Spendgo Identity & OAuth 2.0

#server-to-server

Overview

The Spendgo APIs use the OAuth 2.0 protocol for authentication and authorization. OAuth 2.0 provides API security via scoped access tokens, and the Spendgo Identity Server provides user authentication and single sign-on (SSO) functionality. OAuth 2.0 is the preferred model when the member is required to sign in using a third-party identity before they can access any sensitive member profile information.

To begin, Spendgo will provide you with OAuth 2.0 client credentials. Then your client application requests an access token from Spendgo, extracts a token from the response, and uses the token with the Spendgo API that you want to access.

Getting Your Credentials

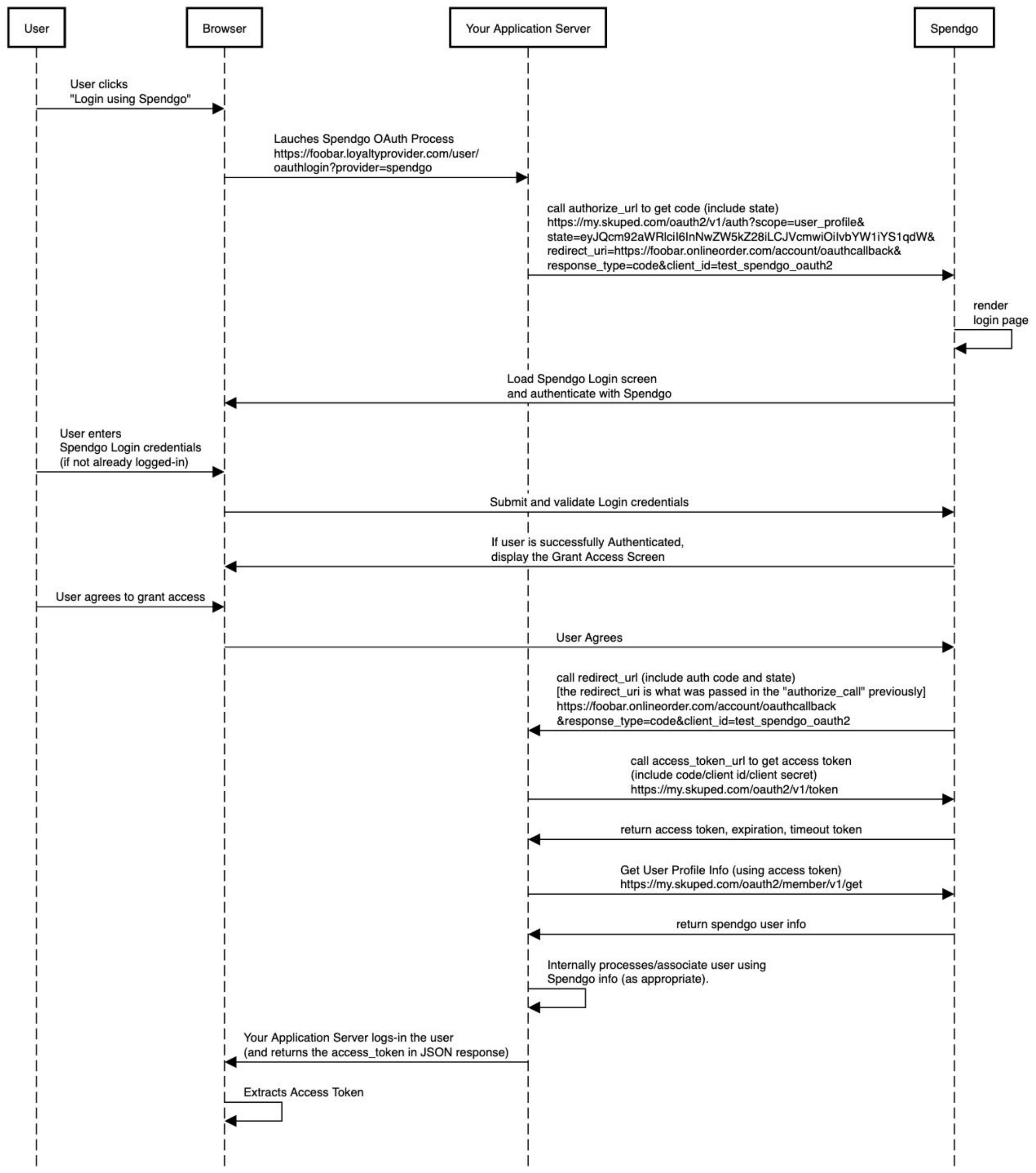
Spendgo will provide you with the below credentials to be used in the OAuth 2.0 process.

- Client ID
- Shared Client Secret
- Authorize URL

Note: *Security credentials are environment and account-specific. If you have access to multiple Spendgo accounts, use the credentials that are associated with the account and environment that you want to access.*

You need to share with your Spendgo contact a list of your redirect URLs to be whitelisted. Once the OAuth 2.0 process is complete, it will only recognize the whitelisted URLs as the allowed callback URLs.

Workflow


User
Browser
Your Application Server
Spendgo
User clicks "Login using Spendgo"
Lauches Spendgo OAuth Process https://foobar.loyaltyprovider.com/user/oauthlogin?provider=spendgo
call authorize_url to get code (include state) https://my.skuped.com/oauth2/v1/auth?scope=user_profile& state=eyJQcm92aWRlciI6InNwZW5kZ28iLCJVcmwiOiIvbYW1iYS1qdW& redirect_uri=https://foobar.onlineorder.com/account/oauthcallback& response_type=code&client_id=test_spendgo_oauth2
render login page
Load Spendgo Login screen and authenticate with Spendgo
User enters Spendgo Login credentials (if not already logged-in)
Submit and validate Login credentials
If user is successfully Authenticated, display the Grant Access Screen
User agrees to grant access
User Agrees
call redirect_url (include auth code and state) [the redirect_uri is what was passed in the "authorize_call" previously] https://foobar.onlineorder.com/account/oauthcallback &response_type=code&client_id=test_spendgo_oauth2
call access_token_url to get access token (include code/client id/client secret) https://my.skuped.com/oauth2/v1/token
return access token, expiration, timeout token
Get User Profile Info (using access token) https://my.skuped.com/oauth2/member/v1/get
return spendgo user info
Internally processes/associate user using Spendgo info (as appropriate).
Your Application Server logs-in the user (and returns the access_token in JSON response)
Extracts Access Token

All applications follow the OAuth 2 standard defined workflow when accessing using OAuth 2.0. At a high-level OAuth 2.0 process will follow these steps:

1. User clicks login

The authorization sequence begins when the user clicks login within a regular web application.

2. Redirect to Spendgo Identity Server

Your application server generates and makes the server-side redirect request to authorize. [Read more]

Authorization Url

```
https://my.<ENVIRONMENT>.com/oauth2/v1/auth?response_type=code&client_id=<C
LIENT_ID>&redirect_uri=<YOUR_REDIRECT_URL>&state=<SECURE_RANDOM>&scope=user
_profile
```

Url parameters

`response_type`	string	Required. The value has to be “code”.
`client_id`	string	Required. The client ID provided by Spendgo.
`redirect_uri`	string	Required. This is the URL that the partner expects to be called after the SSO Authentication process is completed. This is the URL that was shared initially by the Partner and whitelisted by Spendgo.
`scope`	string	Required. The value needs to be “user_profile”. The information pertaining to this scope is name, email address and phone number.
`state`	string	Required. An unguessable random string. It is used to protect against cross-site request forgery attacks.

Example Url

```
https://my.skuped.com/oauth2/v1/auth?scope=user_profile&state=eyJQcm92aWRlc
iI6InNwZW5kZ28iLCJVcmwiOiIvbYW1iYS1qdW&redirect_uri=https://foobar.onlineor
der.com/account/oauthcallback&response_type=code&client_id=test_spendgo_oau
```

```
th2
```

3. User decides whether to grant permission to your app

Spendgo handles the user authentication where the user logs in with their Spendgo account. After logging in, the user is asked whether they are willing to grant permissions that your application is requesting. This process is called *user consent*.

Response

If the user grants permission, the Spendgo Server generates the authorization code and invokes the redirect URI with the code and the state as query parameters as defined in the OAuth 2.0 specification. Your application can use this authorization code to obtain an access token in a subsequent step.

If the user does not grant the permission, the server returns an error and redirects back to your application without authentication. A user only has to grant permission once to an application — on subsequent logins the grant screen is not shown.

Example response (200)

```
https://foobar.onlineorder.com/account/oauthcallback?code=59a825a3ba9a3482e
6b9cd77072d043932237bba1589ec252476563f65c1625dde08344&state=eyJQcm92aWRlci
I6InNwZW5kZ28iLCJVcmwiOiIvbYW1iYS1qdW
```

3. Exchange authorization code for an access token

After the successful authentication, your application will exchange the authorization code for an access token. The access token can be used to fetch customer information by your application server.

HTTP request

```
POST https://my.<ENVIRONMENT>.com/oauth2/v1/token
```

JSON Body

`grant_type`	The authorization code returned after a successful authentication request for a member. The value has to be "authorization_code".
`client_id`	The API Key shared by Spendgo.

`client_secret`	string	Base 64 encoded key.
`redirect_uri`	string	Required. This is the URL that the partner expects to be called after the SSO Authentication process is completed. This is the URL that was shared initially by the Partner and whitelisted by Spendgo.
`code`	string	The authorization code returned in the prior step.

Example request

```
curl -k -v -X POST -d 'grant_type=authorization_code' -d
'client_id=test_spendgo_oauth2' -d
'client_secret=<BASE64ENCODED_SPENDGO_PARTNER_SECRET_KEY>' -d
'redirect_uri=https://foobar.onlineorder.com/account/oauthcallback' -d
'code=59a825a3ba9a3482e6b9cd77072d043932237bba1589ec252476563f65c1625dde083
44' 'https://my.skuped.com/oauth2/v1/token'
```

In the example above, the `<BASE64ENCODED_SPENDGO_PARTNER_SECRET_KEY>` must be replaced with a valid secret key in order to test.

Returns

Returns an access token that can be used to access a Spendgo API.

JSON response

`access_token`	string	Access token.

Example response (200)

```
{
      "access_token":
"6915ab99857fec1e6f2f6c078583756d0c09d7207750baea28dfbc3d4b0f2cb80"
}
```

4. Access customer information using the access token

After your application server obtains an access token, it invokes the Spendgo Member API using the access token and sending it as “Authorization: Bearer” in the HTTP header.

Related Documents

- Add Transaction Using Signed Requests
- API Reference