

- 1. Reference
 - [1.1. Next.js Authentication Made Easy with Microsoft Entra ID](#)
 - [1.2. Next.js 16 Middleware DEPRECATED - Authentication In Proxy Or Data Access Layer?](#)
 - [1.3. Overview of user consent and how to manage it in Microsoft Entra | Microsoft](#)
 - [1.4. Add Tailwind CSS to an Existing Next js Project](#)
 - [1.5. How to style Material UI component with Tailwindcss in React project](#)
- 2. Project Structure
- 3. App Registrations
 - [3.1. Permissions Needed](#)
- 4. Steps
 - [4.1. create nextjs app](#)
 - [4.2. Create .env file](#)
 - [4.3. Configure Git](#)
 - [4.4. Test nextjs](#)
 - [4.5. Check Authentication](#)
 - [4.6. Install NextAuth.js v5](#)

1. Reference

1.1. Next.js Authentication Made Easy with Microsoft Entra ID

[Next.js Authentication Made Easy with Microsoft Entra ID](#)

1.2. Next.js 16 Middleware DEPRECATED - Authentication In Proxy Or Data Access Layer?

[Next.js 16 Middleware DEPRECATED - Authentication In Proxy Or Data Access Layer?](#)

1.3. Overview of user consent and how to manage it in Microsoft Entra | Microsoft

[Overview of user consent and how to manage it in Microsoft Entra | Microsoft](#)

1.4. Add Tailwind CSS to an Existing Next js Project

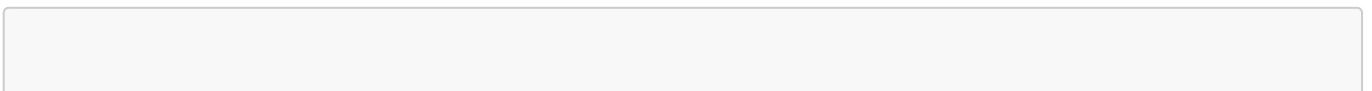
[Add Tailwind CSS to an Existing Next js Project](#)

1.5. How to style Material UI component with Tailwindcss in React project

[How to style Material UI component with Tailwindcss in React project](#)

2. Project Structure

The following is the desire structure



```
web-calibre/
├── src/
│   ├── app/
│   │   ├── api/
│   │   │   ├── auth/
│   │   │   └── upload/
│   │   ├── dashboard/
│   │   ├── layout.tsx
│   │   └── page.tsx
│   ├── components/
│   ├── lib/
│   │   ├── prisma.ts
│   │   └── theme.ts
│   └── middleware.ts
├── prisma/
└── .env
```

All Routes (Pages & APIs)

API ROUTES

NextAuth routes [...nextauth]

Custom API for OneDrive sessions

Protected user area

Global Layout

Landing Page

MUI Components (FileTable, Navbar)

Config (Prisma, Theme, Auth)

<-- MUI Theme Defined Here

<-- Protected Routes Logic

DB Schema

Secrets

3. App Registrations

WebCalibre2 Application (client) ID = 'b549931a-f491-436b-b7bc-d37d8ca3c17e'

Object ID ='0b8256a2-e2ac-472b-896a-4b5845bc32fe'

Directory (tenant) ID = '1c06ce7c-7884-4796-8652-d4c32d75a5d0'

Expires = 1/3/2028

Value='O3p8Q~oMph-0kSLwkvezEJzJdx_iHGOjJjrr5Pa1A'

Secret ID='6a242be1-c711-4cc3-a132-03c1f57993ed'

Microsoft Azure

Search resources, services, and docs (G+/)

Copilot

slohning@lin

DEFAULT DIRECTORY

Home > App registrations > WebCalibre2

WebCalibre2 | Certificates & secrets

Search

Got feedback?

Overview

Quickstart

Integration assistant

Diagnose and solve problems

Manage

Branding & properties

Authentication (Preview)

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles

Owners

Roles and administrators

Manifest

Support + Troubleshooting

Got a second to give us some feedback? →

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates (0) Client secrets (1) Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value	Secret ID	Copied
WebCalibre2	1/3/2028	O3p8Q~oMph-0kSLwkvezEJzJdx_iHGOjJjrr...	6a242be1-c711-4cc3-a132-03c1f57993ed	

Since you are using NextAuth.js, the URL must follow a very specific pattern.

For local development: `http://localhost:3000/api/auth/callback/azure-ad`

For production: `https://your-domain.com/api/auth/callback/azure-ad`

Note: Replace `azure-ad` with whatever ID you give the provider in your code. By default, in NextAuth, it is usually `azure-ad`.

WebCalibre3

Display name	WebCalibre3
Application (client) ID	a95ef644-cb9d-465c-8dae-f822a78a8ac3
Object ID	1e1f2bda-661c-4ad0-9aae-82313c603076
Directory (tenant) ID.	1c06ce7c-7884-4796-8652-d4c32d75a5d0

Client credentials

Description	Expires	Value	Secret ID
WebCalibre3	1/5/2028	VZ48Q~EOsgZcgBI25cc0zys.h9zM7hN9I7DhZdnW	d647cf28-80ed-45fb-a89e-e2bc323a40e9

3.1. Permissions Needed

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission

✓ Grant admin consent for Default Directory

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (3)				...
Files.ReadWrite	Delegated	Have full access to user files	No	...
offline_access	Delegated	Maintain access to data you have given it access to	No	✓ Granted for Default Dire... ...
User.Read	Delegated	Sign in and read user profile	No	✓ Granted for Default Dire... ...

Other permissions granted for Default Directory

4. Steps

4.1. create nextjs app

The following command creates the nextjs folder structure

Yes, Next.js has a specific syntax to create a project in your current directory instead of creating a new subfolder.

To do this, navigate to your empty project folder in the terminal and run:

```
npx create-next-app@latest .
```

Important Details:

1. The Dot (.): This tells the installer to use the current directory as the project root.
2. Naming Conflict: If the folder name contains capital letters (e.g., WebCalibre), the command will fail with an error. Next.js requires the root folder name to be all lowercase and URL-friendly. You must rename your folder to web-calibre before running the command. This caused me rename to all lower case.
3. Empty Folder: The directory must be empty (except for hidden files like .git). If you already have files in there, the installer will stop to prevent overwriting your work.

What happens next? Even though you are using the current directory, the interactive prompt will still ask you:

"What is your project named?" You can just hit Enter to accept the current folder name.

Configuration: It will then ask you about TypeScript, ESLint, Tailwind CSS, the src/ directory, and the App Router.

Based on our previous discussions, I recommend selecting "Yes" for TypeScript, ESLint, Tailwind, the src/ directory, and the App Router to match the professional structure we planned for WebCalibre.

4.2. Create .env file

In Next.js, the distinction between .env and .env.local is primarily about security and **overriding defaults**. While both store key-value pairs, they are handled differently by Git and the Next.js runtime.

1. Security & Version Control (Git) The most important reason is that .env.local is automatically ignored by Git in standard Next.js projects.
 - .env: Intended for "safe" default values that you want to share with your entire team. It is committed to your repository so that every developer has the same baseline configuration.
 - .env.local: Intended for secrets (API keys, database passwords, Azure secrets). Because it is not committed to Git, your private credentials never end up on GitHub or in your production code history.
2. Priority & Overriding Next.js uses a "hierarchy" of loading.¹ If you define the same variable in both files, .env.local wins. This allows you to set a generic placeholder in .env for the team, but use your specific local developer settings on your own machine. For example:
 - In .env: DATABASE_URL=postgres://localhost:5432/mydb (The team's default)
 - In .env.local: DATABASE_URL=postgres://admin:password123@localhost:5432/my_private_dev_db (Your personal setup)

The Next.js Loading Order Next.js looks for variables in this specific order (from highest priority to lowest):

1. process.env (System environment variables)

2. .env.development.local (Only during npm run dev)
3. .env.local (Available in all environments except test)
4. .env.development
5. .env (The final fallback)

```
DATABASE_URL="postgresql://stephen:Web2025$$$@192.168.1.210:5432/webcalibre2"
```

```
# Credentials for your Microsoft App (NextAuth)
AZURE_AD_CLIENT_ID="549931a-f491-436b-b7bc-d37d8ca3c17e"
AZURE_AD_CLIENT_SECRET="6a242be1-c711-4cc3-a132-03c1f57993ed"
AZURE_AD_TENANT_ID="1c06ce7c-7884-4796-8652-d4c32d75a5d0"
```

Summary Comparison Table

Feature	.env	.env.local
Purpose	Shared defaults/constants	Private secrets & local overrides
Committed to Git?	Yes	No (Added to .gitignore)
Sensitive Data?	NEVER	YES (API Keys, Secrets)
Scope	All developers/environments	Just your machine

****Best Practice Tip: **** Since .env.local isn't shared on GitHub, it's a good idea to create a file named .env.example in your project. This file should contain the names of the keys (e.g., AZURE_AD_CLIENT_SECRET=) but leave the values blank, so other developers know which variables they need to create on their own machines.

4.3. Configure Git

Nextjs initializes git by default. But it doesn't add some configs that I do

```
git config --global push.followTags true
```

```
git remote add origin "/Users/stephenlohnning/Library/CloudStorage/OneDrive-Personal/Documents/10_GIT_Repositories/124_WebCalibre2.git"
```

```
git remote -v origin /Users/stephenlohnning/Library/CloudStorage/OneDrive-Personal/Documents/10_GIT_Repositories/124_WebCalibre2.git (fetch) origin
/Users/stephenlohnning/Library/CloudStorage/OneDrive-Personal/Documents/10_GIT_Repositories/124_WebCalibre2.git (push)
```

4.4. Test nextjs

```
npm run dev

> 124_webcalibre2@0.1.0 dev
```

```
> next dev

▲ Next.js 16.1.1 (Turbopack)
- Local:      http://localhost:3000
- Network:    http://192.168.1.100:3000
- Environments: .env

✓ Starting...
✓ Ready in 648ms
```

If it working proceed

4.5. Check Authentication

I ran through the Video:-

[Next.js Authentication Made Easy with Microsoft Entra ID](#)

4.6. Install NextAuth.js v5

`npm install next-auth@beta`

Setup Environment The only environment variable that is mandatory is the `AUTH_SECRET`. This is a random value used by the library to encrypt tokens and email verification hashes. (See Deployment to learn more). You can generate one via the official Auth.js CLI running:

`npx auth secret`

created `.env.local`

```
AUTH_SECRET="7cz3Z4kUI2kB3mdAPo58iioUDLSRJ92X8+boEixv08k=" # Added by `npx
auth`. Read more: https://cli.authjs.dev
```