

## Integración con PAEC desde una aplicación PHP

### Configuración de la webapp

La adaptación de una webapp para ser securizada con PAEC consta de dos pasos:

1. Modificar en el código la obtención de datos de usuario, para leerlos desde PAEC.
2. Modificar los links de cierre de sesión, para que hagan un logout en PAEC.

### Obtención de los datos del usuario

Los datos del usuario se pueden leer en un array `$_SERVER`, con el prefijo `OIDC_CLAIM`:

```
$_SERVER['OIDC_CLAIM_preferred_username']
```

### Cierre de sesión en PAEC

Para cerrar la sesión, se debe hacer un GET al endpoint de logout de OIDC, y borrar la cookie de sesión, además del `session_destroy()`.

Link de logout:

```
<a href="https://tst.autenticar.gob.ar/auth/realms/<reino>
/protocol/openid-connect/logout?redirect_uri=http://<host>
/<app>/logout.php">Logout</a>';
```

Logout en PHP:

```
<?php
// Destruir todas las variables de sesión.
$_SESSION = array();
// Si se desea destruir la sesión completamente, borre también la
cookie de sesión.
// Nota: ¡Esto destruirá la sesión, y no la información de la
sesión! if (ini_get("session.use_cookies")) { $params
= session_get_cookie_params(); setcookie(session_name(),
'', time() -
42000,
$params["path"], $params["domain"],
$params["secure"], $params["httponly"]
);
// Borrado de la sesión mod_auth_openidc
setcookie('mod_auth_openidc_session', '', time() - 42000,
```



```
$params["path"], $params["domain"],
$params["secure"], $params["httponly"]
);
}
// Finalmente, destruir la sesión. session_destroy();
// Redirect al index. echo '<script
type="text/javascript"> window.location =
"http://localhost/php-5.x/index.php"
</script>';
?>
```

## Configuración del servidor Apache

### 1) Instalar y habilitar el módulo [mod\\_auth\\_openidc](#).

En conf/httpd.conf:

```
# Módulo para configurar un VirtualHost con
OpenID Connect.

LoadModule auth_openidc_module modules/mod_auth_openidc.so
```

### 2) En el VirtualHost, agregar:

```
OIDCProviderMetadataURL <openid_configuration>
OIDCClientID <client_id>
OIDCClientSecret <client_secret>
OIDCRedirectURI <redirect_uri>
OIDCRemoteUserClaim preferred_username
OIDCCryptoPassphrase clave <Location
/<app>/secure/> AuthType openid-connect
    Require valid-user
</Location>
```

donde:

OIDCProviderMetadataURL es la URL de la configuración OpenID Connect.

En PAEC, se encuentra en:

```
https://tst.autenticar.gob.ar/auth/realms/<reino>/well-known/openid-configuration
```

OIDCClientID: ID del cliente OIDC.

OIDCClientSecret **Secreto del cliente OIDC.**

OIDCRedirectURI: URL donde se recibe el login desde PAEC. Por restricciones de Apache HTTP Server, no puede accederse a esta URL directamente con un GET

OIDCRemoteUserClaim: El nombre de usuario a mostrar.

OIDCCryptoPassphrase: Passphrase requerida.

El tag `Location` define la URL securizada de la app (debe excluir la landing page).

## Ejemplo de configuración de Servidor Apache2

En el siguiente punto se detalla cómo securizar una aplicación PHP hosteada en Apache2 sin tener que modificar el código de la aplicación.

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    OIDCProviderMetadataURL
    [http://10.1.1.1:8280/auth/realms/demo/.well-known/openid-configuration]
    OIDCClientID [localhost]
    OIDCClientSecret [b9dbc992-4376-4e39-af-6f5dff88dd99]
    OIDCRedirectURI
    [http://localhost/appapache/index.html]
    OIDCCryptoPassphrase [clave]
    <Location /app-apache/>
        AuthType openid-connect
        Require valid-user
    </Location>

</VirtualHost>
```