

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

Отчет

Домашняя работа №4

“Изучение основ работы с менеджером зависимостей  
npm”

Выполнил:  
Стукалов Артем

Группа  
К33392

Проверил:  
Добряков Д. И.

Санкт-Петербург

2023 г.

## **Задача**

Изучить основные команды пакетного менеджера NPM и научиться стартовать проект на Vue/React

## **Ход работы**

Выполнение работы будет продемонстрировано на примере создания приложения на React.

В рамках данной работы будут затронуты следующие аспекты:

- 1) Менеджер NPM пакетов. Основные команды запуска скриптов и установки зависимостей.
- 2) Различные варианты старта React приложения. Варианты сборщиков(бандлеров) веб приложения.

## **Менеджер пакетов NPM**

NPM - базовый менеджер пакетов, который входит в состав node js. Основными его задачами являются:

- 1) Менеджмент и установка зависимостей/библиотек, использованных в проект
- 2) Удобный запуск различных скриптов разработки, сборки, тестирования проекта.
- 3) Публикация пакетов в npm registry.

В качестве альтернатив NPM можно рассмотреть и другие менеджеры, которые работают гораздо быстрее и в некоторых аспектах предоставляют более удобный функционал. Например: yarn или pnpm.

## **Создание проекта на React**

Существует множество способов создать проект на React. Но можно выделить 2 пути:

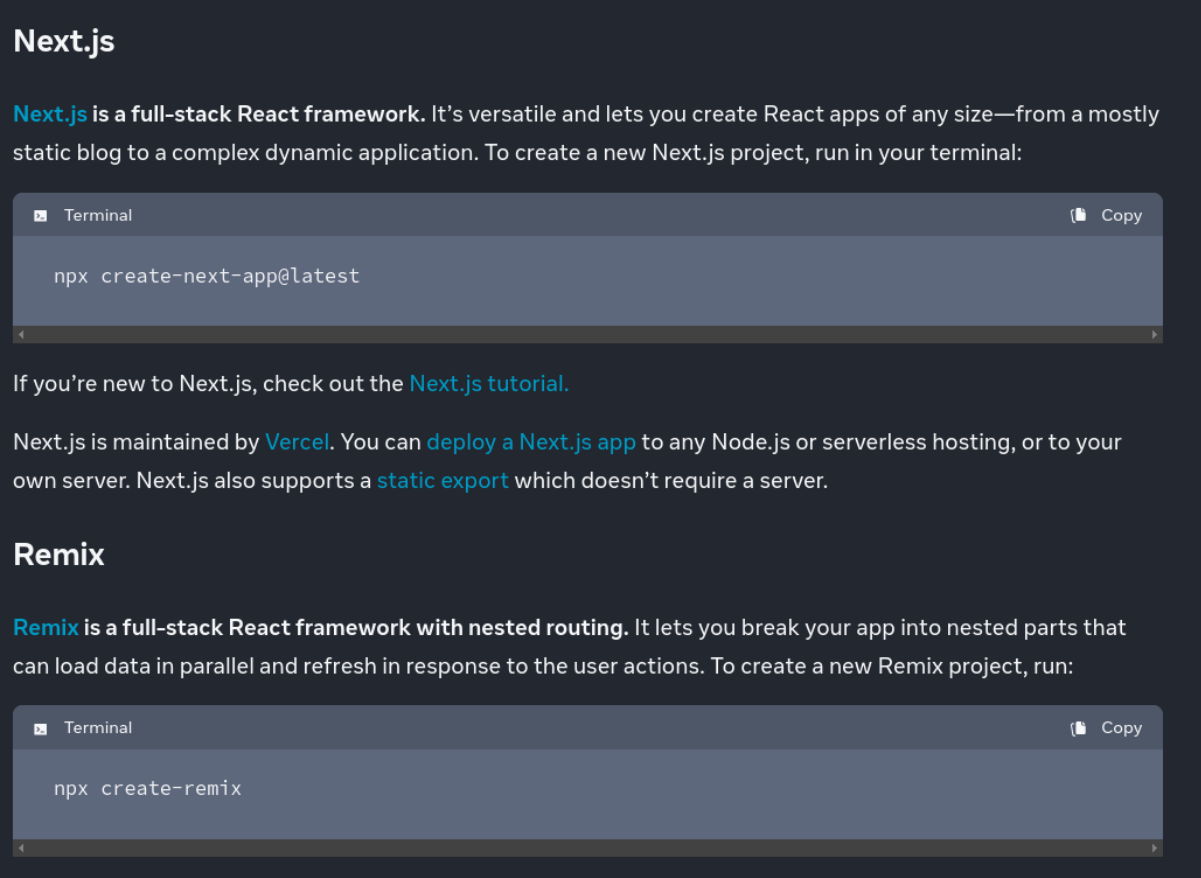
- 1) Полная ручная настройка проекта и бандлера.
- 2) Использовать готовые решения, которые предоставляют шаблоны.

## **Ручная настройка:**

Для начала нужно инициализировать проект, используя команду “npm init”. Далее устанавливаем сам React: “npm install react react-dom”. Выбор сборщика огромен: webpack, gulp, rollup и тд. Говорить о каждом можно очень долго, поэтому рассмотрим наиболее простой вариант с использованием готового шаблона.

## Готовые решения

Официальная документация React предлагает использовать либо next js либо remix.



**Next.js**

Next.js is a full-stack React framework. It's versatile and lets you create React apps of any size—from a mostly static blog to a complex dynamic application. To create a new Next.js project, run in your terminal:

```
npx create-next-app@latest
```

If you're new to Next.js, check out the [Next.js tutorial](#).

Next.js is maintained by [Vercel](#). You can [deploy a Next.js app](#) to any Node.js or serverless hosting, or to your own server. Next.js also supports a [static export](#) which doesn't require a server.

**Remix**

Remix is a full-stack React framework with nested routing. It lets you break your app into nested parts that can load data in parallel and refresh in response to the user actions. To create a new Remix project, run:

```
npx create-remix
```

Но оба варианта устанавливают дополнительные зависимости, что как мне не очень хорошо для базового ознакомления с React. Поэтому рассмотрим вариант установки “голового” React.

Воспользуемся Vite для создания проекта. Vite - набор инструментов для создания и управления разработки проекта. Под капотом использует Rollup для сборки проекта. В базовой версии не тянет никаких лишних зависимостей, которые бы влияли на React.

Просто пропишем “npm create vite@latest”, введем название проекта, выберем React в качестве библиотеки и ванильный вариант на чистом JS.

```
astukalov:~/Documents$ npm create vite@latest
✓ Project name: ... temp=react
✓ Package name: ... temp-react
✓ Select a framework: > React
✓ Select a variant: > JavaScript

Scaffolding project in /home/astukalov/Documents/temp=react...

Done. Now run:

  cd temp=react
  npm install
  npm run dev
```

Как можем видеть проект получился максимально компактным. Были добавлены лишь линтеры для более удобной разработки.

```
GNU nano 6.4                               ./tem
{
  "name": "temp-react",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-warnings 0",
    "preview": "vite preview"
  },
  "dependencies": {
    "react": "^18.2.0",
    "react-dom": "^18.2.0"
  },
  "devDependencies": {
    "@types/react": "^18.2.15",
    "@types/react-dom": "^18.2.7",
    "@vitejs/plugin-react": "^4.0.3",
    "eslint": "^8.45.0",
    "eslint-plugin-react": "^7.32.2",
    "eslint-plugin-react-hooks": "^4.6.0",
    "eslint-plugin-react-refresh": "^0.4.3",
    "vite": "^4.4.5"
  }
}
```

Перейдем в папку проекта и установим зависимости: “npm install”. Теперь запустим проект командой “npm run dev”. Проект готов к разработке.



# Vite + React

count is 0

Edit `src/App.jsx` and save to test HMR

Click on the Vite and React logos to learn more

## Вывод

В процессе выполнения работы были освоены навыки базового использования менеджера зависимостей `pnpm`, а также получены навыки создания базового React проекта.