**Next + Okta+GoogleCalendarApi**

# Next+Okta

This instruction is for people who want to build an authentication program based on next and okta. Please follow the steps below.

1. Register an account on the website: <https://www.okta.com/free-trial/>.

A screenshot of a computer

Description automatically generated

Note: you probably notice that we need to use work email here. Please don’t choose “Customer Identity Cloud” option since it will use auth0 instead of okta to build the authentication process. Okta purchased auth0 in 2021, which means the two produces belong to the same company. However, we plan to use Okta in this demo.

* 1. If you don’t want to use work email, the replacement is as follows:

1. Download "okta for Windows" from <https://github.com/okta/okta-cli/releases>.
2. Unzip the downloaded file to a directory. For instance, F:\nextoktademo.
3. Use the cmd and enter the directory, then execute the command:

Okta register

A screenshot of a computer

Description automatically generated

1. Follow the instructions to create an account.
2. Login on the website: https://developer.okta.com/login/
3. Login the website. When you first log in, the login page will be like the below:

A screenshot of a computer

Description automatically generated

* 1. If you don’t login for the first time, you need to enter the organization URL, the first page looks like the below:

A screenshot of a computer

Description automatically generated

Note: You can get the url after you register the account. It’s the domain name. On my account, it’s dev-85274789.

1. Create an application:

A screenshot of a computer

Description automatically generated

1. Choose “OIDC” and “Web Application”

A screenshot of a application

Description automatically generated

1. Fill out the necessary fields:

A screenshot of a computer

Description automatically generated

Note: The number 3 is key. If you use another port such as 3001, please modify it accordingly. I am unsure the number 4 is perfect and need to test it later.

1. Check the “Allow everyone in your organization to access” for simplicity.

A screenshot of a computer

Description automatically generated

Note: we can adjust it accordingly.

1. Add a user for the test.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Note: When you add a user and set up the password, you will find the password expires. This means you must reset the password when you use this user to log in.

1. Create an api token just in case you want to use the command line.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Note down the token, please.

**For now, we finished the okta settings on the server.**

Next, let’s set up the demo project(Check step 1 below and start from step 2 for simplicity).

1. We can still use the okta command to create a demo project.

A screenshot of a computer

Description automatically generated

Note: You need a token to log in. The token is what we created in step 8 above. In addition, the Okta command doesn’t support the nextJs project. There is an outdated nextjs project on GitHub:

<https://github.com/oktadev/okta-nextjs-typescript-example>

We can’t use it directly and need to modify it accordingly.

1. Please download the nextJs project on GitHub(I modified the project mentioned on step 1 above), then open the project with VSCODE editor:

<https://github.com/guardiangel/okta-nextjs-typescript-example>

A screen shot of a computer

Description automatically generated

1. Add a .env.local file on the root directory of the project and add content like below:

A screenshot of a computer

Description automatically generated

Note: we can get the client id and secret from the app we created in step 3 above.

A screenshot of a computer

Description automatically generated

1. Install the dependencies.

A screenshot of a computer

Description automatically generated

1. Start the project and click the “Log in” button

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a movie

Description automatically generated

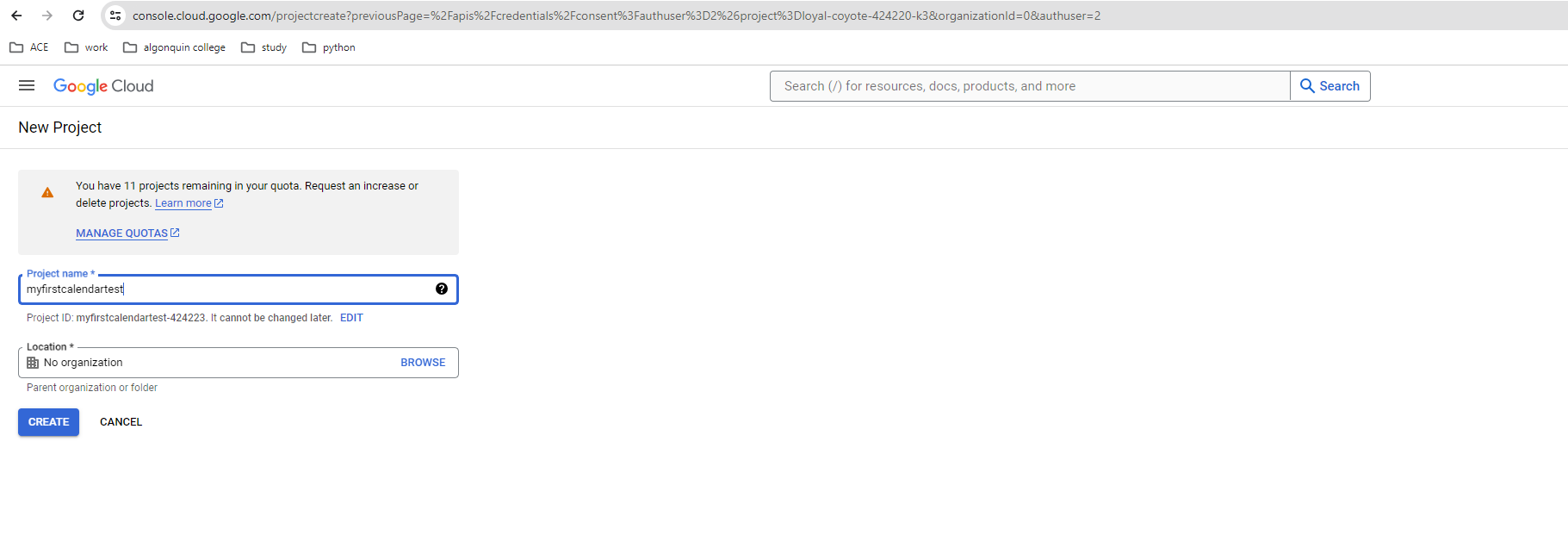
You can check the login email and token on the url: <http://localhost:3000/loginbtn>

A screenshot of a computer

Description automatically generated

# GoogleCalendarApi

1. Open <https://console.cloud.google.com/>, and create a project.



2.Enable “Google Calendar Api”

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

3. OAuth consent screen, create necessary info.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Note: I have already choose google calendar api in another app, so it won’t display here. This option is only for demo.**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Testing.This status indicates the current app is in testing phase and only allows 100 users.

    if we want to publish app, click the “PUBLISH APP” button and get the below information:

    Push to production?

    Your app will be available to any user with a Google Account.

    You've configured your app in a way that requires verification. To complete verification, you will need to provide:

    An official link to your app's Privacy Policy.

    A YouTube video showing how you plan to use the Google user data you get from scopes.

    A written explanation telling Google why you need access to sensitive and/or restricted user data.

    All your domains verified in Google Search Console.

4. Create credentials

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Note: Note down the Client ID and Client secret for later use.**

Since the app is in testing status, only the users added here can test in client app. If the app is published, all the users that has a google account can have access to the app.

 5. Test:

5.1 Download the demo project from :

<https://github.com/guardiangel/okta-nextjs-typescript-example>

5.2 Add .env.local file and add content:



5.3 http://localhost:3000/googlecalendar and click "Sign in with Google".

A screenshot of a computer

Description automatically generated

A computer screen shot of a card

Description automatically generated

5.4 Enter the created email in step 3 above (Probably will choose the email if logged in before),

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

    5.3 If no errors happens, visit http://localhost:3000/googlecalendar again. The page will display all the events in the google calendar.

A white background with a black border

Description automatically generated with medium confidence

    5.4 Try to add new events in the google calendar associated with the email. New events will be displayed immediately.

A screenshot of a calendar

Description automatically generatedA screenshot of a calendar

Description automatically generated

A screenshot of a computer

Description automatically generated