Client React

**2024**

A

BNPN

27.12.2024



Оглавление

[Структура клиентского приложения 2](#_Toc186190148)

[Установка 3](#_Toc186190149)

[Код клиентского приложения 4](#_Toc186190150)

[1. my-shop/client/README.md 4](#_Toc186190151)

[2. my-shop/client/public/index.html 5](#_Toc186190152)

[3. my-shop/client/public/favicon.ico 6](#_Toc186190153)

[4. my-shop/client/src/App.js 6](#_Toc186190154)

[5. my-shop/client/src/index.js 7](#_Toc186190155)

[6. my-shop/client/src/api.js 8](#_Toc186190156)

[7. my-shop/client/package.json 9](#_Toc186190157)

[8. my-shop/client/src/components/Auth.js 10](#_Toc186190158)

[9. my-shop/client/src/components/Products.js 12](#_Toc186190159)

[10. my-shop/client/src/components/Cart.js 14](#_Toc186190160)

[11. my-shop/client/src/components/Header.js 15](#_Toc186190161)

[Запуск клиентского приложения 16](#_Toc186190162)

Создание клиентской части интернет магазина может быть реализовано на различного рода фреймворках или библиотеках. Мы создадим простое клиентское приложение на React. Оно будет взаимодействовать с сервером.

# Структура клиентского приложения

Css

my-shop/

└── client/

├── public/

│ ├── index.html

│ └── favicon.ico

├── src/

│ ├── components/

│ │ ├── Auth.js

│ │ ├── Products.js

│ │ ├── Cart.js

│ │ ├── Header.js

│ ├── App.js

│ ├── index.js

│ └── api.js

├── package.json

└── README.md

# Установка

Перейдите в папку client:

Bash  
  
cd my-shop/client

Создайте новое приложение React с помощью Create React App:

Bash

npx create-react-app .

Установите Axios для выполнения HTTP-запросов:

Bash

npm install axios

# Код клиентского приложения

# my-shop/client/README.md

Markdown

# My Shop - Client

This is the client part of the My Shop application built using React.

## Installation

1. Navigate to the client directory:

bash

cd client

markdown

2. Install dependencies:

bash

npm install

shell

## Running the Application To start the application, run:

Bash

npm start

sql

This will start the React development server and open the application in your default web browser. The application will be accessible at `http://localhost:3000`.

## Features

- User registration

- User login

- View products

- Add products to cart

- View cart items

# my-shop/client/public/index.html

html language-xml

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>My Shop</title>

</head>

<body>

<div id="root"></div>

</body>

</html>

# my-shop/client/public/favicon.ico

Файл favicon.ico обычно представляет собой иконку для веб сайта. Вы можете создать свой собственный favicon изображение или использовать готовое.

# my-shop/client/src/App.js

Javascript

import React, { useState } from 'react';

import Header from './components/Header';

import Auth from './components/Auth';

import Products from './components/Products';

import Cart from './components/Cart';

function App() {

const [token, setToken] = useState(null);

return (

<div>

<Header token={token} setToken={setToken} />

{token ? (

<>

<Cart token={token} />

<Products token={token} />

</>

) : (

<Auth setToken={setToken} />

)}

</div>

);

}

export default App;

# my-shop/client/src/index.js

Javascript

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App';

ReactDOM.render(<App />, document.getElementById('root'));

# my-shop/client/src/api.js

javascript

import axios from 'axios';

const API\_URL = 'http://localhost:5000'; // URL вашего API

// Функция для регистрации пользователя

export const registerUser = async (login, password) => {

const response = await axios.post(`${API\_URL}/auth/register`, { login, password });

return response.data;

};

// Функция для авторизации пользователя

export const loginUser = async (login, password) => {

const response = await axios.post(`${API\_URL}/auth/login`, { login, password });

return response.data;

};

// Функция для получения всех продуктов

export const fetchProducts = async () => {

const response = await axios.get(`${API\_URL}/products`);

return response.data;

};

// Функция для добавления продукта в корзину

export const addProductToCart = async (token, productId) => {

const response = await axios.put(`${API\_URL}/cart`, { productId }, {

headers: {

Authorization: `Bearer ${token}`

}

});

return response.data;

};

// Функция для получения корзины

export const fetchCart = async (token) => {

const response = await axios.get(`${API\_URL}/cart`, {

headers: {

Authorization: `Bearer ${token}`

}

});

return response.data;

};

# my-shop/client/package.json

Json

{

"name": "my-shop",

"version": "0.1.0",

"private": true,

"homepage": ".",

"dependencies": {

"axios": "^0.21.1",

"react": "^17.0.2",

"react-dom": "^17.0.2",

"react-scripts": "4.0.3"

},

"scripts": {

"start": "react-scripts start",

"build": "react-scripts build",

"test": "react-scripts test",

"eject": "react-scripts eject"

},

"eslintConfig": {

"extends": [

"react-app",

"react-app/jest"

]

},

"browserslist": {

"production": [

">0.2%",

"not dead",

"not op\_mini all"

],

"development": [

"last 1 chrome version",

"last 1 firefox version",

"last 1 safari version"

]

}

}

# my-shop/client/src/components/Auth.js

Javascript

import React, { useState } from 'react';

import { registerUser, loginUser } from '../api';

function Auth({ setToken }) {

const [login, setLogin] = useState('');

const [password, setPassword] = useState('');

const [isRegistering, setIsRegistering] = useState(true);

const handleSubmit = async (e) => {

e.preventDefault();

if (isRegistering) {

await registerUser(login, password);

} else {

const data = await loginUser(login, password);

setToken(data.token);

}

};

return (

<div>

<h2>{isRegistering ? 'Register' : 'Login'}</h2>

<form onSubmit={handleSubmit}>

<input

type="text"

placeholder="Login"

value={login}

onChange={(e) => setLogin(e.target.value)}

required

/>

<input

type="password"

placeholder="Password"

value={password}

onChange={(e) => setPassword(e.target.value)}

required

/>

<button type="submit">{isRegistering ? 'Register' : 'Login'}</button>

</form>

<button onClick={() => setIsRegistering(!isRegistering)}>

Switch to {isRegistering ? 'Login' : 'Register'}

</button>

</div>

);

}

export default Auth;

# my-shop/client/src/components/Products.js

Javascript

import React, { useEffect, useState } from 'react';

import { fetchProducts, addProductToCart } from '../api';

function Products({ token }) {

const [products, setProducts] = useState([]);

useEffect(() => {

const getProducts = async () => {

const data = await fetchProducts();

setProducts(data);

};

getProducts();

}, []);

const handleAddToCart = async (productId) => {

await addProductToCart(token, productId);

alert('Product added to cart!');

};

return (

<div>

<h2>Products</h2>

<ul>

{products.map(product => (

<li key={product.product\_id}>

{product.title} - ${product.price}

<button onClick={() => handleAddToCart(product.product\_id)}>Add to cart</button>

</li>

))}

</ul>

</div>

);

}

export default Products;

# my-shop/client/src/components/Cart.js

Javascript

import React, { useEffect, useState } from 'react';

import { fetchCart } from '../api';

function Cart({ token }) {

const [cart, setCart] = useState([]);

useEffect(() => {

const getCart = async () => {

const data = await fetchCart(token);

setCart(data);

};

getCart();

}, [token]);

return (

<div>

<h2>Your Cart</h2>

<ul>

{cart.map(item => (

<li key={item.product\_id}>

{item.title} - ${item.price}

</li>

))}

</ul>

</div>

);

}

export default Cart;

# my-shop/client/src/components/Header.js

Javascript

import React from 'react';

function Header({ token, setToken }) {

return (

<header>

<h1>My Shop</h1>

{token ? (

<button onClick={() => setToken(null)}>Logout</button>

) : (

<span>Please login to purchase</span>

)}

</header>

);

}

export default Header;

# Запуск клиентского приложения

Перейдите в папку client, если еще не сделали этого:

Bash

Запустите приложение:

Bash

npm start

Ваше клиентское приложение теперь должно быть доступно по адресу http://localhost:3000. Вы можете зарегистрироваться, войти в систему, просматривать продукты и добавлять их в корзину.

Заключение

Эта структура и функциональность клиентского приложения должны помочь вам начать взаимодействовать с серверной частью интернет магазина. Вы можете расширить приложение, добавив дополнительные функции, такие как просмотр деталей продукта, оформление заказа и т.д.