

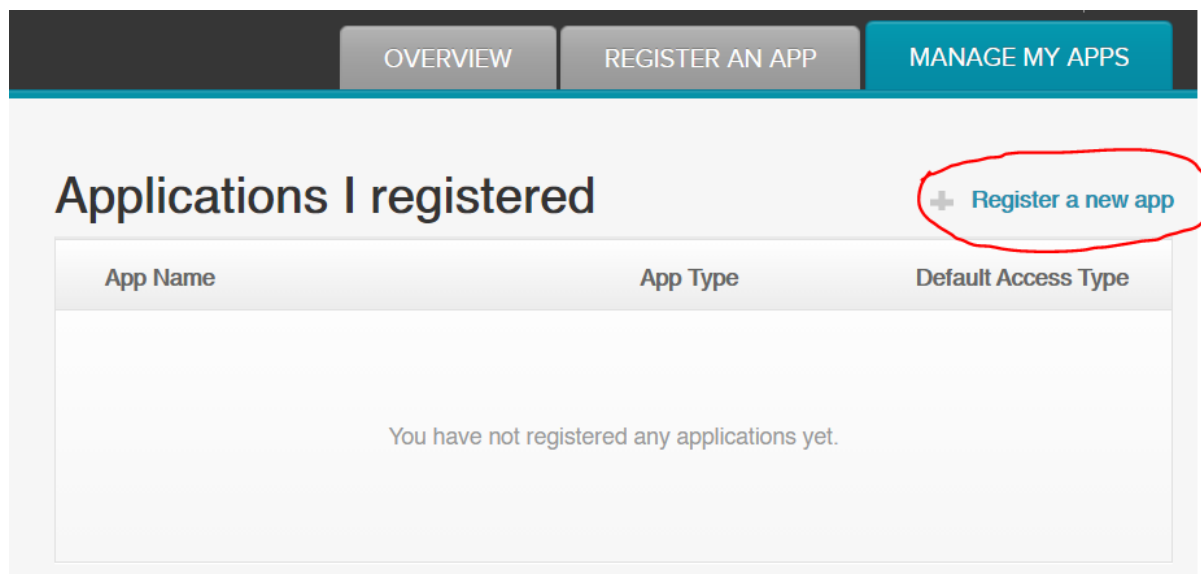
How to get the OAuth Token for the Fitbit API

Here I have used the implicit code flow for my project.

Steps:

1. Go to <https://dev.fitbit.com/apps> and Login

2. Register a new app



3. Fill up the fields

Register an application

Application Name *

HR Monitor

Description *

HR Monitor



Application Website URL *

https://github.com/arpanghosh8453



Organization *

Arpan Ghosh

Organization Website URL *

https://github.com/arpanghosh8453

Terms of Service URL *

https://github.com/arpanghosh8453

Privacy Policy URL *

https://github.com/arpanghosh8453

OAuth 2.0 Application Type *

☐ Server

☐ Client

☒ Personal



Redirect URL *

http://localhost:8888



Default Access Type *

☐ Read & Write

☒ Read Only



[+ Add a subscriber](#)

☒ I have read and agree to the [terms of service](#)

Register

Cancel

4. Click on the OAuth tutorial page

Applications I registered

[+ Register a new app](#)

Application HR Monitor

HR Monitor

[Edit Application Settings](#)[Delete Application](#)

[Reset Client Secret](#)[Revoke Client Access Tokens](#)

OAuth 2.0 Client ID
[REDACTED]

Client Secret
[REDACTED]

Redirect URL
http://localhost:8888

OAuth 2.0: Authorization URI
https://www.fitbit.com/oauth2/authorize

OAuth 2.0: Access/Refresh Token Request URI
https://api.fitbit.com/oauth2/token

[OAuth 2.0 tutorial page](#)

Subscribers endpoint stats.

Subscriber ID	Verified at / Verification code	Stats
---------------	---------------------------------	-------

[← View all applications](#)

5. Choose the Implicit Grant flow and Click on the link below there

OAuth 2.0 tutorial page

For a detailed explanation of OAuth 2.0, see the Fitbit API [documentation](#).

1: Authorize

- First, choose the type of flow your application will use. Implicit grant flow is for use in client-side applications that cannot keep a secret because they distribute their source code to the client (web apps, mobile apps). The authorization code flow is for server-side applications that can keep a secret. If possible, use the authorization code flow, because while both flows are secure, it provides additional security.

Flow type ☒ Implicit Grant Flow ☐ Authorization Code Flow

- Enter all of your application's relevant data below. You can find this data at [dev.fitbit.com](#).

Fitbit URL:
Fitbit API URL:
OAuth 2.0 Client ID:
Client Secret:
Redirect URI:

- Choose below what user data you'd like to have access to.

Select Scopes

☒ activity ☒ heartrate ☒ location ☒ nutrition
☒ profile ☒ settings ☒ sleep ☒ social
☒ weight

- The default expiration times are 1 hour for the authorization code flow, and 1 day for the implicit grant flow. The expiration time for the implicit grant flow can be set to certain values; see [the docs](#) for details.

Expires In(ms):

- We've generated the authorization URL for you, all you need to do is just click on link below:

[https://www.fitbit.com/oauth2/authorize?response_type=token&client_id=\[REDACTED\]&redirect_uri=http%3A%2F%2Flocalhost%3A8888&scope=activity%20heartrate%20location%20nutrition%20profile%20settings%20sleep%20social%20weight&expires_in=604800](https://www.fitbit.com/oauth2/authorize?response_type=token&client_id=[REDACTED]&redirect_uri=http%3A%2F%2Flocalhost%3A8888&scope=activity%20heartrate%20location%20nutrition%20profile%20settings%20sleep%20social%20weight&expires_in=604800)

6. Grant the Access from the page opened.



HR Monitor by [Arpan Ghosh](#) would like the ability to access the following data in your Fitbit account for 1 week

Warning! This app is not using HTTPS to securely obtain your permission.

- ☒ Allow All
 - ☒ activity and exercise
 - ☒ Fitbit devices and settings
 - ☒ sleep
 - ☒ friends ⓘ
 - ☒ food and water logs ⓘ
 - ☒ weight ⓘ
 - ☒ profile ⓘ
 - ☒ heart rate
 - ☒ location and GPS

If you allow only some of this data, HR Monitor may not function as intended. Learn more about these permissions [here](#).

Deny

Allow

Data shared with HR Monitor will be governed by Arpan Ghosh's privacy policy and terms of service. You can revoke this consent at any time in your Fitbit [Privacy Policy](#). More information about these permissions can be found [Terms of Service](#).

7. Click on allow and it will say the page cannot be reached. That's fine. Just see the change in url and copy everything from the resulting url (after you click allow) after the # symbol and put them in the next field (Parse response) of the page.

2: Parse response

After the user consents and clicks the "allow" button, copy and paste the ending part of the URL, starting from the #. For instance for url:
`https://localhost/#_scope=nutrition&user_id=28GVHZ&token_type=Bearer&expires_in=593433&access_token=blablaToken` copy and paste
`#scope=nutrition&user_id=28GVHZ&token_type=Bearer&expires_in=593433&access_token=blablaToken` in input field below

3 Make Request

Finally, when you have an access token, you can start making requests. If you had a token before, you don't need to go through steps 2-3, just paste your token below and make sure you enter your app data in step 1. We only support GET requests at the moment in this tutorial. But please feel free to check out other types of requests in the docs too on your own.

OAuth 2.0 Access Token:

API endpoint URL:

8. It will show you the granted Auth Token there and you can replace the token part of the code with the granted token shown here. I am not including a screenshot of the page due to security reasons.