

Гайд по изменению кода Википедии WFRP

Структура проекта

```
wfrp-wiki/src/  
├── components/           # React компоненты  
│   ├── WikiArticle.tsx  # Отображение статей  
│   ├── Sidebar.tsx      # Боковая панель + навигация  
│   ├── CreateArticle.tsx # Создание новых статей  
│   ├── EditArticle.tsx  # Редактирование статей  
│   ├── LoginForm.tsx    # Форма входа  
│   └── HomePage.tsx     # Главная страница  
├── hooks/                # Логика приложения  
│   ├── useWikiData.ts   # Управление статьями  
│   └── useAuth.ts        # Авторизация  
├── types.ts              # TypeScript типы  
├── index.css              # Глобальные стили  
└── App.tsx               # Главный компонент
```

Изменение дизайна

Изменить цвета и стили

```
# Файл: src/index.css  
# Изменять CSS переменные и TailwindCSS классы
```

Изменить шрифты

```
/* src/index.css */
body {
  font-family: 'Ваш шрифт', Georgia, serif;
}
```

Изменить макет страницы

```
// src/App.tsx - основная структура
// src/components/Sidebar.tsx - боковая панель
```



Изменение функциональности

1. Добавить новое поле в статьи

Шаг 1: Обновить типы

```
// src/types.ts
export interface WikiArticle {
  title: string;
  content: string;
  category: string;
  author?: string;
  lastModified?: string;
  tags?: string[]; // НОВОЕ ПОЛЕ
}
```

Шаг 2: Обновить форму создания

```
// src/components/CreateArticle.tsx
// Добавить поле ввода для tags
```

Шаг 3: Обновить отображение

```
// src/components/WikiArticle.tsx
// Показать новое поле в статье
```

2. Изменить логику авторизации

```
// src/hooks/useAuth.ts
const login = (username: string, password: string): boolean => {
  // Изменить логин/пароль или добавить новых пользователей
  if (username === 'новый_логин' && password === 'новый_пароль') {
    // ...
  }
}
```

3. Добавить новую страницу

Создать компонент:

```
// src/components/НоваяСтраница.tsx
import React from 'react';

export const НоваяСтраница: React.FC = () => {
  return <div>Контент новой страницы</div>;
};
```

Добавить в навигацию:

```
// src/App.tsx
// Добавить состояние и условие для новой страницы
```



Изменение контента

Добавить статьи через JSON

```
// public/wfrp_wiki_data.json
{
  "Новая статья": {
    "title": "Новая статья",
    "content": "# Заголовок\nТекст статьи...",
    "category": "Основное",
    "links": ["связанная статья"]
  }
}
```

Изменить главную страницу

```
// src/components/HomePage.tsx
// Изменить содержимое главной страницы
```

✳ Процесс разработки

1. Запуск в режиме разработки

```
cd wfrp-wiki  
pnpm dev # Запуск на http://localhost:5173
```

2. Отслеживание изменений

- Изменения автоматически перезагружаются
- Ошибки отображаются в консоли браузера (F12)
- TypeScript ошибки показываются в терминале

3. Тестирование

```
# Проверить сборку  
pnpm build  
  
# Проверить типы  
pnpm type-check # если настроено
```



Применение изменений

1. Локальное тестирование

```
pnpm dev  
# Проверить все функции работают
```

2. Сборка для продакшена

```
pnpm build  
# Результат в папке dist/
```

3. Развертывание

```
# Скопировать содержимое dist/ на ваш сервер  
cp -r dist/* /path/to/your/server/
```



Частые изменения

Изменить название сайта

```
// src/components/Sidebar.tsx  
<h2 className="text-xl font-bold mb-4">  
  Новое название  { /* ЗДЕСЬ */ }  
</h2>
```

Добавить новую категорию статей

```
// src/types.ts  
export type ArticleCategory = 'Основное' | 'Разделы' | 'Служебное'  
| 'Новая категория';  
  
// src/components/CreateArticle.tsx  
// Добавить в список категорий
```

Изменить цвета категорий

```
// src/components/Sidebar.tsx
const getCategoryColor = (category: string) => {
  switch (category) {
    case 'Новая категория': return 'text-purple-600'; // НОВЫЙ
    // ...
  }
}
```

Добавить новую иконку

```
# Установить Lucide React иконки (уже установлено)
# Импортировать и использовать
```

```
import { НоваяИконка } from 'lucide-react';

<НоваяИконка className="w-4 h-4" />
```



Отладка

Общие проблемы

```
# Проблемы с зависимостями
rm -rf node_modules pnpm-lock.yaml
pnpm install

# Проблемы с TypeScript
# Проверить src/types.ts на ошибки

# Проблемы с стилями
# Проверить src/index.css и Tailwind классы
```

Полезные команды

```
# Очистить кэш
pnpm dev --force

# Проверить ошибки линтера
pnpm lint # если настроено

# Анализ размера сборки
pnpm build --analyze # если настроено
```



Полезные советы

1. **Всегда тестируйте локально** перед развертыванием
2. **Сохраняйте резервные копии** данных перед большими изменениями
3. **Используйте TypeScript** - он поможет избежать ошибок

4. **Следуйте структуре проекта** - не смешивайте логику и компоненты

5. **Комментируйте сложный код** для будущих изменений

Горячие клавиши в разработке:

- `F12` - Консоль браузера
 - `ctrl+c` - Остановить сервер разработки
 - `ctrl+Shift+R` - Жесткая перезагрузка браузера
-

Автор: MiniMax Agent | **Обновлено:** 19.06.2025