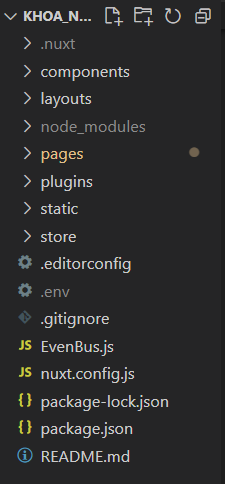
# Cấu trúc của một project NuxtJs



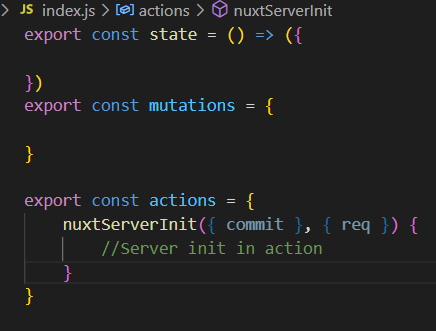
* Nuxt.config.js nơi thiết lập các thư viện được import: axios, element ui, .env, …
* Thư mục components chứa các thành phần cần thiết của một trang web: header, navbar, footer.
* Thư mục layouts chứa các file default layout, empty hoặc error. Tuỳ mỗi trang mà ta sử dụng layout tương ứng.
* Thư mục pages là thư mục chính chứa view của trang web. Mỗi một page sẽ đại diện cho routes trên đường dẫn.
* Thư mục static dùng để chứa các file css, js, cũng như các thư viện của boostrap
* File .env chứa các biến môi trường như đường dẫn đến BASEWEB hoặc sau này có thể chứa đường dẫn đến mongoDB.



* File api.json để chứa các api phía backend.



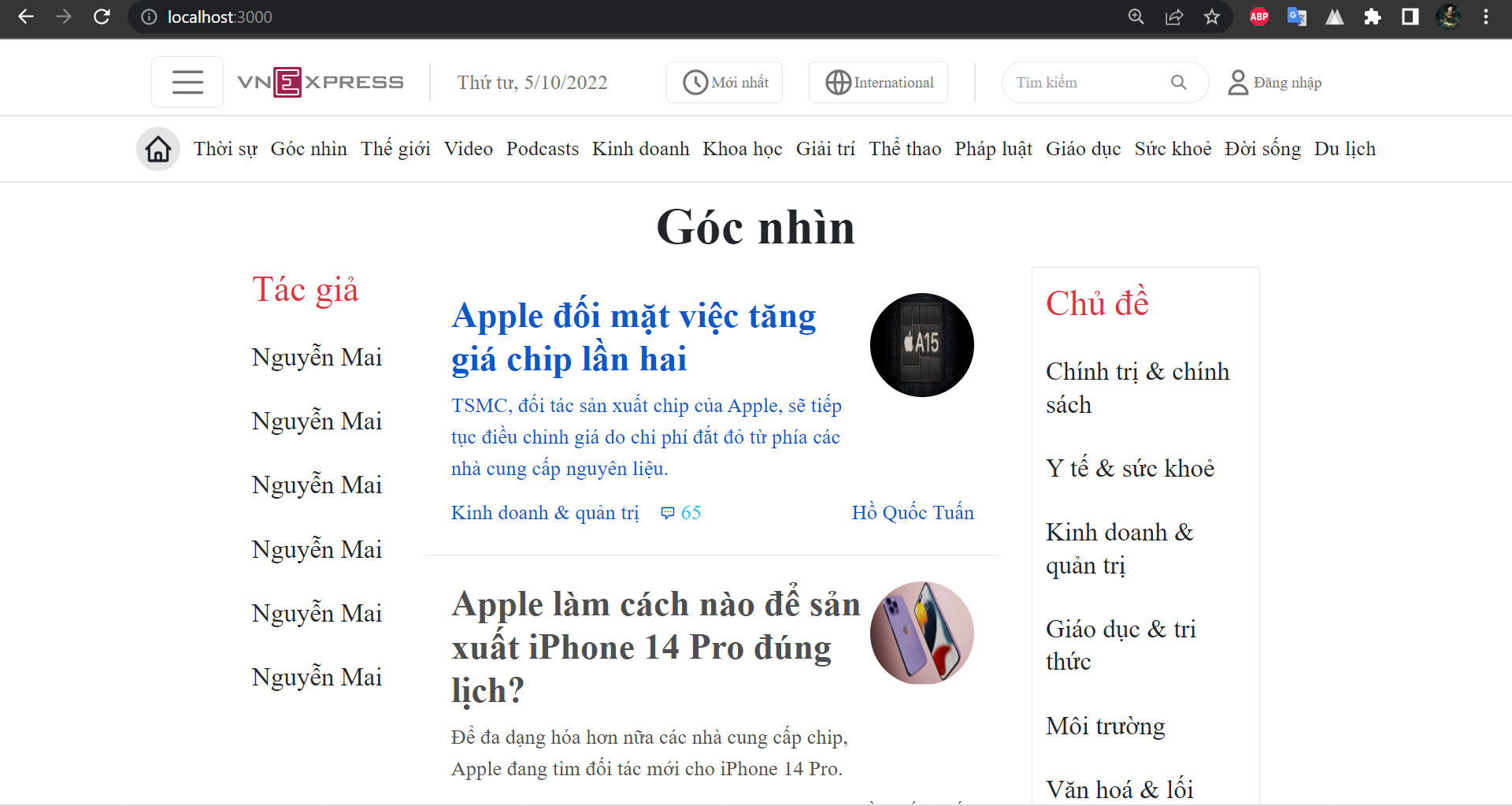
* File package.json dùng để chứa các dependencies.
* Thư mục store: Nơi gọi tới các api.
  + File chính của store là index.js bao gồm state, mutations, actions để khai báo cho ứng dụng biết.



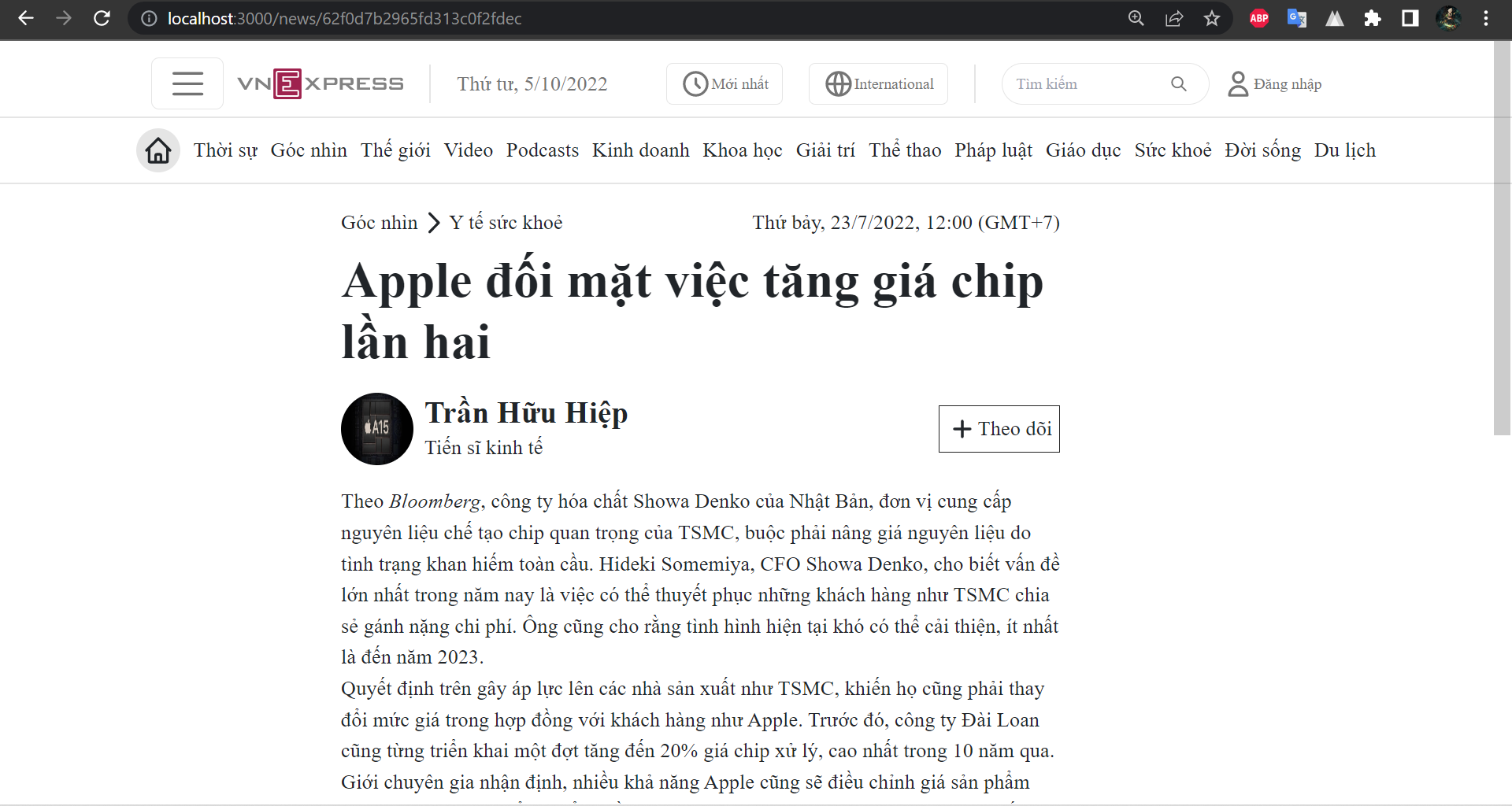
* State khởi tạo các biến
* Mutations: Cập nhật lại các biến
* Actions: Nơi gọi các api xử lý.

# Chức năng

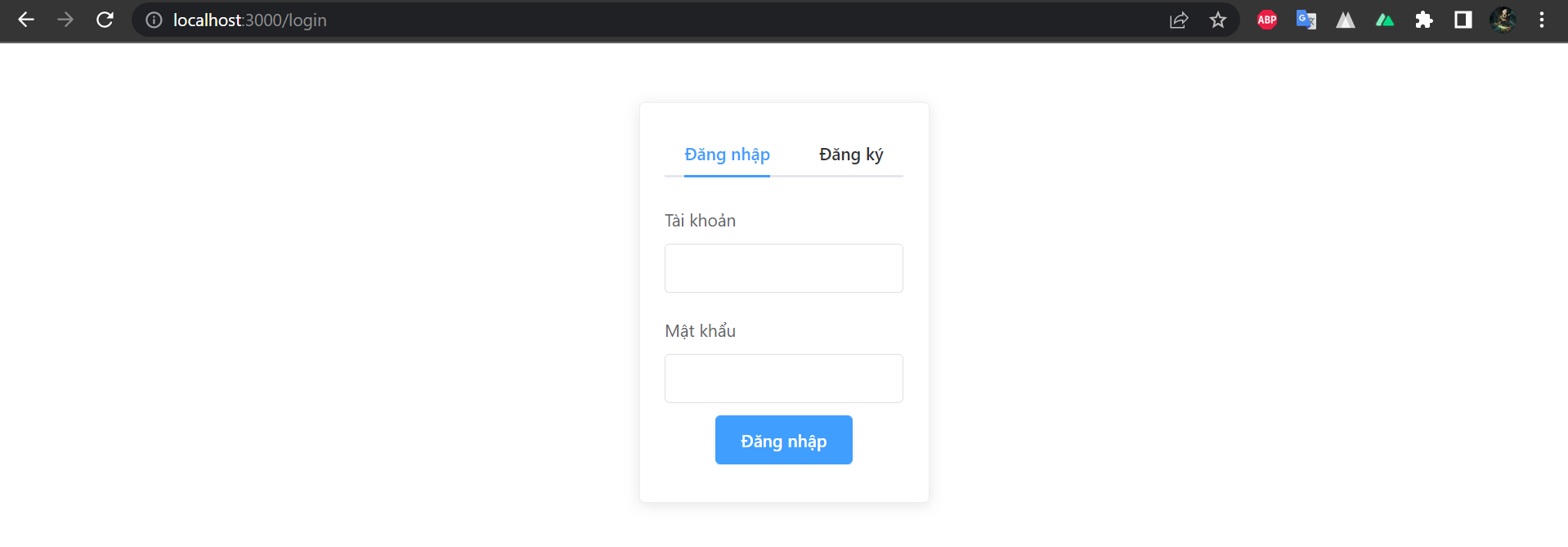
* Trang web hiển thị các tin tức.



* Hiển thị chi tiết của một tin

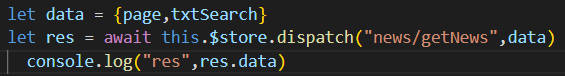


* Đăng nhập



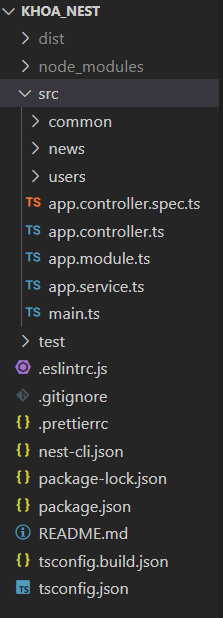
# Cách gọi api

* Thực hiện lệnh $store.dispatch để tạo action.



* Store sẽ nhận action và tìm đến action đó.

# Cấu trúc nestjs



Mô hình trong nest:

Controller: Nơi đầu tiên tiếp nhận các yêu cầu của client gửi đến. Nơi tiếp nhận request từ client như GET, POST, PUT, DELETE. Tìm tới các hàm tương ứng và nhận dữ liệu từ body, param, query truyền cho lớp service xử lý.

Service: Lớp trung gian nhận dữ liệu từ controller và xử lý logic hoặc chuyển kết quả cho một lớp trung gian khác xử lý. Lớp này không được phép truy cập vào database mà phải thông qua một lớp khác là lớp Helper.

Helper là lớp được phép truy vấn trên cơ sở dữ liệu. Thực hiện truy vấn xem, thêm, xoá, sửa…

File entity: Ánh xạ một bảng trong database

Lớp dto: Lớp này đóng gói các thuộc tính trong entity. Được lớp service gọi đến nhằm giảm đi lượng thông tin không cần phải truyền vào.

GET: Lấy ra bản ghi trong cơ sở dữ liệu

POST: Tạo mới bản ghi

PUT: Thay đổi toàn bộ bản ghi

PATCH: Thay đổi một phần bản ghi

DELETE: Xoá bản ghi

# Chức năng đăng nhập

Sử dụng token để nhận biết người dùng nào đăng nhập hay chưa bằng cách:

* Khi người dùng đăng nhập thành công, hệ thống sẽ lưu accessToken và requestToken.
* AccessToken có thời hạn ngắn dùng để truy cập vào một vài chức năng.
* Khi AccessToken hết hạn, người dùng cần yêu cầu một accessToken mới. Lúc này cần tới requestToken (có thời hạn rất dài).
* Kiểm tra request token của người dùng hợp lệ sẽ refresh cho người dùng một access token mới và một request token mới.