

Servicios web - III

Ahora que ya sabemos de que hablamos ([Servicios Web – I](#) y [Servicios Web – II](#)) os proponemos un ejercicio. Nosotros ya lo hicimos en su momento y es enriquecedor para la nueva etapa que **Velneo V7** introducirá con la llegada de **QML** y **JavaScript**.

Intentaremos explicar paso a paso como realizar un cliente de **twitter** básico mediante **OAuth** y **API's REST/XML**.

Conceptos básicos OAuth: Usuarios, OAuth Consumer y OAuth Service Provider

Para poder hablar de **OAuth**, se necesitan tres partes, un servidor o proveedor de servicios, usuarios y un consumidor:

- **OAuth Service Provider** o proveedor de servicios OAuth: Sitios o servicios web que contienen información de usuarios cuyo acceso es restringido. Algunos de los más conocidos son **Facebook, Twitter o Youtube**. Estos proveedores ponen a disposición de los desarrolladores una API que soporta el protocolo de autenticación OAuth.
- **Usuarios**: Sin los usuarios, no existiría OAuth. Por usuario se entiende cualquier persona que tiene una cuenta de usuario en un *Service Provider*.
- **OAuth Consumer**: Cualquier sitio o aplicación web, móvil o de escritorio que solicita permiso a un usuario para acceder a sus datos de acceso restringido que alberga un *Service Provider*. El usuario puede autorizar o denegar el acceso del *consumer* a sus datos.

API Key y Callback URL

Cada *OAuth Service Provider* os proporcionará un API Key (un *string* de letras y números) para identificar que las peticiones que recibe mediante su API vienen de un *OAuth Consumer* autorizado, es decir, vuestra aplicación.

A su vez, cada *OAuth Service Provider* os pedirá que indiquéis un *Callback URL*, es decir, una dirección URL que apunte a un archivo de vuestra aplicación el cual se encargará de procesar la respuesta de autorización (o desautorización) de acceso a los datos de la cuenta del Usuario en el *OAuth Service Provider*.

Fujo en OAuth 1.0a

- Obtenemos un *Request Token*
- Solicitamos la autorización del *Usuario* (para acceder a los datos de su cuenta) enviándole a una página especial de login del *Service Provider*.
- Cambiamos el *Request Token* por un *Access Token*

Punto de partida

Como punto de partida utilizaremos un ejemplo implementado íntegramente en QML y javascript

(<https://gitorious.org/qt-qml-demo-playground/qt-qml-demo-playground/trees/master/twitter-oauth>)

Este ejemplo contiene dos librerías javascript importantes:

- **sha1.js**: Una librería que implementa SHA-1
- **OAuth.js**: Una librería que implementa OAuth 1 (no usar en producción)

Obtenemos un *Request Token*

Realizamos la solicitud del token (OAuth.qml – línea 70):

```
70 function stepOne(){
71     var xhr = OAuthLogic.createOAuthHeader("POST", "https://api.twitter.com/oauth/request_token",
72     [{"oauth_callback","oob"}]);
73     xhr.onreadystatechange = function() {
74         if (xhr.readyState == XMLHttpRequest.DONE) {
75             //console.log(xhr.status+'\n'+xhr.getAllResponseHeaders()+'\n'+xhr.responseText+xhr.responseXML);
76             var response = xhr.responseText.split('&');
77             if(response.length != 3)
78                 return;
79             token = response[0].split('=')[1];
80             secret = response[1].split('=')[1];
81             if(response[2].split('=')[1] != 'true')
82                 console.log("Error: " + response[2]);
83             step = 2;
84             webItem.url = "https://api.twitter.com/oauth/authorize?oauth_token="+token;
85         }
86     }
87     xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
88     xhr.setRequestHeader("Accept-Language", "en");
89     xhr.send();
90 }
```

Obteniendo el token y la clave secreta (línea 78 y 79).

Solicitamos la autorización del Usuario (para acceder a los datos de su cuenta) enviándole a una página especial de login del Service Provider.

Preparamos la petición de autorización (línea 83).

Realizamos la petición de autorización (línea 88).

Rellenamos los datos de autenticación (línea 94 y 95).

Aceptamos el formulario (Línea 96)

```
90 function stepTwo(){
91     step = 3;
92     var page = webItem.html;
93     //console.log("Step2\n" + page);
94     webItem.evaluateJavaScript("document.getElementById(\"username_or_email\").value = \"\" + username + "\"");
95     webItem.evaluateJavaScript("document.getElementById(\"session[password]\").value = \"\" + password + "\"");
96     webItem.evaluateJavaScript("document.getElementsByTagName(\"input\")[5].click();")
97 }
```

Cambiamos el *Request Token* por un *Access Token*

Para ello recogemos el pin (línea 99).

Componemos una nueva solicitud (en este caso para el access_token) pasándole como parámetros el pin, token y secret (línea 100).

```
98 function stepThree(){
99     var pin = webItem.evaluateJavaScript("document.getElementById(\"oauth_pin\").innerHTML;").replace(/[ \n]*/g, "");
100     var xhr = OAuthLogic.createOAuthHeader("POST", "https://api.twitter.com/oauth/access_token", [{"oauth_verifier":pin}],
("token":token,"secret":secret));
101     xhr.onreadystatechange = function() {
102         if (xhr.readyState == XMLHttpRequest.DONE) {
103             //console.log("STEP 3\n"+xhr.status+"\n"+xhr.getAllResponseHeaders()+"\n"+xhr.responseText+xhr.responseXML);
104             var response = xhr.responseText.split('&');
105             if(response.length != 4)
106                 return;
107             token = response[0].split('=')[1];
108             secret = response[1].split('=')[1];
109             //username = response[3].split('=')[1];
110             step = 0;
111             authorized = true;
112             authenticationCompleted();
113         }
114     }
115     xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
116     xhr.setRequestHeader("Accept-Language", "en");
117     xhr.send();
118 }
```

Ya estamos autenticados.

Realizamos las llamadas a la API

Ahora que estamos autenticados podemos realizar cualquier llamada a la API de twitter, por ejemplo... a la timeline.

```
53 function reload() {
54     var url = '';
55     if(mode == 'self')
56         url = 'http://api.twitter.com/1/statuses/home_timeline.xml'
57     else if(mode == 'user')
58         url = 'http://api.twitter.com/1/statuses/user_timeline.xml?screen_name='+tags
59     else
60         url = 'http://api.twitter.com/1/statuses/public_timeline.xml'
61     var xhr:
62     if(oauth.authorized){
63         xhr = OAuthLogic.createOAuthHeader("GET", url, undefined, {"token":oauth.token, "secret":oauth.secret});
64     }else{
65         xhr = new XMLHttpRequest;
66         xhr.open("GET", url);
67     }
68     xhr.onreadystatechange = function() {
69         if (xhr.readyState == XMLHttpRequest.DONE) {
70             xmlModel.xml = xhr.responseText;
71         }
72     }
73     xhr.send();
74 }
```

Esperamos que esta introducción os sea de ayuda para hacer vuestro primeros pinitos con **OAuth**.

¿Te gustaría tener un cliente de Facebook o twitter integrado en tú aplicación empresarial?

Desde hace unos meses estamos trabajando en sendos clientes de **twitter** y **facebook** para **PaaSOS**. Estos dos clientes están implementados íntegramente mediante **QML** y **JavaScript**. Llegado el momento serán 100% funcionales en **Velneo V7**.



The screenshot shows a Twitter feed with the following tweets:

- Top Tweet:** "Annus Horribilis. Las ventas en España caen un 18% en 2011" <http://t.co/CvGtHgct>
- RT @UPyD:** Diputados que cobran un extra de 1.800 euros al mes para alojamiento pese a tener casa en Madrid <http://t.co/3CjKx9qj>
- RT @locodelpelorojo:** Los diputados q retiran la ayuda al alquiler para jóvenes cobran 1.800€ al mes como ayuda a la vivienda <http://t.co/...>
- Jobs merece reconocimiento x romper molde d prototipo d ejecutivo norteamericano, no sólo x fabricar productos muy superiores #bio_SteveJobs**
- I just ousted Maria P. as the mayor of Luis Lopez Allue on @foursquare!** <http://t.co/f3XBfPxn>
- Muy interesante! "@gfiorelli1: Interesting PDF study about Search Click Through Rate >**
- De los creadores de <http://t.co/BuqYc3dd>, "Creo que la cosa va para largo..."** <http://t.co/h9lOtaur>

The bottom navigation bar includes icons for: Home, Status, Mentions, Lists, Search, Favourites, Refresh, More, and Hide.

Se utilizan las **API REST** de **twitter** y **facebook**. También se utilizan los protocolos de Autenticación **OAuth 1a y 2** por lo son un buen punto de partida para el estudio de la implementación mediante javascript de sendos clientes para **API's REST** y **OAuth** .