta tarea le corresponde al administrador funcional del SSOWEB y se solicita mediante un ticket NAOS del tipo
- Modificación de la aplicación a integrar para invocar los métodos correspondientes proporcionados por SimpleSamlPHP

2.2.1 Configuración básica SimpleSAMLPHP

Una vez descargado el recurso SimpleSAMLPHP preconfigurado para el SSOWEB de la Junta de Andalucía, es necesario realizar los siguientes pasos:

1. Crear el directorio /opt/software/simplesamlphp en la máquina en la que se va a ubicar el servidor de aplicaciones sobre el que se ejecuta la aplicación que se desea integrar. De aquí en adelante a este directorio se le refenciará como $SIMPLESAMLPHP_DIR

    1. Si va a utilizar otra ruta del sistema de ficheros a tal efecto, deberá modificar las siguientes propiedades del fichero / $SIMPLESAMLPHP_DIR/config/config.php:

        - certdir
        - loggingdir
        - datadir

2. Otorgar el directorio anteriormente creado permisos de lectura, escritura y ejecución para el usuario que ejecuta el servicio correspondiente al servidor Apache

3. Descomprimir el paquete SimpleSAMLPHP_JDA.tar.gz en la carpeta anteriormente creada

4. Abrir una consola del sistema y lanzar el siguiente comando:

   “tr -c -d '0123456789abcdefghijklmnopqrstuvwxyz' </dev/urandom |dd bs=32 count=1 2>/dev/null;echo”

   Recuperar la salida del comando y cambiar el valor de la propiedad “secretsalt” del fichero /$SIMPLESAMLPHP_DIR/config/config.php por el valor obtenido.

5. Editar también en el fichero /$SIMPLESAMLPHP_DIR/config/config.php la propiedad 'baseurlpath' estableciendo para ella una url de la forma: 'https://<nombre_del_servidor>/simplesaml/'

6. Editar el fichero /$APACHE_DIR/apache/conf/httpd.conf y añadir el siguiente bloque de texto al final del fichero:

```
<IfModule alias_module>
      Alias /simplesaml/ "/opt/software/simplesamlphp/www/"
      <Directory "/opt/software/simplesamlphp/www">
         AllowOverride None
         Options None
         Order allow,deny
         Allow from all
      </Directory>
</IfModule>
```

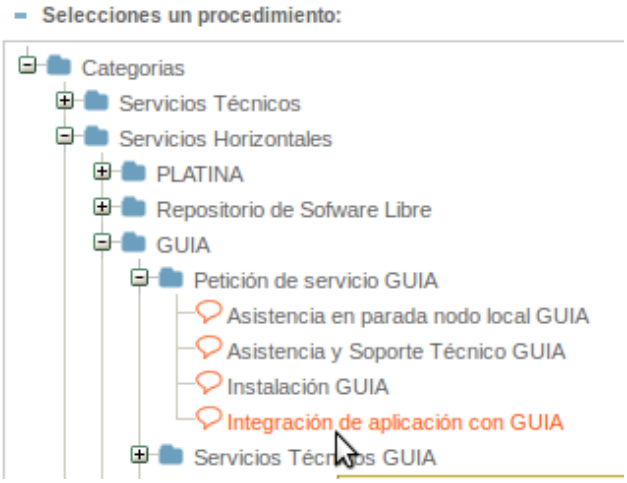
   Si ha desplegado SimpleSamlPHP en otro directorio diferente deberá modificar el texto indicado en el punto anterior para adaptarlo con los directorios correctos
7. Parar y arrancar Apache para que los cambios tomen efecto.
8. Comprobar que el punto de montaje de la aplicación web es correcto:
   1. Escribir https://<nombredelservidor>/simplesaml/
   2. Pulsar en “Entrar como administrador”. Cuando solicite una contraseña introducir: 123456
      1. Si se desea cambiar esta contraseña se puede editar el archivo /$SIMPLESAMLPHP_DIR/config/config.php y cambiar la propiedad “auth.adminpassword” por la contraseña deseada.
9. Pulsar en la pestaña “Configuración” y en el apartado “Verificación de su instalación de PHP” verificar que todo lo marcado como “Necesario” supera la verificación. Lo marcado como “Necesario para LDAP” no es obligatorio.
10. En el apartado “Configuración” también, pulsar sobre el enlace “Sanity

check of your simpleSAMLphp setup" y comprobar que se superan todas las comprobaciones.

11. El SimpleSamlPHP que se distribuye con esta guía está preconfigurado con los metadatos necesarios para los entornos de integración, preproducción y producción, por lo tanto Vd sólo tendrá que seleccionar en su integración contra qué entorno del SSOWEB se quiere integrar. Para ello establezca en el fichero /$SIMPLESAMLPHP_DIR/config/authsources.php los parámetros indicados en la siguiente tabla a su valor correspondiente para la entrada 'default-sp' que comienza en la línea 13 del fichero:

	entityID	***idp***
Integración	https://ssoweb.int.i-administracion.junta-andalucia.es:443/opensso	https://ssoweb.int.i-administracion.junta-andalucia.es:443/opensso
Preproducción	https://ssoweb.pre.juntadeandalucia.es/opensso	https://ssoweb.pre.juntadeandalucia.es/opensso
Producción	https://ssoweb.juntadeandalucia.es/opensso	https://ssoweb.juntadeandalucia.es/opensso

12. Una vez configurado el entorno destino del SSOWEB a utilizar, acceda al Portal de Usuario de NAOS y solicite mediante un ticket del tipo "Integración de Aplicación con GUIA". En el ticket creado indique la siguiente información:
    1. Nombre de la aplicación a integrar
    2. Consejería u organismo al que pertenece la aplicación a integrar
    3. Datos de contacto del responsable del servicio asociado a la aplicación a integrar: nombre, apellidos, teléfono y correo electrónico
    4. Url de metadatos SimpleSamlPHP de la aplicación integrar. Será del tipo https://<nombre_del_servidor>/simplesaml/module.php/saml/sp/metadata.php/default-sp



Si no puede ver este procedimiento en NAOS, solicite permisos para ello utilizando un ticket del tipo “Incidencia General”.

13. Una vez dado de alta su sistema en el SSOWEB correspondiente ya sólo falta que modifique el código de su aplicación para que invoque correctamente los métodos que SimpleSamlPHP le ofrece para la integración utilizando el protocolo SAML2.0. En el siguiente apartado se muestra una integración de ejemplo que debe servirle como referencia de las modificaciones a realizar.

3 Ejemplo de Integración

A continuación se muestra una aplicación de ejemplo que integra simpleSamlPhp y autentica con OpenAm.

3.1 Directorio de instalación.

La estructura del directorio de la aplicación deberá tener una estructura similar a la siguiente:

```
-rw-rw-r--  1 openam openam  1934 Mar 13 14:52 SamlAuthentication.php
-rw-rw-r--  1 openam openam   764 Mar 13 14:51 Configuration.php
-rw-rw-r--  1 openam openam   216 Mar 13 14:51 test.php
drwxr-xr-x  2 openam openam  4096 Mar 13 14:46 config
drwxr-xr-x  3 openam openam  4096 Mar 13 14:33 docs
drwxr-xr-x  5 openam openam  4096 Mar 13 14:33 lib
drwxr-xr-x  2 openam openam  4096 Mar 13 14:33 schemas
drwxr-xr-x  2 openam openam  4096 Mar 13 14:33 attributemap
drwxr-xr-x  2 openam openam  4096 Mar 13 14:33 metadata-templates
drwxr-xr-x 10 openam openam  4096 Mar 13 14:33 www
drwxr-xr-x  2 openam openam  4096 Mar 13 14:33 bin
drwxr-xr-x  2 openam openam  4096 Mar 13 14:33 dictionaries
drwxr-xr-x  2 openam openam  4096 Mar 13 14:33 metadata
drwxr-xr-x 47 openam openam  4096 Mar 13 14:33 modules
drwxr-xr-x  2 openam openam  4096 Mar 13 14:33 config-templates
drwxr-xr-x  2 openam openam  4096 Mar 13 14:33 extra
drwxr-xr-x  3 openam openam  4096 Mar 13 14:33 templates
```

3.2 Archivos PHP de ejemplo

Los ficheros de este ejemplo se han dejado sobre la carpeta “www” del directorio base del software. **Esta aplicación es meramente una prueba de conceptos y por ello se ha incluido dentro de la carpeta del software**. En el caso de querer integrar la aplicación fuera de las rutas aquí indicadas, se han de corregir adecuadamente las directrices “require_once”.

- **Configuration.php**

Esta clase contiene la configuración (información del SP y el IDP) necesaria para lógica de autenticación.

```
<?php



/**
 * Descripcion de Configuration
 * esta clase devuelve la configuracion de los actores sobre SAML2
 * @author sbayo
 */
class Configuration {
    //put your code here
    /**
     * Devuelve el identificador del service provider. El valor debe coincidir con
     * la configuracion del archivo config/authsources.php (default-sp)
     * @return SP
     */
    public static function getServiceProvider() {
        return 'default-sp';
    }
    /**
     * Devuelve el identificador del identity provider. El valor debe coincidir con
     * la configuracion del archivo metadata/saml20-idp-remote.php
     *
     * @return IDP
     */
    public static function getIdentityProvider(){
        return 'http://lbja.sia.es:8080/opensso';
    }

    /**
     * Devuelve el valor de la redirección con el logout
     */
    public static function getUrlLogout(){
        return './logoff.html';
    }
}

?>
```

- **SamlAuthentication.php**

Esta clase es un envoltorio para abstraer al programador de la lógica de autenticación de SAML.

```
<?php
require_once('../lib/_autoload.php');
require_once ('./Configuration.php');
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 * Description of SamlAuthentication
```

```
 *
 * @author sbayo
 */
class SamlAuthentication {

    private static $params = NULL;
    private static $as = NULL;

    /**
     * Comprueba la autenticacion para un usuario
     * @return si el usuario esta autenticado o no.
     */
    public static function isAuth() {
        $sp = Configuration::getServiceProvider();
        self::$as = new SimpleSAML_Auth_Simple($sp);
        $authenticated = self::$as->isAuthenticated();
        return $authenticated;
    }

    /**
     *  realiza la autetnicación y si se produce un error muestra un código 00001
     */
    public static function performAuth() {
        $idp = Configuration::getIdentityProvider();
        $sp = Configuration::getServiceProvider();
        try {
            if (isset(self::$as)) {
                //self::$as->login(array('saml:idp' => $idp,));
                self::$as->requireAuth(array('saml:idp' => $idp,));
            } else {
                self::$as = new SimpleSAML_Auth_Simple($sp);
                self::$as->requireAuth(array('saml:idp' => $idp,));
                //self::$as->login(array('saml:idp' => $idp,));
            }
        } catch (Exception $e) {
            die('ERROR:00001');
        }
    }

    /**
     * retorna el array de atributos de SAML devueltos por el IDP.
     * @return type
     */
    public static function getSamlData() {
        if (isset(self::$as)) {
            $attrs = self::$as->getAttributes();
            return $attrs;
        } else {
            die('ERROR:00002');
        }
    }
    /**
     * Devuelve el valor de un atributo de la aserción SAML determinado.
     * @param type $param Nombre del atributo
     * @return type  el valor del atributo (si es multivaluado solo se devolver el primero
(0)
     */
    public static function get($param) {
        if (isset(self::$as)) {
            $attrs = self::$as->getAttributes();
            $arrdata = $attrs[$param];
            if (isset($arrdata)) {
```

```
                $arrval = $arrdata[0];
                return $arrval;
            }
        }
    }
    /**
     * realiza el logout .
     *
     */
    public static function logout(){
        if (self::$as->isAuthenticated()){
            self::$as->logout(Configuration::getUrlLogout());
        }
    }

}

?>
```

- **auth.php**

Este script php implementa las llamadas a la capa de autenticación SAML.

```
<?php
require_once 'SamlAuthentication.php';
    if (!SamlAuthentication::isAuth()){
        SamlAuthentication::performAuth();
    }else{
        print_r(SamlAuthentication::getSamlData());
    }
?>
```

- **logoff.php**

Este script php implementa las llamadas a la capa de logout.

```
<?php
require_once 'SamlAuthentication.php';
    if (!SamlAuthentication::isAuth()){
        SamlAuthentication::logout();
    }
?>
```

- **logoffsuccess.html**

Esta página contiene el texto a mostrar en el caso de invocar al logout.

```
<!DOCTYPE html>
<html>
    <head>
        <title>logout correcto</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
        <div><a href="./AppExample.html">volver a la aplicacion de ejemplo </a></div>
    </body>
</html>
```

- **AppExample.html**

Página con los enlaces de login y logout.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Ejemplo de aplicacion PHP</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
        <span> <a href="./auth.php">Hacer login</a></span>
        <span> <a href="./logoff.php">Hacer logout</a></span>
    </body>
</html>
```