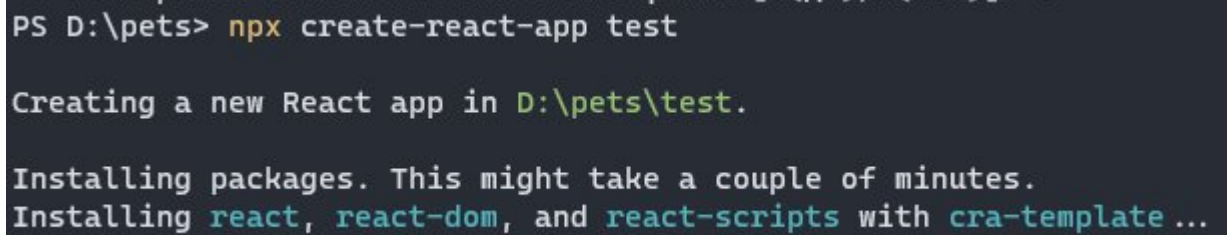


React — JavaScript-библиотека для создания SPA (Single Page Application) веб приложений, использующая для отображения интерфейсов язык разметки JSX, эффективно преобразовывающий JavaScript код в HTML элементы и манипулирующий DOM деревом.

Для начала работы с React необходимо установить стабильную версию Node.js на ваш рабочий компьютер. После этого необходимо создать новое React приложение при помощи команды `npx create-react-app <Название приложения>`. Утилита командной строки самостоятельно установит все необходимые пакеты для начала работы.

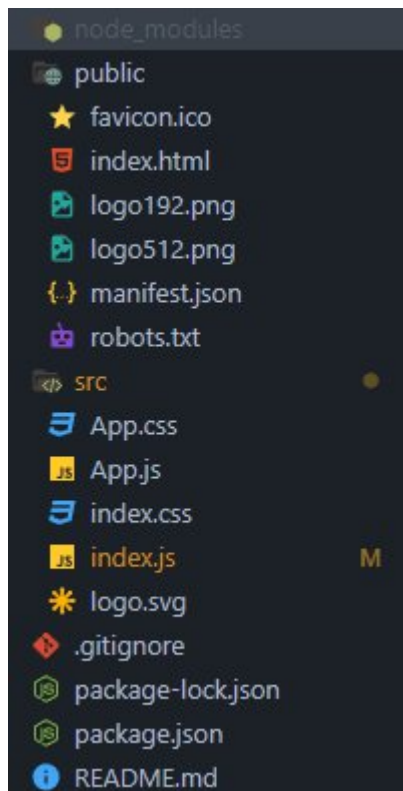


```
PS D:\pets> npx create-react-app test

Creating a new React app in D:\pets\test.

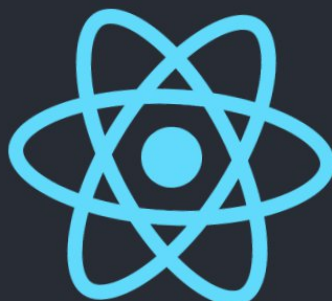
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template ...
```

После установки в директории появится папка вашего нового проекта. Она состоит из основных директорий: `node_modules` — папка node пакетов, `public` — папка для статических файлов, `src` — папка, в которой хранится основной код React приложения. В ней же происходит основной процесс разработки приложения.



При создании приложения создаётся базовый компонент страницы, представленный ниже. Для локального запуска приложения используется команда **npm start**.

```
1  import logo from './logo.svg';
2  import './App.css';
3
4  function App() {
5    return (
6      <div className="App">
7        <header className="App-header">
8          <img src={logo} className="App-logo" alt="logo" />
9          <p>
10             Edit <code>src/App.js</code> and save to reload.
11          </p>
12          <a
13            className="App-link"
14            href="https://reactjs.org"
15            target="_blank"
16            rel="noopener noreferrer"
17          >
18            Learn React
19          </a>
20        </header>
21      </div>
22    );
23  }
24
25  export default App;
```



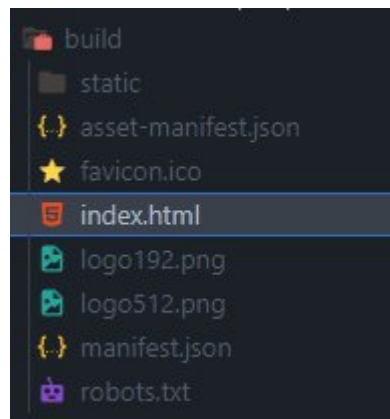
Edit `src/App.js` and save to reload.

[Learn React](#)

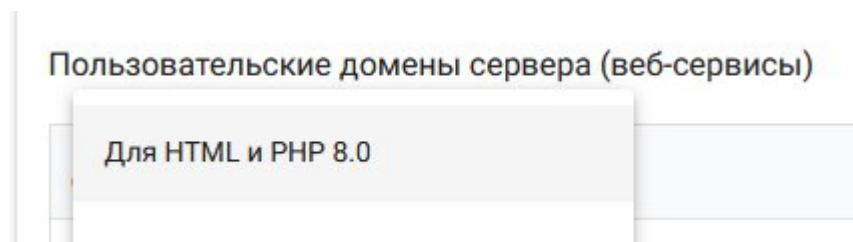
Для того, чтобы собрать приложение и разместить его на хостинге, необходимо выполнить команду **npm run build** и дождаться окончания процесса сборки. В результате будет сгенерирована папка `build`, в которой будут содержаться все файлы, необходимые для работы приложения (все файлы из директории `public`, а также сгенерированные статические js файлы в папке `static`, уже подключённые к html странице).

```
"scripts": {  
  "start": "react-scripts start",  
  "build": "react-scripts build",  
  "test": "react-scripts test",  
  "eject": "react-scripts eject"  
},  
"devDependencies": {
```

```
PS D:\pets\test> npm run build
```



После сборки приложения папку `build` можно переместить на сервер и создать новый домен, используя пункт «Для HTML и PHP 8.0»



Добавление домена

Имя домена без точек:

домен

test

.std-1083.ist.mospolytech.ru

Папка приложения (проекта) на сервере:

путь

/home/std/build

Единый скрипт, на который направлять все HTTP-запросы (необязательно, для фреймворков):

/home/std/build/путь к единому скрипту-обработчику HTTP-запросов

Путь к папке со статическими файлами (не обязательно):

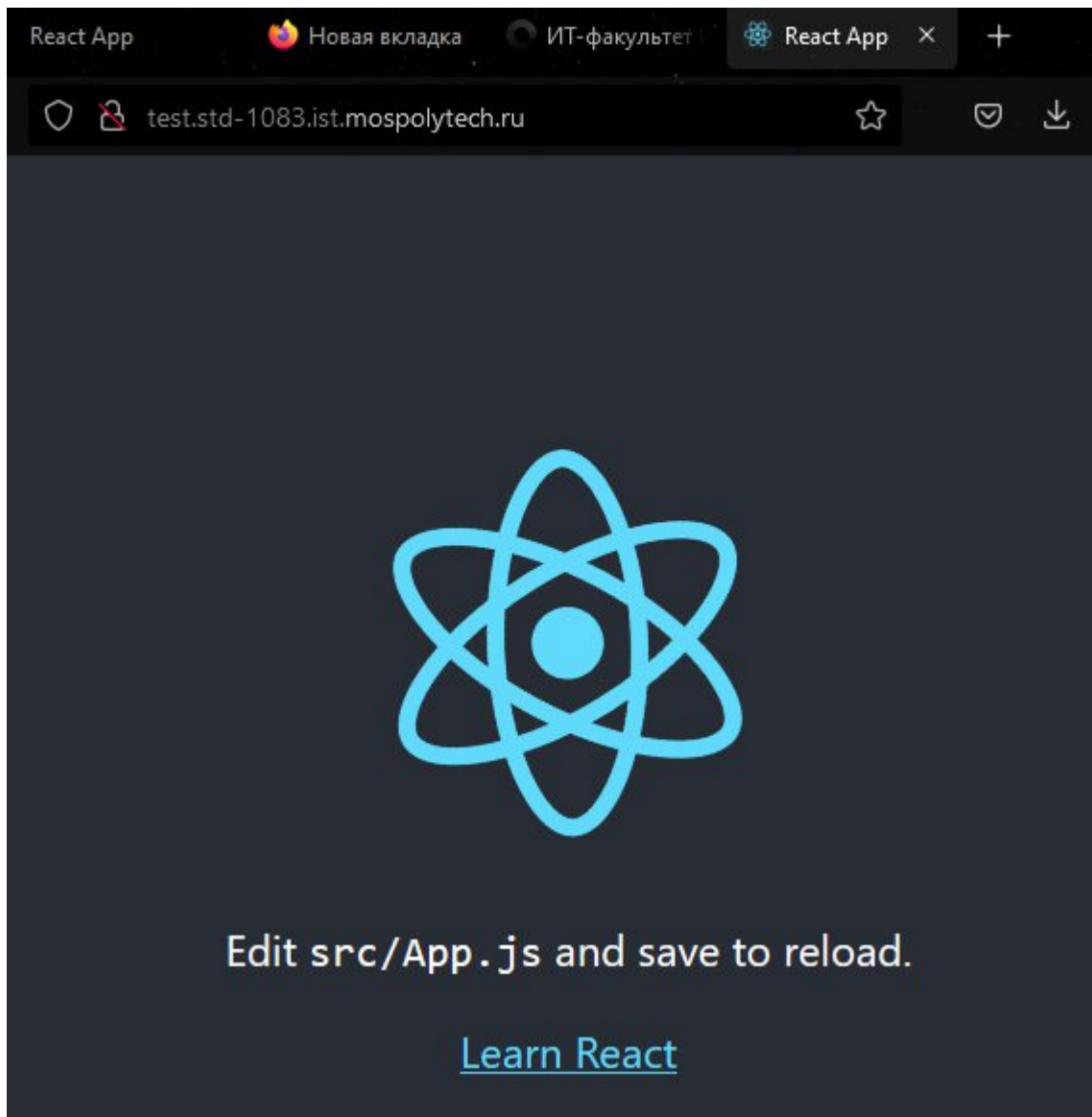
/home/std/путь к папке со статикой

URL папки со статическими файлами (не обязательно):

test.ist.mospolytech.ru/URI папки со статикой

Отмена

Добавить



Ваше React приложение размещено на хостинге fit.mospolytech и готово к работе.

К плюсам данного веб-фреймворка можно отнести модульность, возможность использовать множество различных библиотек, если какой-то из элементов сборки не удовлетворяет вашим потребностям. Экосистема React активно развивается и предоставляет набор из инструментов для хранения состояния и обмена данными (Redux, MobX, XState и т.д.), инструментов для организации маршрутизации на странице (React-Router), CSS и CSS-in-JS библиотек (StyledCSS, Tailwind, Emotion и др.), а также сотен библиотек готовых компонентов для ускорения разработки.

К минусам стоит отнести отсутствие единой модульной базы, в связи с чем стек технологий может значительно отличаться от проекта к проекту.

Отсутствие строгих требований к организации директорий может привести к затруднениям в разработке больших приложений, состоящих из десятков страниц и сотен компонентов. React также предлагает разные концептуально методы построения компонентов (классы и функции), что может затруднить понимание кода в незнакомой методологии.