# Integrate with Prisma

## Setup Prisma

1. npm I prisma

2. npx prisma init – generate prisma schema file

3. open prisma schema file and setup the database and connection url. Prisma support multi type of database, for this case we use mongo db

Text

Description automatically generated

4. Update database while model update(migration) with command npx prisma db push

## Prisma data model

1. Data type allow nullable, exp: String?

2. List of data type:

- String

- DateTime

-Int

3. api attribute

-@id

-@default(auto())

-@default(autoincrement())

- @unique

- @updatedAt

- @default(now())

- @db.ObjectId

- @relation

- @map

## Prisma data model relations

Example relationship between Tweet and User model

Text

Description automatically generated

Text

Description automatically generated

# Create Nuxt 3 App

1. npx nuxi init \_\_\_

2. npm i

# Nuxt 3 server (api endpoint)

1. Basic structure

**Text

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated with medium confidence**

**/api/auth/login – post method**

2. index and param

Graphical user interface, text, application

Description automatically generated

3. Deep url – /api/user/tweets – post method

Graphical user interface

Description automatically generated with medium confidence

# Auth Process

## Register with bcrypt package

1. npm I bcrypt

2. encrypt password



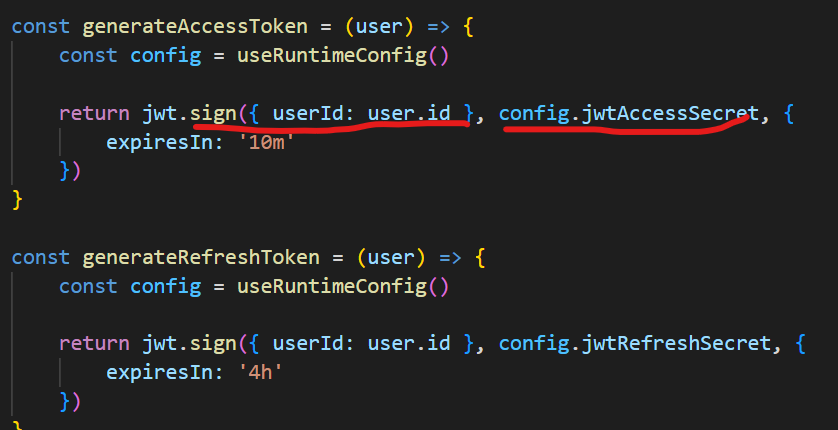
3. Verify password



## Login with jwt

1. npm I jsonwebtoken

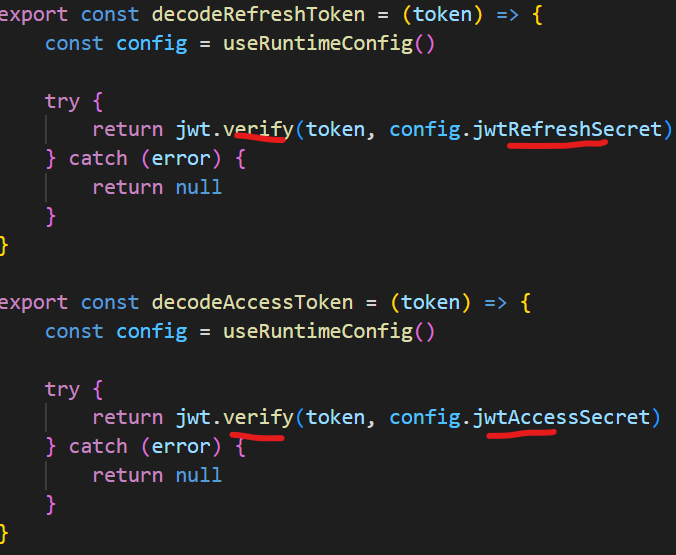
2. Generate access token or refresh token



* Combine value to jwt token which can decoding to get data to the onward process.
* Config was setup on nuxt.config.js.

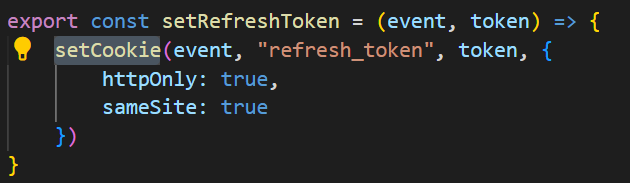
3. Decode token – 2 ways

First method

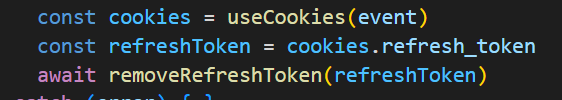


-verify function will validate token and return decode value, exp decoded.userId

4. set cookie



5. get cookie



Second method

1. npm I jwt-decode, then



# Tailwindcss

## Setup

1. npm install @nuxtjs/tailndcss

2. add register tailwindcss in nuxt.config file

Graphical user interface, text

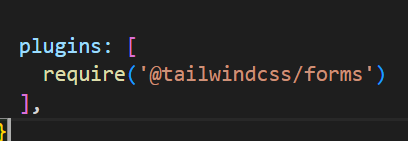
Description automatically generated

3. npx tailwindcss init – generate tailwind.config file

## Tailwind Input class stylish

1. npm I @tailwindcss/forms

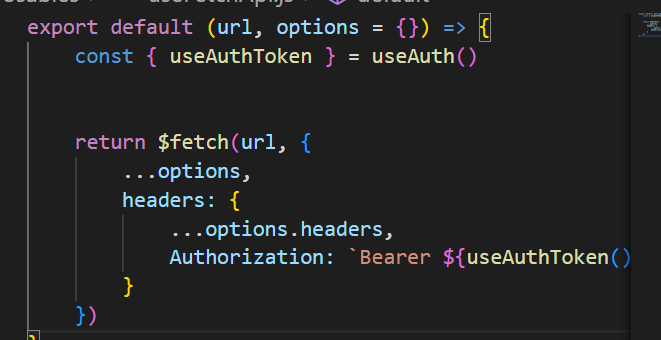
2. Register it to tailwind.config.js



# Others

## url-pattern package - Middleware for api endpoint

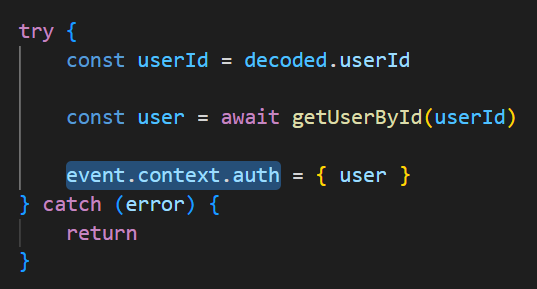
1. Pass authorization to request



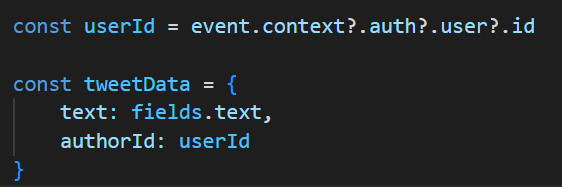
2. Hit the middleware

Text

Description automatically generated



-assign user data to context, then we can get user data globally, exp:



3. Execute authorize endpoint



## Cloudinary – Free cloud for image storage(limit)

Text

Description automatically generated

## mitt – listerning event package

1. add file at composable folder

Text

Description automatically generated

2. listening

Text

Description automatically generated

3. fire event

Text

Description automatically generated with medium confidence

## Heroicons – Ease to use, provide solid and outline version icon

Icon

Description automatically generated with medium confidence

1. npm I @heroicons/vue

## Human-time



1. npm I human-time

2. pass datetime value

Text

Description automatically generated

## Formidable - A Node.js module for parsing form data, especially file uploads.

1. npm I formidable

Text

Description automatically generated

2. Parsing fields and files

Text

Description automatically generated

Text

Description automatically generated