

andris9 / xoauth2

Watch

3

Star

40

Fork

14

Code

Issues1

Pull requests0

Projects0

Pulse

Graphs

XOAuth2 token generation with node.js

32 commits

1 branch

12 releases

7 contributors

MIT

Branch: master

New pull request

Find file

Clone or download

andris9 v1.2.0

Latest commit 07da76d on Jul 29

src	Remove dependency on jsonwebtoken.	4 months ago
test	Always use seconds for timeout. Bumped version to 1.0.0	2 years ago
.gitignore	Always use seconds for timeout. Bumped version to 1.0.0	2 years ago
.jshintrc	Adds support for yahoo oauth2	a year ago
.travis.yml	Updated travis config, added gitter webhook	2 years ago
CHANGELOG.md	v1.2.0	3 months ago
Gruntfile.js	Always use seconds for timeout. Bumped version to 1.0.0	2 years ago
LICENSE	Always use seconds for timeout. Bumped version to 1.0.0	2 years ago
README.md	Add support for Google service accounts.	4 months ago
package.json	v1.2.0	3 months ago

README.md

xoauth2

XOAuth2 token generation with node.js

Installation

```
npm install xoauth2
```

Usage

xoauth2 generates XOAUTH2 login tokens from provided Client and User credentials.

Use `xoauth2.createXOAuth2Generator(options)` to initialize Token Generator

Possible options values:

- `user` *(Required)* User e-mail address
- `accessUrl` *(Optional)* Endpoint for token generation (defaults to <https://accounts.google.com/o/oauth2/token>)
- `clientId` *(Required)* Client ID value
- `clientSecret` *(Required)* Client secret value

- **refreshToken** (*Required*) Refresh token for an user
- **accessToken** (*Optional*) initial access token. If not set, a new one will be generated
- **timeout** (*Optional*) TTL in **seconds**
- **customHeaders** (*Optional*) custom headers to send during token generation request `yahoo` requires `Authorization: Basic Base64(clientId:clientSecret)`
- **customParams** (*Optional*) custom payload to send on getToken request `yahoo` requires `redirect_uri` to be specified

See <https://developers.google.com/identity/protocols/OAuth2WebServer#offline> for generating the required credentials

For Google service account the option values are:

- **service** (*Required*) Service account email.
- **user** (*Required*) User e-mail address
- **scope** (*Required*) OAuth2 scope.
- **privateKey** (*Required*) Private key issued for the service account in PEM format, as a string.
- **serviceRequestTimeout** (*Optional*) Expiration value to use in the token request in **seconds**. Maximum is 3600.
- **accessUrl** (*Optional*) Endpoint for token generation (defaults to <https://accounts.google.com/o/oauth2/token>)
- **accessToken** (*Optional*) initial access token. If not set, a new one will be generated
- **timeout** (*Optional*) TTL in **seconds**
- **customHeaders** (*Optional*) custom headers to send during token generation request
- **customParams** (*Optional*) custom payload to send on getToken request

Methods

Request an access token

Use `xoauth2obj.getToken(callback)` to get an access token. If a cached token is found and it should not be expired yet, the cached value will be used.

Request for generating a new access token

Use `xoauth2obj.generateToken(callback)` to get an access token. Cache will not be used and a new token is generated.

Update access token values

Use `xoauth2obj.updateToken(accessToken, timeout)` to set the new value for the xoauth2 access token. This function emits 'token'

Events

If a new token value has been set, 'token' event is emitted.

```

xoauth2obj.on("token", function(token){
  console.log("User: ", token.user); // e-mail address
  console.log("New access token: ", token.accessToken);
  console.log("New access token timeout: ", token.timeout); // TTL in seconds
});

```

Example

```

var xoauth2 = require("xoauth2"),
    xoauth2gen;

xoauth2gen = xoauth2.createXOAuth2Generator({
  user: "user@gmail.com",
  clientId: "{Client ID}",
  clientSecret: "{Client Secret}",
  refreshToken: "{User Refresh Token}",

```

```
    customHeaders: {
      "HeaderName": "HeaderValue"
    },
    customPayload: {
      "payloadParamName": "payloadValue"
    }
  });

// ... or for a Google service account
xoauth2gen = xoauth2.createXOAuth2Generator({
  user: "user@gmail.com",
  service: '{Service Email Address}',
  scope: 'https://mail.google.com/',
  privateKey: '{Private Key in PEM format}'
});

// SMTP/IMAP
xoauth2gen.getToken(function(err, token){
  if(err){
    return console.log(err);
  }
  console.log("AUTH XOAUTH2 " + token);
});

// HTTP
xoauth2gen.getToken(function(err, token, accessToken){
  if(err){
    return console.log(err);
  }
  console.log("Authorization: Bearer " + accessToken);
});
```

License

MIT

