

Ejemplo de uso del flujo OAuth2 usando la librería oauth2lib:

```
1. <?php
2. include 'oauth_client/src/OAuth.php';
3. include 'phpPoA-2.4/PoA.php';
4. $poa = new PoA("", "");
5. $auth = $poa->authenticate();
6. $attrs = $poa->getAttributes();
7. $filePAPI = dirname(dirname(__FILE__)). '/oauth2lib/oauth_client/config/clientConfig.xml';
8. $oauthPAPI = new OAuth($filePAPI);
9. $oauthPAPI->requestAccessToken($attrs);
10. $oauthPAPI->requestResourceRS($oauthPAPI->getRs(), $oauthPAPI->getRequest_type());
11. $contentPAPI = $oauthPAPI->getResource();
12. ?>
```

1. Una vez obtenida la aserción (mediante phpPoA para el caso PAPI), se llama al constructor de la clase OAuth, el cual, entre otras cosas, carga el archivo de configuración que se le pasa como parámetro.
2. Se llama al método de clase **requestAccessToken**, el cual recibe como parámetro un array con los atributos de la aserción PAPI (obtenido a través del método `getAttributes()` de la clase PoA de phpPoA) o una cadena con la aserción SAML2.
3. Este método, internamente, se encargará de hacer la petición del token de acceso al Servidor de Autorización que tenga configurado ese cliente. Aunque primero comprobará que el usuario que va a hacer la petición no tenga ya un token de acceso guardado (en la variable `$_SESSION` o en una base de datos dba), en cuyo caso se omite la petición al AS. En caso de haber petición, seguirá los siguientes pasos:
 - 3.1. Si el usuario no tiene token de acceso guardado, se usa el método de la clase OAuthClient, **doAccessTokenRequest**, que recibe como parámetros el AS, el scope, la aserción, el `assertion_type` y el `grant_type`, todos ellos sacados de la configuración del cliente.
 - 3.2. Este método primero comprueba que la petición se hace de manera segura, mediante protocolo https, genera la petición y la manda mediante cURL al AS. Una vez obtenida la respuesta, se procesa y se guarda el token en el atributo de clase *access_token*.
4. Una vez obtenido el token de acceso, se pasará a pedir el recurso mediante la función **requestResourceRS**, el cual recibe como parámetros la URL del Servidor de Recursos y el tipo de petición que se hará (puede ser `HTTP_Authorization_Header`, `URI_Query_Parameter` o `Form-Encoded_Body_Parameter`). El Servidor de Recursos comprobará que el token de acceso es válido y devolverá el recurso solicitado, al cual se puede acceder a través del método **getResource** de la clase OAuth.

