



# Информация для установки и эксплуатации ПО

На 5 листах



## Общие положения

АРРОВОХ - это программный код, базирующийся на современных технологиях Frontend разработки, чтобы реализовывать гибкие и масштабируемые Web-приложения.

## Системные требования

1. Node.js 18.17 или выше;
2. macOS, Windows (включая WSL) или Linux.

## Поддержка браузеров

1. Браузеры Chromium 108 версии и выше – Chrome, Yandex, Edge;
2. Firefox 101 версии и выше;
3. Safari 15.4 версии и выше;
4. IE (любая версия) не поддерживаются.

## Документация по технологиям

1. Node.js - платформа разработки: <https://nodejs.org/docs/latest-v20.x/api/>
2. Typescript - язык разработки: <https://www.typescriptlang.org/docs/>
3. React - библиотека для создания веб-приложений: <https://react.dev/learn>
4. Next.js - React фреймворк для создания веб-приложений: <https://nextjs.org/docs>
5. Mantine - UI фреймворка для разработки веб-интерфейсов:  
<https://mantine.dev/getting-started/>
6. Redux Toolkit - библиотека для создания централизованного хранилища данных: <https://redux-toolkit.js.org/rtk-query/overview>

## Инструкция по установке и запуску

1. Скачайте исходный код проекта. После покупки на вашу почту пользователя GitFlic предоставится доступ к Git репозиторию для скачивания актуальной версии проекта. В течение 72 часов после активации доступа, вы можете скачать ваш экземпляр. Если вы не обладаете аккаунтом GitFlic - запросите у нас ZIP архив с исходным кодом вашего экземпляра ПО;
2. Проверьте системные требования к ОС (см. выше);
3. Установите Node.js в вашу ОС (см. документацию выше);
4. Установите зависимости Node.js. В папке проекта необходимо выполнить команду выбранного вами пакетного менеджера (YARN, NPM, PNPM и.т.д). Пример запуска команды для установки зависимостей при использовании стандартного пакетного менеджера NPM:

```
# команда для установки Node.js зависимостей  
npm install
```

5. Запустите проект. В папке проекта необходимо выполнить команду для запуска проекта в режиме разработки. Пример команды запуска проекта при использовании стандартного пакетного менеджера NPM:

```
# команда для запуска проекта в режиме разработки  
npm run dev
```

В результате выполнения данной команды ваш сайт должен запуститься по следующему URL - <http://localhost:3000>. Если порт 3000 занят, то в консольном окне запуска будет указан URL маршрут с другим портом, который будет использован для работы.

6. Поздравляем с запуском АРРОВОХ! Теперь вы можете перейти к разработке вашего проекта. [Рекомендуем обратиться к руководству пользователя.](#)

# Node.js команды для работы

Список Node команд для работы с проектом (выполняются на уровне каталога проекта):

1. `dev`  
Запуск проекта в режиме разработки
2. `pre:build`  
Сборка статического JSON контента
3. `build:stage`  
Сборка проекта в STAGE режиме
4. `start:stage`  
Запуск проекта в STAGE режиме
5. `build`  
Сборка проекта в PRODUCTION режиме
6. `start`  
Запуск проекта в PRODUCTION режиме
7. `analyze`  
Сборка проекта в PRODUCTION режиме для анализа
8. `export`  
Сборка проекта в PRODUCTION режиме и его экспорт
9. `typecheck`  
Анализ проекта на предмет ошибок Typescript
10. `lint`  
Запуск проверки ESLint
11. `lint:fix`  
Запуск проверки ESLint в режиме исправления
12. `jest`  
Запуск Jest
13. `jest:watch`  
Запуск Jest в режиме отслеживания

14. `prettier:check`

Запуск проверки Prettier

15. `prettier:watch`

Запуск проверки Prettier в режиме исправления

16. `test`

Запуск всех команд для проверки корректности кода, включая запуск тестирования Jest