Autenticación de API con OAuth

¿Qué es?

OAuth 1.0a define un protocolo que permite a los usuarios de aplicaciones autorizar a aplicaciones que consuman sus APIs sin necesidad de que las contraseñas viajen en las peticiones.

> Documentación relacionada

Implementación

La implementación concreta que se utiliza es autenticación en un sólo paso, por lo que no es necesaria la obtención de claves adicionales (request_token o access_token), se puede seguir una explicación aquí:

http://oauthbible.com/#oauth-10a-one-legged

Pruebas

En la consola de pruebas: https://api.cnmc.gob.es/

Se pueden ver todos los parámetros, puesto que utiliza autenticación OAuth.

¿Cómo autenticar solicitudes al API de la sede electrónica con el protocolo OAuth?

Guía detallada

Intentamos acceder al recurso securizado:

`https://api.cnmc.gob.es/test/v1/echoseguro?m=Estoesunaprueba`

1. El responsable de la aplicación cliente ha debido solicitar previamente, a través de la sede electrónica, sus credenciales o datos de autorización, por ejemplo: customerKey=dpf43f3p2l4k3l03 customerSecret=kd94hf93k423kf44
2. El cliente también no necesita ejecutar el flujo OAuth y debe usar: access token = null y token secret = null
3. Para firmar la petición, el cliente den¡be usar el algoritmo **HMAC-SHA1**
4. Y este genera la siguiente: nonce string kllo9940pd9333jh y este timestamp 1191242096
5. Los parámetros a enviar, incluyendo los utilizados por el propio servicio, quedan de la siguiente manera:

Name	Value
oauth_consumer_key	dpf43f3p2l4k3l03
oauth_token	null
oauth_nonce	kllo9940pd9333jh
oauth_timestamp	1191242096
oauth_signature_method	HMAC-SHA1
oauth_version	1.0
m	Estoesunaprueba

> Los parámetros OAuth se pueden enviar, esto es para la generación de la firma, como un authentication header (http://tools.ietf.org/html/rfc5849#section-3.5.1)

6. Estos parámetros se transforman siguiendo el RFC de OAuth (http://tools.ietf.org/html/rfc5849#section-3.5) siguiendo los siguientes pasos:
    1. Codificación de los mismos en UTF-8

2. Codificar los parámetros siguiendo URL-Encoding (http://en.wikipedia.org/wiki/Percent-encodinghttp://tools.ietf.org/html/rfc5849#section-3.6)
3. Se ordenan basados en su codificación por URL-Encoding

Name	Value
m	Estoesunaprueba
oauth_consumer_key	dpf43f3p2l4k3l03
oauth_nonce	kllo9940pd9333jh
oauth_signature_method	HMAC-SHA1
oauth_timestamp	1191242096
oauth_token	nnch734d00sl2jdk
oauth_version	1.0

7. Una vez transformados se concatenan en una QueryString:

```
m=Estoesunaprueba&oauth_consumer_key=dpf43f3p2l4k3l03&oauth_nonce=kllo9940pd9333jh&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1191242096&oauth_token=nnch734d00sl2jdk&oauth_version=1.0
```

8. Se normaliza la URL de la petición utilizando la especificación: http://tools.ietf.org/html/rfc5849#section-3.4.1.2 Ej.:

Full URL	https://api.cnmc.gob.es/test/v1/echoseguro
Normalized URL	https://api.cnmc.gob.es/test/v1/echoseguro

9. Para crear la "Signature Base String" se concatenan todas estas partes precedidas del método HTTP solicitado, que es una parte muy importante en los servicios REST:

```
GET&http%3A%2F%2Fapi.sede.cnmc.gob.es%2Ftest%2Fecho&m%3DEstoesunaprueba%26oauth_consumer_key%3Ddpf43f3p2l4k3l03%26oauth_nonce%3Dkllo9940pd9333jh%26oauth_signature_method%3DHMAC-SHA1%26oauth_timestamp%3D1191242096%26oauth_token%3Dnnch734d00sl2jdk%26oauth_version%3D1.0
```

10. Para calcular la firma HMAC-SHA1 usamos dos claves: client secret y token secret para el key del algoritmo HMAC-SHA1. Cada clave se codifica con: UTF8-encoded, URL-encoded y se concatena en una única cadena usando '&' como separador, aunque estén vacíos (persection 3.4.2).

```
kd94hf93k423kf44&pfkkdhi9sl3r4s00
```

11. Con la "Signature Base String", el texto HMAC-SHA1 *concatenamos las claves el cliente deberá generar la firma* (RFC section 3.4.2). El algoritmo HMAC-SHA1 genera una cadena de bytes. Debe estar codificada base64-encoded con '=' para asegurar el "padding" (ver RFC 2045 section 6.8):

```
tR3+Ty81lMeYAr/Fid0kMTYa/WM=
```

12. La firma calculada es entonces añadida a la petición usando el parámetro 'oauth_signature'. Una vez esta firma haya sido verificada por el servidor ya no se debe incluir en el flujo de firma y no es parte de "Signature Base String". Cuando se incluya en la cabecera HTTP debe estar codificada del mismo modo que se transmiten el resto de parámetros.

Name	Value
oauth_signature	tR3+Ty81lMeYAr/Fid0kMTYa/WM=

13. Aunque OAuth no especifica directamente cómo se deben incluir estos parámetros en la petición, al definirlos en la firma para su verificación por el "Service Provider" implícitamente indica cuáles deben incluirse en la petición.

    Los parámetros OAuth pueden incluirse en uno de estos lugares:

    1. URL query (ver RFC 3986 section 3),
    2. OAuth 'Authorization' header (ver section 3.5.1),
    3. En un single-part 'application/x-www-form-urlencoded' POST body (como se define en HTML4).

    Los parámetros firmados non-OAuth pueden incluirse de alguna de estas maneras:

1. Como un elemento de la URL query
2. En un single-part 'application/x-www-form-urlencoded' POST body.

Se recomienda que cuando sea posible los parámetros OAuth se incluyan en OAuth 'Authorization' header y no se incluyan más parámetros en el header.

Usando URL query para los parámetros non-OAuth y OAuth 'Authorization' header para OAuth la petición OAuth-signed HTTP queda así:

```
GET /test/echo?m=Estoesunaprueba HTTP/1.1
Host: http://api.cnmc.gob.es:80/test/v1/echoseguro
Authorization: OAuth realm="http://api.cnmc.gob.es/test/v1/echoseguro",
 oauth_consumer_key="dpf43f3p2l4k3l03",
 oauth_token="nnch734d00sl2jdk",
 oauth_nonce="kllo9940pd9333jh",
 oauth_timestamp="1191242096",
 oauth_signature_method="HMAC-SHA1",
 oauth_version="1.0",
 oauth_signature="tR3%2BTy81lMeYAr%2FFid0kMTYa%2FWM%3D"
```

Documentación Relacionada

- http://tools.ietf.org/html/rfc5849
- http://oauth.net/documentation/
- http://oauth.net/core/1.0a/
- http://hueniverse.com/oauth/
- http://nouncer.com/oauth/authentication.html
- http://hueniverse.com/oauth/guide/terminology/
- http://hueniverse.com/2007/10/04/beginners-guide-to-oauth-part-i-overview/
- http://projects.spring.io/spring-security-oauth/docs/twolegged.html
- 2-Legged http://tools.ietf.org/pdf/draft-ietf-oauth-v2-http-mac-01.pdf