Package: ROAuth (via r-universe)

September 5, 2024

Title R Interface For OAuth

Description Provides an interface to the OAuth 1.0 specification allowing users to authenticate via OAuth to the server of their choice.

Version 0.9.6

Author Jeff Gentry <geoffjentry@gmail.com>, Duncan Temple Lang <duncan@r-project.org>

Maintainer Pablo Barbera <pablo.barbera@nyu.edu>

Depends R (>= 2.12.0)

Imports RCurl, digest, methods

License Artistic-2.0

Date/Publication 2011-06-03 06:06:18

Repository https://pablobarbera.r-universe.dev

RemoteUrl https://github.com/pablobarbera/roauth

RemoteRef HEAD

RemoteSha 75046394eb0b40d755ccf76fe5f788c9600a491a

Contents

`OAuth-class` *Class "OAuth": A class to manage OAuth authentication*

Description

Class `OAuth` wraps and handles OAuth handshakes and signatures for the user within R

Details

The OAuth class is currently implemented as a reference class. An instance of a generator for this class is provided as a convenience to the user as it is configured to handle most standard cases. To access this generator, use the object `OAuthFactory`. See the `examples` section below for an example of how to instantiate an object of class `OAuth`.

In almost all cases, saving an `OAuth` object after handshake and loading it into future sessions will allow it to remain authorized without needing any manual intervention that might have been performed initially, such as the PIN step with Twitter authentication. Use the function `save` to save the credential object to a file and then `load` in another R session to bring it back in - there should be no reason to undergo another handshake by doing this.

The `needsVerifier` argument is optional and defaults to `TRUE`. In almost all cases, the default should be used, the option is primarily provided to enable the examples as the keys provided by the examples are already signed. If you feel that you're in a situation where this should be set to `FALSE`, it's best to double check this.

The `signMethod` to the `handshake` method tells the system which OAuth signature hash to use, one of `HMAC` for `HMAC-SHA1` (default), `RSA` for `RSA-SHA1` (not implemented), or `text` for `plaintext`.

The `customHeader` argument to `OAuthRequest` can be used to pass additional HTTP header commands to the underlying request.

The `curl` arguments can be used to provide a custom curl header, defaulting to a generic `getCurlHandle` call.

Fields

`consumerKey`: The consumer key provided by your application

`consumerSecret`: The consumer secret provided by your application

`needsVerifier`: Whether or not this `OAuth` needs the verification step. Defaults to `TRUE`

`handshakeComplete`: Whether or not the handshaking was successfully completed

`requestURL`: The URL provided for retrieving request tokens

`authURL`: The URL provided for authorization/verification purposes

`accessURL`: The URL provided for retrieving access tokens

`oauthKey`: For internal use

`oauthSecret`: For internal use

`verifier`: For internal use

`signMethod`: For internal use

Methods

`handshake(signMethod='HMAC', curl=getCurlHandle(), browseUrl=TRUE, ...)`: Performs an OAuth handshake using the information provided. If `browseUrl` is `TRUE`, the `browseURL` function will be called on the authentication URL

`isVerified()`: Returns the current verification status

`OAuthRequest(URL, params=character(), method="GET", customHeader=NULL, curl=getCurlHandle(), ...)`: Will sign the URL provided and make an HTTP request using either `POST` or `GET`, determined by `method`, defaulting to `GET`. NOTE: The `URL` argument will be run through `URLencode`, so doing this step beforehand might lead to bad behavior!

`initialize(needsVerifier, ...)`: For internal use

Extends

All reference classes extend and inherit methods from `"envRefClass"`.

Author(s)

Jeff Gentry

References

liboauth: http://liboauth.sourceforge.net/

See Also

`setRefClass`

Examples

```
  ## This example uses a test case from liboauth and the
  ## keys are already pre-signed.  This is an example of
  ## one of the few times \code{needsVerifier} would be \code{FALSE}.
 ## Not run:
  reqURL <- "http://term.ie/oauth/example/request_token.php"
  accessURL <- "http://term.ie/oauth/example/access_token.php"
  authURL <- "NORMALLY YOU NEED THIS"
  cKey <- "key"
  cSecret <- "secret"
  testURL <- "http://term.ie/oauth/example/echo_api.php?method=foo bar"

  credentials <- OAuthFactory$new(consumerKey=cKey,
                                  consumerSecret=cSecret,
                                  requestURL=reqURL,
      accessURL=accessURL,
      authURL=authURL,
                                  needsVerifier=FALSE)
  credentials$handshake()
  ## the GET isn't strictly necessary as that's the default
  credentials$OAuthRequest(testURL, "GET")

## End(Not run)
```

Index