

Реализация веб-редактора форматированного текста на основе библиотеки TipTap в React/TypeScript-приложении

Андриенко Иван Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Пащенко Екатерина Александровна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной работе рассматривается реализация веб-редактора форматированного текста на основе библиотеки TipTap в одностраничном приложении, разработанном с использованием React и TypeScript. Описывается процесс интеграции редактора в административный интерфейс для создания и редактирования новостного контента, а также использование сгенерированного HTML на пользовательских страницах для отображения полной версии новости. Приводится структура компонента TiptapEditor, включающего настройку базового набора расширений StarterKit и расширения ссылок Link, реализацию панели инструментов с кнопками форматирования и обработку изменений через передачу HTML-контента в родительский компонент.

Ключевые слова: TipTap, веб-редактор, React, TypeScript, форматированный текст, новости, пользовательский интерфейс.

Implementation of a web version for the visually impaired based on React and CSS global styles

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University

Student

Paschenko Ekaterina Alexandrovna

Sholom-Aleichem Priamursky State University

Student

Abstract

This paper examines the implementation of a web-based formatted text editor based on the TipTap library in a single-page application developed using React and TypeScript. It describes the process of integrating the editor into the administrative interface for creating and editing news content, as well as using the generated HTML on user pages to display the full version of the news. The structure of the

TiptapEditor component is given, which includes configuring the basic set of StarterKit extensions and Link extensions, implementing a toolbar with formatting buttons, and processing changes by transferring HTML content to the parent component.

Keywords: TipTap, web editor, React, TypeScript, formatted text, news, user interface.

1 Введение

1.1 Актуальность

Современные веб-приложения, ориентированные на публикацию новостного контента и информационных материалов, требуют наличия удобных средств редактирования текстов с поддержкой форматирования. Использование простых текстовых полей без возможности выделения, списков и гиперссылок ограничивает выразительность материалов и снижает удобство работы редакторов и администраторов. В то же время внедрение полноценного визуального редактора должно учитывать архитектуру приложения, особенности фреймворка и требования к безопасности. Для проектов на базе React и TypeScript актуальной задачей является интеграция редактора, который позволит создавать форматированный контент в административной части и корректно отображать его на пользовательских страницах без нарушения типовой структуры компонентов. Библиотека TipTap, основанная на ProseMirror, предоставляет расширяемый и программируемый интерфейс для создания таких редакторов и хорошо сочетается с компонентной моделью React, что делает её привлекательным решением для задач управления контентом новостных разделов.

1.2 Обзор исследований

Вопросы разработки инструментов редактирования контента и их роли в структуре веб-приложений рассматриваются в различных научных работах. В работе О.П. Сотниковой рассматриваются особенности веб-редактирования и требования к деятельности редактора в цифровой среде, что подчёркивает важность инструментов корректного формирования контента [1]. Исследование С.В. Захарина и соавторов анализирует современные веб-технологии, применяемые при создании редакторов и интерактивных интерфейсов обработки данных, включая растровые инструменты [2]. В работе М.М. Ковцура, В.Е. Филимонова и Е.К. Щёголева анализируются подходы к созданию многопользовательских систем на базе React, где особое внимание уделяется организации типизации и архитектуры клиентской части [3]. А.А. Дудак в своём исследовании проводит сравнительный анализ инструментов разработки для проектных систем, выделяя преимущества стека технологий React и TypeScript по производительности и удобству сопровождения кода [4]. В статье Д.А. Хайруллина и М.Р. Богданова рассматриваются особенности применения статической типизации в React-приложениях, включая методики типизации компонентов, хуков и состояния,

а также влияние строгой типизации на надёжность пользовательских интерфейсов [5].

1.3 Цель исследования

Целью данной работы является разработка и описание интеграции редактора форматированного текста на основе TipTap в React/TypeScript-приложение, используемое для публикации новостного контента. В рамках исследования рассматриваются настройка компонента редактора, подключение расширений для списков и ссылок, организация панели инструментов для форматирования текста, передача сгенерированного HTML в административные формы и безопасное отображение результата на пользовательских страницах.

2 Материалы и методы

В качестве основы использовалось одностраничное веб-приложение, разработанное с использованием библиотеки React и языка TypeScript. Для реализации редактора был создан отдельный компонент TiptapEditor, использующий хук useEditor и компонент EditorContent из пакета @tiptap/react.

3 Результаты и обсуждения

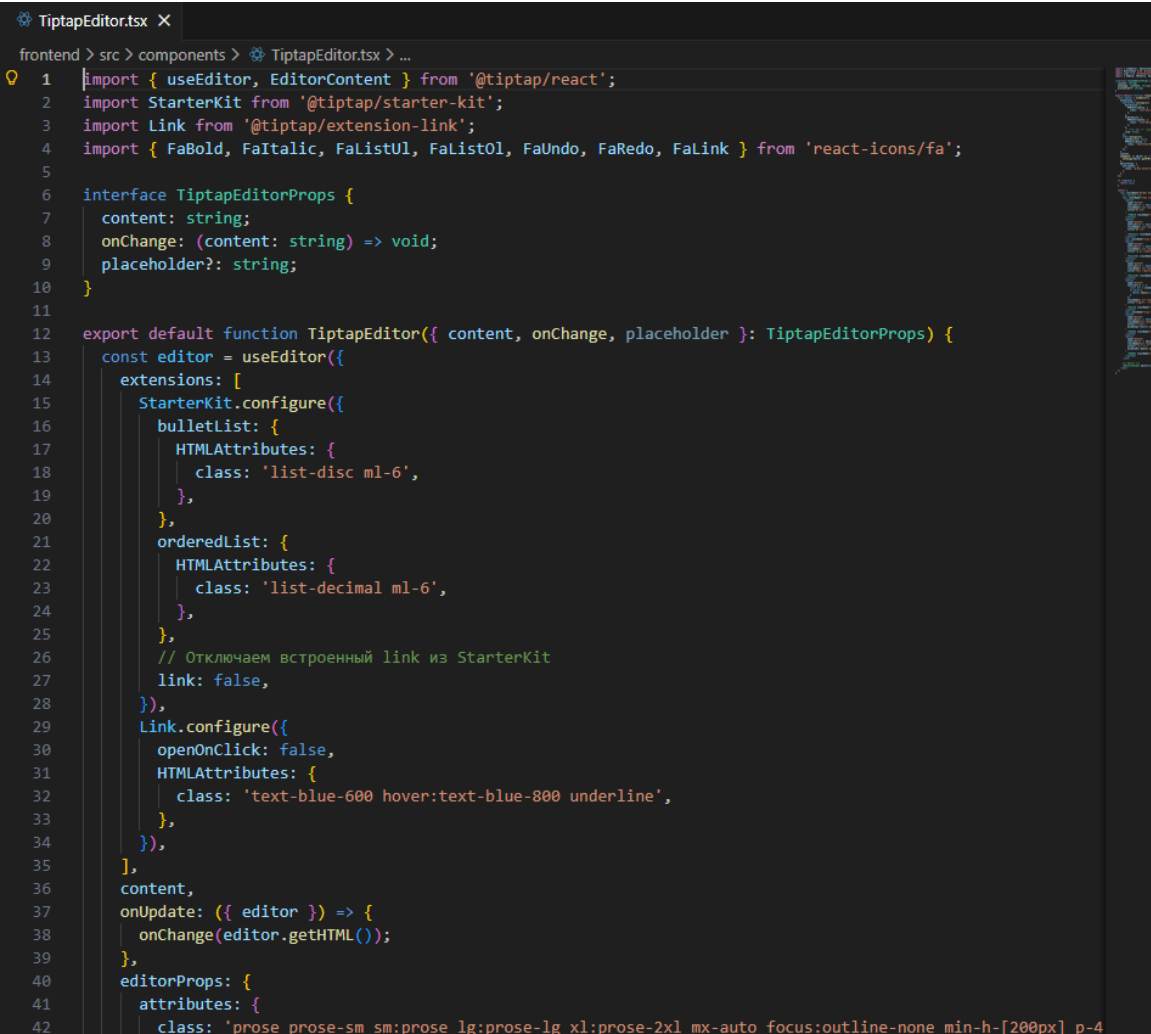
Реализация визуального редактора форматированного текста TipTap в React/TypeScript-приложении была выполнена посредством создания отдельного компонента TiptapEditor, который отвечает за обработку вводимого текста, применение стилей и передачу итогового HTML в административные интерфейсы. В первую очередь был разработан основной компонент редактора, использующий расширяемую архитектуру TipTap и включающий подключение набора расширений StarterKit и Link. Данный компонент определяет структуру редактора, набор инструментов форматирования и параметры отображения текстовой области. Полный фрагмент реализации компонента показан на рисунке (рис. 1,2).

Ключевым элементом редактора является вызов хука useEditor, через который выполняется конфигурация расширений. StarterKit предоставляет базовые операции редактирования текста, включая выделение, списки, абзацы и горячие клавиши, в то время как расширение Link обеспечивает работу с гиперссылками. В процессе конфигурации StarterKit были отключены встроенные ссылки, поскольку их обработка реализуется через отдельный модуль Link. Одновременно были заданы HTML-классы для маркированных и нумерованных списков, что обеспечивает корректное визуальное отображение в соответствии с дизайном сайта.

Помимо расширений редактора, важную роль играет обработчик onUpdate, который вызывается при каждом изменении содержимого и передаёт сформированный HTML в родительский компонент. Это позволяет использовать редактор в качестве контролируемого компонента, синхронизирующегося с административной формой. Для повышения

доступности текстовой области были заданы атрибуты `editorProps`, определяющие внешний вид и минимальные размеры зоны редактирования.

Интерактивная панель инструментов редактора содержит кнопки форматирования, каждая из которых вызывает соответствующие команды цепочки `editor.chain().focus().run()`. Набор инструментов включает операции выделения текста жирным и курсивом, создание маркированных и нумерованных списков, вставку гиперссылок, а также команды отмены и повтора. Все элементы панели используют иконки из библиотеки `react-icons`, что обеспечивает компактное и интуитивно понятное управление редактором.



```

TiptapEditor.tsx X
frontend > src > components > TiptapEditor.tsx > ...
1 import { useEditor, EditorContent } from '@tiptap/react';
2 import StarterKit from '@tiptap/starter-kit';
3 import Link from '@tiptap/extension-link';
4 import { FaBold, FaItalic, FaListUl, FaListOl, FaUndo, FaRedo, FaLink } from 'react-icons/fa';
5
6 interface TiptapEditorProps {
7   content: string;
8   onChange: (content: string) => void;
9   placeholder?: string;
10 }
11
12 export default function TiptapEditor({ content, onChange, placeholder }: TiptapEditorProps) {
13   const editor = useEditor({
14     extensions: [
15       StarterKit.configure({
16         bulletList: {
17           HTMLAttributes: {
18             class: 'list-disc ml-6',
19           },
20         },
21         orderedList: {
22           HTMLAttributes: {
23             class: 'list-decimal ml-6',
24           },
25         },
26         // Отключаем встроенный link из StarterKit
27         link: false,
28       }),
29       Link.configure({
30         openOnClick: false,
31         HTMLAttributes: {
32           class: 'text-blue-600 hover:text-blue-800 underline',
33         },
34       }),
35     ],
36     content,
37     onUpdate: ({ editor }) => {
38       onChange(editor.getHTML());
39     },
40     editorProps: {
41       attributes: {
42         class: 'prose prose-sm sm:prose lg:prose-lg xl:prose-2xl mx-auto focus:outline-none min-h-[200px] p-4
  
```

Рис. 1. Полный код компонента визуального редактора TiptapEditor

```

89     type="button"
90     onClick={() => {
91         const url = window.prompt('Введите URL ссылки:');
92         if (url) {
93             editor.chain().focus().setLink({ href: url }).run();
94         }
95     }}
96     className={`p-2 rounded hover:bg-gray-200 ${editor.isActive('link')} ? 'bg-blue-100 text-blue-600'
97     title="Добавить ссылку"
98     >
99     <FaLink className="w-4 h-4" />
100 </button>
101 <div className="w-px h-6 bg-gray-300 mx-1" />
102 <button
103     type="button"
104     onClick={() => editor.chain().focus().undo().run()}
105     className="p-2 rounded hover:bg-gray-200"
106     title="Отменить"
107     disabled={!editor.can().undo()}
108     >
109     <FaUndo className="w-4 h-4" />
110 </button>
111 <button
112     type="button"
113     onClick={() => editor.chain().focus().redo().run()}
114     className="p-2 rounded hover:bg-gray-200"
115     title="Повторить"
116     disabled={!editor.can().redo()}
117     >
118     <FaRedo className="w-4 h-4" />
119 </button>
120 </div>
121
122 { /* Editor */ }
123 <EditorContent editor={editor} />
124 </div>
125 );
126 }

```

Рис. 2. Полный код компонента визуального редактора TiptapEditor

Интеграция редактора в административную панель была выполнена в компоненте AdminNews. Здесь TiptapEditor используется для заполнения поля полного текста новости, где значение редактора соответствует полю fullText в объекте состояния formData. При каждом изменении содержимого редактор обновляет HTML-контент, который затем сохраняется вместе с другими атрибутами новости. Фрагмент кода, реализующий подключение редактора в административной форме, показан на рисунке (рис. 3).

```

299 </div>
300
301 <div>
302     <label className="block text-sm font-medium text-gray-700 mb-1">
303         Полный текст новости *
304     </label>
305     <TiptapEditor
306         content={formData.fullText}
307         onChange={({content}) => setFormData({ ...formData, fullText: content })}
308         placeholder="Полный текст новости"
309     />
310 </div>
311
312 <div>

```

Рис. 3. Использование компонента TiptapEditor в административной панели новостей

Полученный HTML-контент выводится на пользовательской стороне в компоненте NewsDetail. Здесь используется контейнер div, внутри которого данные отображаются через dangerouslySetInnerHTML. Данный механизм

позволяет отобразить HTML точно в том виде, в котором он был создан редактором. Однако использование него требует строгого контроля за источником данных. В рассматриваемой системе контент создаётся исключительно администраторами через интерфейс TipTap, что исключает возможность внедрения вредоносного кода со стороны пользователей. Кроме того, TipTap автоматически экранирует потенциально опасные элементы, обеспечивая дополнительный уровень защиты. Фрагмент пользовательской части отображения приведён на рисунке (рис. 4).

```
180
181      { /* Полный текст */ }
182      <div className="prose prose-lg max-w-none mb-8">
183        <div
184          className="text-gray-800 leading-relaxed"
185          dangerouslySetInnerHTML={{ __html: news.fullText }}
186        />
187      </div>
188
```

Рис. 4. Отображение результата работы TipTap на пользовательской странице NewsDetail

Дополнительные меры безопасности включают ограничение источников редактирования текста и отказ от принятия контента со стороны внешних пользователей. Это делает использование `dangerouslySetInnerHTML` безопасным в рамках текущей архитектуры. При отсутствии данных редактор формирует пустой HTML-блок, который корректно обрабатывается фронтендом через стандартные fallback-механизмы.

После реализации функциональной части редактора и его интеграции в административный интерфейс была проведена визуальная оценка итогового пользовательского опыта (рис. 5).

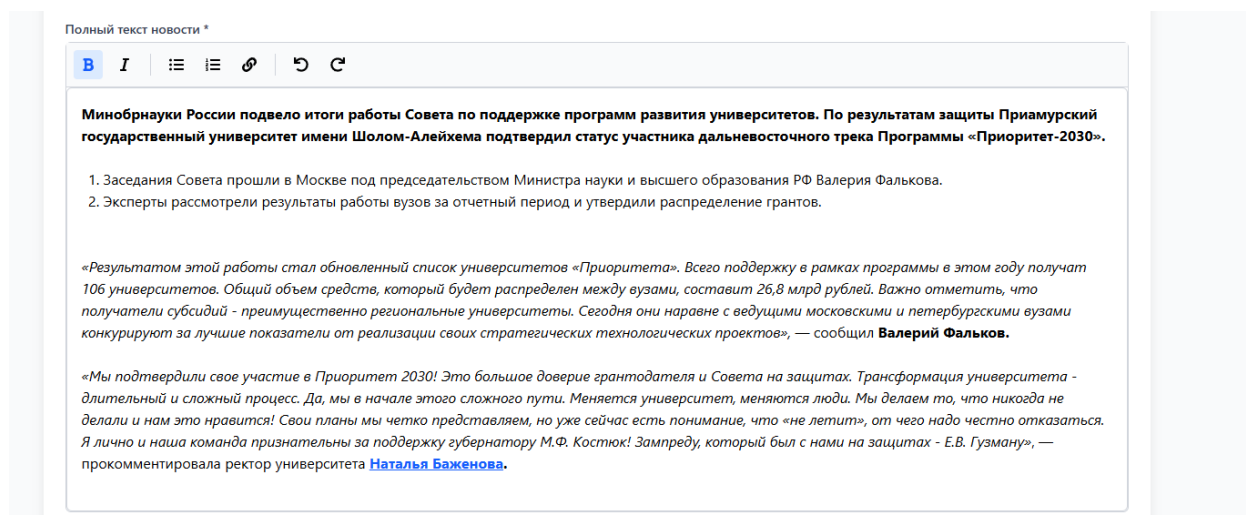


Рис. 5. Внешний вид визуального редактора TipTap в административной части сайта

На пользовательской стороне итоговый HTML-контент отображается полностью отформатированным, включая корректную передачу структурных элементов, списков, ссылок и выделений (рис. 5).

Минобрнауки России подвело итоги работы Совета по поддержке программ развития университетов. По результатам защиты Приамурский государственный университет имени Шолом-Алейхема подтвердил статус участника дальневосточного трека Программы «Приоритет-2030».

1. Заседания Совета прошли в Москве под председательством Министра науки и высшего образования РФ Валерия Фалькова.

2. Эксперты рассмотрели результаты работы вузов за отчетный период и утвердили распределение грантов.

«Результатом этой работы стал обновленный список университетов «Приоритета». Всего поддержку в рамках программы в этом году получают 106 университетов. Общий объем средств, который будет распределен между вузами, составит 26,8 млрд рублей. Важно отметить, что получатели субсидий - преимущественно региональные университеты. Сегодня они наравне с ведущими московскими и петербургскими вузами конкурируют за лучшие показатели от реализации своих стратегических технологических проектов», — сообщил Валерий Фальков.

«Мы подтвердили свое участие в Приоритет 2030! Это большое доверие грантодателя и Совета на защитах. Трансформация университета - длительный и сложный процесс. Да, мы в начале этого сложного пути. Меняется университет, меняются люди. Мы делаем то, что никогда не делали и нам это нравится! Свои планы мы четко представляем, но уже сейчас есть понимание, что «не летит», от чего надо честно отказаться. Я лично и наша команда признательны за поддержку губернатору М.Ф. Костю! Зампреду, который был с нами на защитах - Е.В. Гузмину», — прокомментировала ректор университета Наталья Баженова.

Рис. 5. Отображение форматированного текста, созданного через TipTap, на пользовательской странице

4 Выводы

Результаты работы демонстрируют успешную интеграцию визуального редактора TipTap в структуру React/TypeScript-приложения, что позволило создать полноценный инструмент для формирования и редактирования новостного контента. Разработанный компонент TiptapEditor обеспечил поддержку необходимых элементов форматирования, включая списки, ссылки и базовые текстовые операции, а также корректную передачу HTML-контента в административные формы. Использование механизма dangerouslySetInnerHTML для отображения данных на пользовательской стороне показало свою надёжность при условии контролируемого источника контента. Проведённое тестирование подтвердило стабильность работы редактора и корректность отображения форматированного текста, что подтверждает целесообразность применения TipTap в качестве решения для управления структурированным текстом в веб-приложениях.

Библиографический список

1. Сотникова О.П. Веб-редактирование: содержание и формы деятельности редактора: дис. ... канд. филол. наук. М.: Московский государственный университет печати имени Ивана Фёдорова, 2014.
2. Захарин С.В., Коровкин А.Д., Кириленко У.О., Рапота М.В., Коновалов А.В. Исследование технологий веб-разработки для реализации растрового редактора // Теоретическая информатика и компьютерные технологии. Материалы I научно-практической конференции. Москва, 2024. С. 50–52.
3. Ковцур М.М., Филимонов В.Е., Щёголев Е.К. Исследование подходов для реализации многопользовательского React-приложения на языке TypeScript // Мобильный бизнес: перспективы развития и реализации систем радиосвязи в России и за рубежом. Материалы 53-й Международной конференции. Москва, 2024. С. 58–61.
4. Дудак А.А. Сравнительный анализ инструментов разработки для систем

-
- управления проектами: преимущества стек-технологий TypeScript и React // Новая наука: от идеи к результату. 2024. № 9. С. 32–40.
5. Хайруллин Д.А., Богданов М.Р. Изучение использования TypeScript в React-приложениях, преимущества статической типизации, подходы к типизации компонентов, хуков и состояния // Международный журнал гуманитарных и естественных наук. 2024. № 12-3 (99). С. 233–238.