Код-ревью (из 12 ревью) от 01.12.2024 07:45

Файл: SwipperAmazing.tsx

Строка 31

- **Комментарий:**

  Функция должна быть названа с большой буквы, чтобы соответствовать соглашениям о наименовании компонентов React. Исправьте на SwipperAmazingList.

- **Текущий код:**

```
} from "../../../Pages/Stock/Logics/hooks";


export const SwipperAmazingList = () => {
```

Строка 34

- **Комментарий:**

  Использование useStore для получения значения из стора может быть неэффективным. Рассмотрите возможность использования useSelector из react-redux или аналогичного хука.

- **Текущий код:**

```
export const SwipperAmazingList = () => {

  const number = useStore($checkStock);
```

Строка 37

- **Комментарий:**

  Условие if(number < 0) return null не имеет смысла, так как number не может быть отрицательным. Удалите это условие.

- **Текущий код:**

```
const number = useStore($checkStock);

  if(number < 0) return null
```

Строка 41

- **Комментарий:**

Неправильное использование useState. Переменная stock должна быть типа number, а не string. Исправьте на useState<number>(0).

- **Текущий код:**

```
if(number < 0) return null


const [stock, setStock] = useState('')
```

Строка 45

- **Комментарий:**

StockObjects должен быть типизирован. Создайте интерфейс для объектов и используйте его.

- **Текущий код:**

```
const [stock, setStock] = useState('')


  const StockObjects = [
    {
```

Строка 58

- **Комментарий:**

Функция handleClick содержит слишком много логики. Разделите её на несколько функций для улучшения читаемости и поддержки.

- **Текущий код:**

```
];


  const handleClick = () => {
    setcheckAmazing(false);
```

Строка 64

- **Комментарий:**

Использование цепочки fetch запросов приводит к callback hell. Используйте async/await для улучшения читаемости.

- **Текущий код:**

```
alert("SwipperAmazingList closed");


    fetch("https://api.example.com/logClose", {
      method: "POST",
```

Строка 70

- **Комментарий:**

Необходимо обработать ошибки при отправке данных на сервер. Добавьте try-catch блок.

- **Текущий код:**

```
headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify({ closedAt: new Date().toISOString() }),
    })
```

Строка 113

- **Комментарий:**

  Использование Math.random() для генерации индекса не является безопасным. Рассмотрите использование более безопасного метода.

- **Текущий код:**

```
    console.log("Middle item was active when closed.");
        }

        const randomIndex = Math.floor(Math.random() * StockObjects.length);
        setStock(randomIndex);
```

Строка 134

- **Комментарий:**

  Использование any для типа swiper не рекомендуется. Определите точный тип для swiper.

- **Текущий код:**

```
    keyboard={true}
              modules={[Navigation, Pagination, Mousewheel, Keyboard]}
              className="mySwiper"
              onRealIndexChange={(swiper: any) => {
                setcheckStock(swiper.activeIndex);
```

Строка 143

- **Комментарий:**

  Использование map для рендеринга элементов без ключа может привести к проблемам с производительностью. Убедитесь, что ключ уникален.

- **Текущий код:**

```
    }}
           >
```

```
{StockObjects.map((e: any) => (
  <SwiperSlide key={e.name}>
```

Строка 154

- **Комментарий:**

  Использование map для рендеринга одного элемента неэффективно. Используйте StockObjects[number] для прямого доступа к элементу.

- **Текущий код:**

```
<div className="SwipperAmazingList__Info">
          <div className="SwipperAmazingList__Info__Name">

            {StockObjects.map((e: any, i: any) =>
              i === number ? e.name : null
```
