



How to show Now Playing in Spotify with Next.js

Written on January 25, 2021 by Theodorus Clarence.

Last updated November 12, 2021. [⌚ See changes](#)

[⌚ 5 min read](#) [⌚ 12 728 views](#)

[Read in Bahasa Indonesia](#)

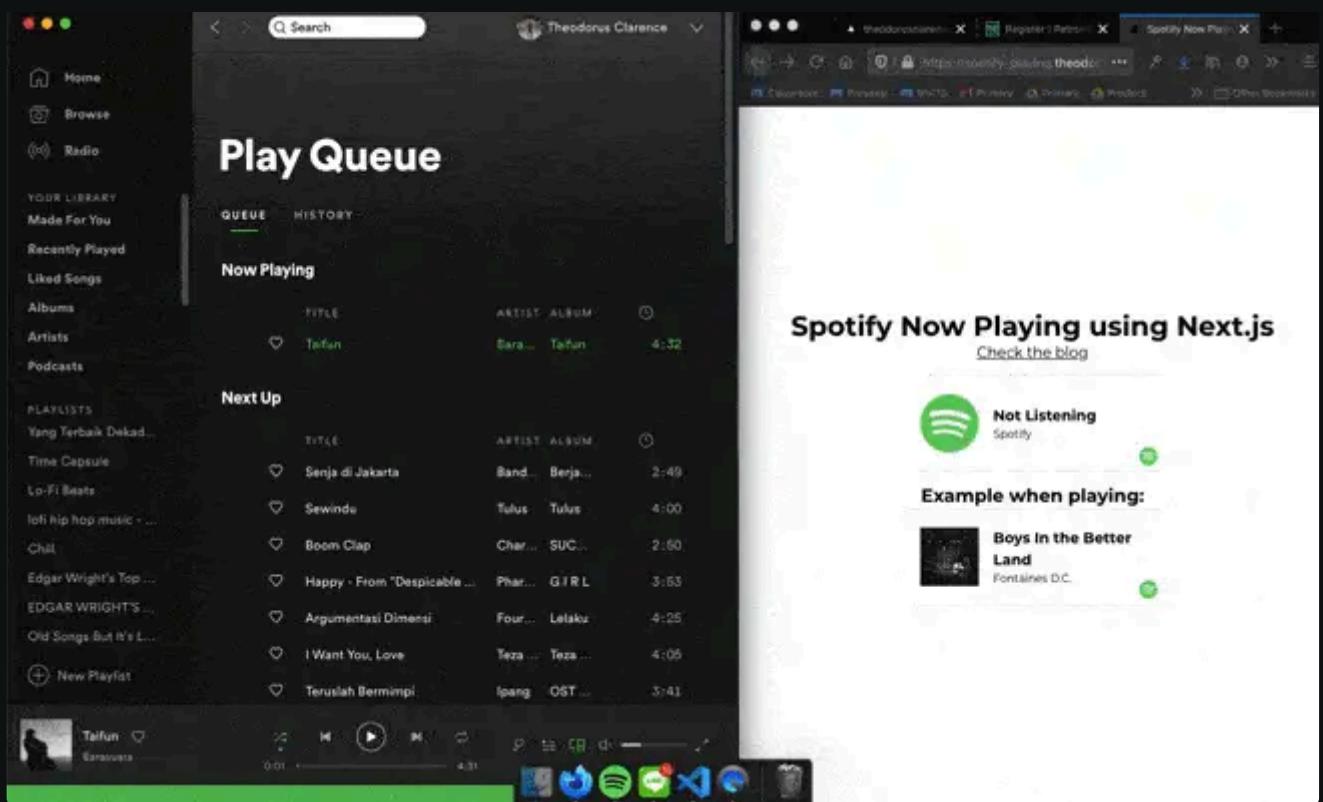
Introduction

On this website, I use Spotify API to show what song is currently playing. Now, I will try to explain on how do I do that.

Next.js has a built-in API in a single application, so choosing Next.js is a perfect choice so we don't expose our API keys.

[Skip to main content](#)





Website Demo

1. Create an Application on Spotify Developer

- Go to [Spotify Developer](#) Website.
- Click **Create An App**
- Fill in App Name, App description, then **Create**

Dashboard



CREATE AN APP

App name
Belajar Spotify API

App description
[Untuk Belajar API Spotify](#)

I understand that I will need [permission from Spotify](#) before using this app in a commercial project

I understand and agree with Spotify's [Developer Terms of Service](#) and [Branding Guidelines](#)

[CANCEL](#) [CREATE](#)

- Next, you will be provided with `Client ID` and `Client Secret` (Don't give this secret to anyone)

[← BACK TO DASHBOARD](#)

Belajar Spotify API

Untuk Belajar API Spotify

Client ID eaccb97f6d0e405897adf1dd80b95c01

Client Secret a4195c2d0a4243609e697e602e70b762 [RESET](#)

[HIDE CLIENT SECRET](#)

- Open edit settings, then fill out `http://localhost:3000` on RedirectURIs

The screenshot shows the Spotify for Developers dashboard. In the center, a white modal window titled "EDIT SETTINGS" is open. Inside the modal, there are several input fields and sections:

- Application name:** Belajar Spotify API
- Application description:** Untuk Belajar API Spotify
- Website:** Add a website (with placeholder text: Where the user may obtain more information about this application (e.g. http://mysite.com).)
- Redirect URIs:** http://localhost:3000 Remove
- Additional Redirect URI:** https://example.com/callback/ (with an "ADD" button next to it) (with placeholder text: White-listed addresses to redirect to after authentication success OR failure (e.g. http://mysite.com/callback/))

At the top right of the modal, there are "EDIT SETTINGS" and "LOGOUT" buttons. The background of the dashboard is dark, showing a chart for "Users This Month" with a value of 0.

The first step is done! Now, you need to do authentication to get access & refresh token

2. Authenticate Your Account

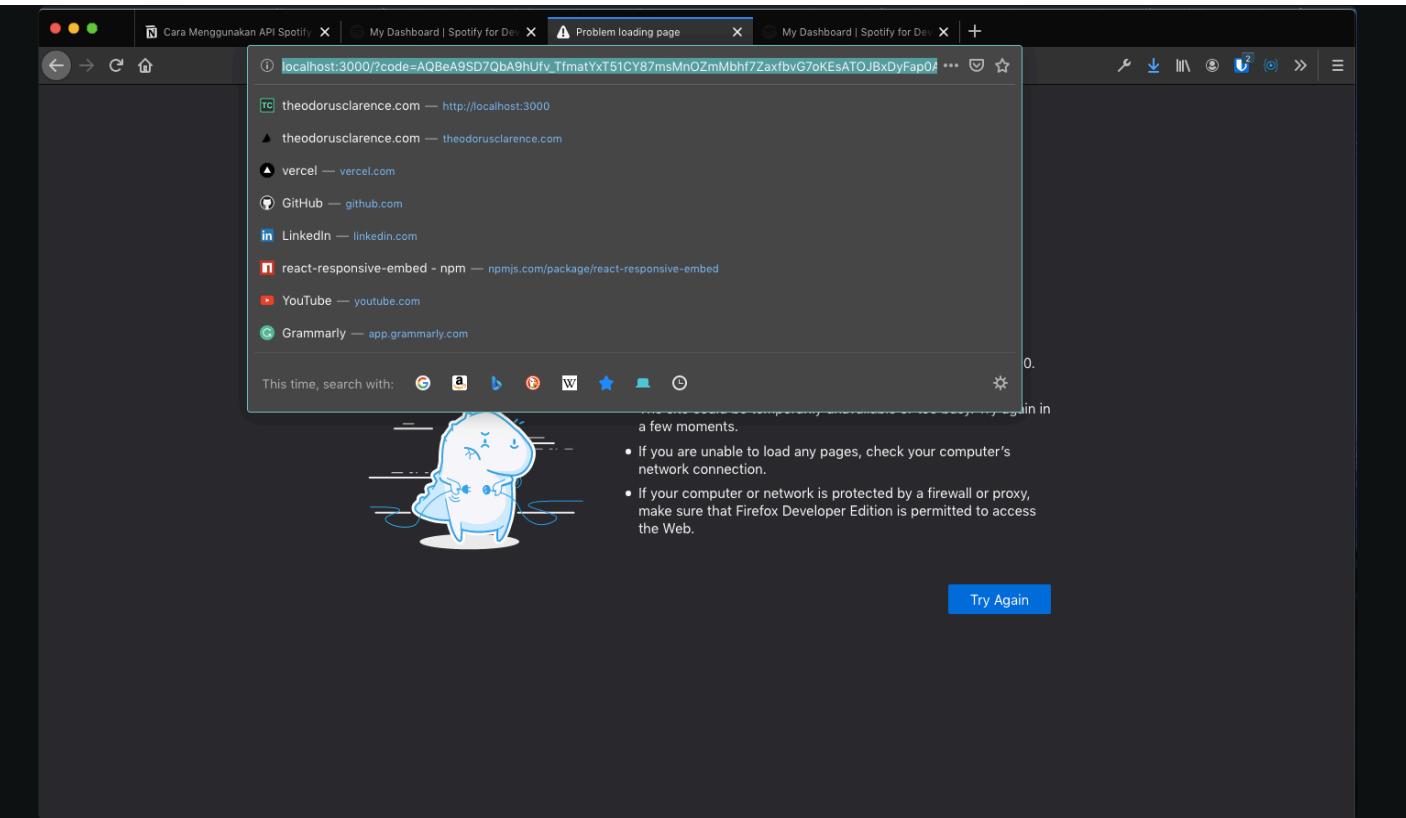
To do authentication, we need to prepare `CLIENT_ID` and put it into this link below:

```
https://accounts.spotify.com/authorize?client_id=CLIENT_ID_HERE&response_type=code&redirect_uri=http://localhost:3000&scope=user-read-currently-playing
```

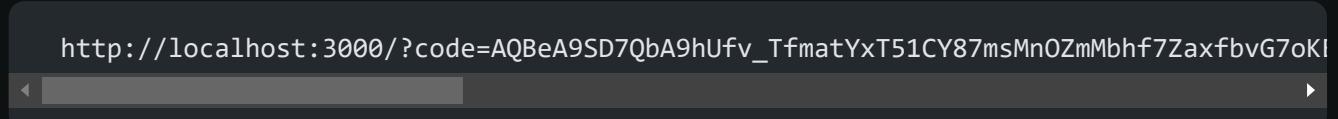
example:

```
https://accounts.spotify.com/authorize?client_id=eaccb97f6d0e405897adf1dd80b95c01&response_type=code&redirect_uri=http://localhost:3000&scope=user-read-currently-playing
```

Open that link in a browser, then you will get redirected into a uri, copy the website link



Example of the redirect link:



`code` that you have is the one after (=). Don't forget to write it down.

Next we need to get authorization client that is already encrypted with base64, use this [website](#) to encrypt with the format of `client_id:client_secret`

For example:

Encode to Base64 format

Simply enter your data then push the encode button.

```
eaccb97f6d0e405897adf1dd80b95c01:a4195c2d0a4243609e697e602e70b762
```

 To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.

Destination character set.

Destination newline separator.

Encode each line separately (useful for when you have multiple entries).

Split lines into 76 character wide chunks (useful for MIME).

Perform URL-safe encoding (uses Base64URL format).

 Live mode OFF Encodes in real-time as you type or paste (supports only the UTF-8 character set).

> ENCODE < Encodes your data into the area below.

```
ZWFjY2I5N2Y2ZDB1NDA1ODk3YWVmMWRkODBiOTVjMDE6YTQxOTVjMmQwYTQyNDM2MDIINjk3ZTYwMmU3MGI3NjI=
```

Next, write down the encrypted `base64`

After that, open up terminal/cmd, and run this command, don't forget to change the `base64` and the `code` to yours.

```
curl -H "Authorization: Basic CHANGE_BASE64_HERE"
-d grant_type=authorization_code -d code=CHANGE_CODE_HERE -d redirect_uri=http%3A
%2F%2Flocalhost:3000 https://accounts.spotify.com/api/token
```

*make sure the command is in one line

Example:

```
curl -H "Authorization: Basic ZWFjY2I5N2Y2ZDB1NDA1ODk3YWVmMWRkODBiOTVjMDE6YTQxOTVjI
```

You will get a JSON like this:

```
{  
  "access_token": "BQDKx07h1I1wA3esGK9zCFWn97XORJEPjwAHAEIxCnDXcmy9GbEuPacquwWvpiM4NvkvGMq-3bdA5crjCCKN4eWB6SErv-IkefQL0FZTJ6_KvV5nadyGeHjCx_cyDG7j8PdPTUfbNjLB5FILEl06dQzJm2go6djgvtteGet7tUW8 -d redirect_uri=http://localhost:3000 https://accounts.spotify.com/api/token  
  "token_type": "Bearer",  
  "expires_in": 3600,  
  "refresh_token": "AQAtxXvnzRTt4c2-2_Av2WyJQKwXUW_hMVN6QNiqv2i8A2ZE1Varmvvhqyc8Pf-Z5n827FTFxTpHq5E3k0srlRW3TuJWxjVQsw0icR0zo3BXRFt2FB2Qfj-pFaZwY-qc8",  
  "scope": "user-read-currently-playing"  
}
```

Example on the terminal:

```
Clarence — Clarence@Clarences-MacBook-Air — ~ — zsh — 128x33  
  
~ curl -H "Authorization: Basic ZWFjY2I5N2Y2ZDBlNDA1ODk3YWVmMWRk0DBi0TVjMDE6YTQx0TVjMmQwYTQyNDM2MDlLNjk3ZTYwMmU3MGI3NjI=" -d grant_type=authorization_code -d code=AQB39qMjySfUaLYByxD0kMcZH0FQkvLSDWmBcrU-mG_qglg3uE8UpnYRvvR36R2UIZy9fa8J1Ho1NoAC58SRe-Nan4NVkvGMq-3bdA5crjCCKN4eWB6SErv-IkefQL0FZTJ6_KvV5nadyGeHjCx_cyDG7j8PdPTUfbNjLB5FILEl06dQzJm2go6djgvtteGet7tUW8 -d redirect_uri=http://localhost:3000 https://accounts.spotify.com/api/token  
{"access_token":"BQDKx07h1I1wA3esGK9zCFWn97XORJEPjwAHAEIxCnDXcmy9GbEuPacquwWvpiM4d33gJHVOP9KUxY8AXkpXc-_zRFZBfneHM2vEeV1Fbfr-0Mw94oimlNf77dRiyxPpm4IUVNlloUwgYcfkA00","token_type":"Bearer","expires_in":3600,"refresh_token":"AQAtxXvnzRTt4c2-2_Av2WyJQKwXUW_hMVN6QNiqv2i8A2ZE1Varmvvhqyc8Pf-Z5n827FTFxTpHq5E3k0srlRW3TuJWxjVQsw0icR0zo3BXRFt2FB2Qfj-pFaZwY-qc8","scope":"user-read-currently-playing"}  
~
```

what we need to write down is the `refresh_token`. That token will last infinitely

Now you can do the request with Next.js or another backend application

3. Creating API routes in Next.js

If you need a reference to create the app, you can [check my repository](#). Don't forget to star it!

You can also use my [nextjs-tailwind-starter](#) by using:

```
npx create-next-app -e https://github.com/theodorusclarence/nextjs-tailwind-starter
```

After all app is installed, add querystring dependency

```
yarn add querystring
```

or

```
npm i querystring
```

Next, make an api route in /pages/api/spotify.js

If you want the TypeScript version, checkout the code on [GitHub](#)

```
// /pages/api/spotify.js
import queryString from 'querystring';

const {
  SPOTIFY_CLIENT_ID: client_id,
  SPOTIFY_CLIENT_SECRET: client_secret,
  SPOTIFY_REFRESH_TOKEN: refresh_token,
} = process.env;

const basic = Buffer.from(`${client_id}:${client_secret}`).toString('base64');
const NOW_PLAYING_ENDPOINT = `https://api.spotify.com/v1/me/player/currently-playing`;
const TOKEN_ENDPOINT = `https://accounts.spotify.com/api/token`;

const getAccessToken = async () => {
  const response = await fetch(TOKEN_ENDPOINT, {
    method: 'POST',
    headers: {
      Authorization: `Basic ${basic}`,
      'Content-Type': 'application/x-www-form-urlencoded',
    },
    body: queryString.stringify({
      grant_type: 'refresh_token',
      refresh_token,
    })
  });
  const json = await response.json();
  return json.access_token;
}

export default getAccessToken;
```

```
    }),

  });

  return response.json();
};

export const getNowPlaying = async () => {
  const { access_token } = await getAccessToken();

  return fetch(NOW_PLAYING_ENDPOINT, {
    headers: {
      Authorization: `Bearer ${access_token}`,
    },
  });
};

export default async (_, res) => {
  const response = await getNowPlaying();

  if (
    response.status === 204 ||
    response.status > 400 ||
    response.data.currently_playing_type !== 'track'
  ) {
    return res.status(200).json({ isPlaying: false });
  }

  const data = {
    isPlaying: response.data.is_playing,
    title: response.data.item.name,
    album: response.data.item.album.name,
    artist: response.data.item.album.artists
      .map((artist) => artist.name)
      .join(', '),
    albumImageUrl: response.data.item.album.images[0].url,
    songUrl: response.data.item.external_urls.spotify,
  };
}

res.status(200).json(data);
`
```

Add .env.local with authorization data that we have written (change the data below with yours)

```
SPOTIFY_CLIENT_ID=eaccb97f6d0e405897adf1dd80b95c01
SPOTIFY_CLIENT_SECRET=a4195c2d0a4243609e697e602e70b7
SPOTIFY_REFRESH_TOKEN=AQAtxXvnzRTt4c2-2_Av2WyJQKWxUW_hMVN6QNiqv2i8A2ZE1Varmvdhqyc8F
```

Your API is ready with route: `GET <https://localhost:3000/api/spotify>`

4. Use API with Next.js

For data fetching, we can use [SWR](#). SWR is a great library that can fetch the API periodically. SWR will do refetching each time we refocus a the window. Also, install react-icons to get the spotify logo.

```
yarn add swr react-icons
```

or

```
npm i swr react-icons
```

Add SWR in pages/index.jsx like this:

TypeScript Version

```
import useSWR from 'swr';

export default function Home() {
  const fetcher = (url) => fetch(url).then((r) => r.json());
  const { data } = useSWR('/api/spotify', fetcher);
  return (
    <>
    <section className='bg-gray-600'>
      <main className='flex items-center justify-center'>
        {console.log(data)}
    </main>
  </section>
)
```

```
        </main>
    </section>
</>
);
}
```

There will be 2 types of JSON data that will be sent by the Spotify API:

1. When there is no song playing

```
{
  "isPlaying": false
}
```

2. When there is a song playing

```
{
  "album": "Menari Dengan Bayangan",
  "albumImageUrl": "https://i.scdn.co/image/ab67616d0000b273d62368848886590605",
  "artist": "Hindia",
  "isPlaying": true,
  "songUrl": "https://open.spotify.com/track/080PqLv99g8avzmxQepmiw",
  "title": "Besok Mungkin Kita Sampai"
}
```

So, we can conditionally render the data like this:

```
<a
  target='_blank'
  rel='noopener noreferrer'
  href={
    data?.isPlaying
      ? data.songUrl
      : 'https://open.spotify.com/user/erence21?si=yTsrZT5JSH0p7tn3ist7Ig'
  }
  className='relative flex w-72 items-center space-x-4 rounded-md border p-5 tra
>
<div className='w-16'>
  {data?.isPlaying ? (
    <img alt='Song icon' />
  ) : (
    <img alt='No song icon' />
  )}
</div>
<div className='flex-grow pl-4'>
  {data?.isPlaying ? (
    <div>
      {data.title} - {data.artist}
    </div>
  ) : (
    <div>
      {data.album}
    </div>
  )}
</div>
<div className='flex-shrink'>
  {data?.isPlaying ? (
    <button>Stop</button>
  ) : (
    <button>Play</button>
  )}
</div>
</div>
</a>
```

```
<img  
    className='w-16 shadow-sm'  
    src={data?.albumImageUrl}  
    alt={data?.album}  
/>  
) : (  
    <SiSpotify size={64} color={'#1ED760'} />  
)  
</div>  
  
<div className='flex-1'>  
    <p className='component font-bold'>  
        {data?.isPlaying ? data.title : 'Not Listening' }  
    </p>  
    <p className='font-dark text-xs'>  
        {data?.isPlaying ? data.artist : 'Spotify' }  
    </p>  
</div>  
<div className='absolute bottom-1.5 right-1.5'>  
    <SiSpotify size={20} color={'#1ED760'} />  
</div>  
.
```

You can check out the live deployment on spotify-playing.theodorusclarence.com

Thanks and Goodluck!



 Tweet this article



haafidz-jp Dec 28, 2021

thank you tutor nya ka!

↑ 1



0 replies



fiqryq Jan 4, 2022

mantap maass tengkyu

↑ 1



0 replies



tszhong0411 Jan 29, 2022

Thank you! ❤️

↑ 1



0 replies



khandelwaldev Dec 10, 2023

Thank You 😊

↑ 1



0 replies

Write

Preview

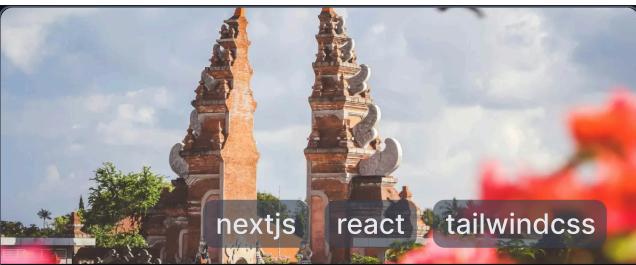
Aa

Sign in to comment

M↓

Sign in with GitHub

Other posts that you might like



nextjs react tailwindcss

How to Create Fully Reusable React Components

⌚ 9 min read ⚽ 4 582 views

April 21, 2024

Creating a component is fairly easy, but doing them correctly is a different story.



nextjs react

Advanced React Patterns

⌚ 10 min read ⚽ 1 425 views

March 03, 2024

List of react advanced patterns complete with examples.

Enjoying this post?

Don't miss out 😊. Get an email whenever I post, no spam.

[Subscribe Now](#)

[Edit this on GitHub](#)

[← Back to blog](#)

[Source Code](#)

[Design](#)

[Docs](#)

[Book Notes](#)

[Starter Templates](#)

[Analytics](#)

[Statistics](#)

[Guestbook](#)

[Subscribe](#)

[RSS](#)

Reach me out



© Theodorus Clarence 2024 • Got any feedback?