

# Установка create-react-app

---

В этой лабораторной мы установим **create-react-app** (CRA) и познакомимся с файловой структурой приложения. CRA - это удобный инструмент для быстрого старта работы с React.js. Он легко устанавливается и позволяет начать работу с React.js в считанные минуты.

1. Убедитесь, что на вашей машине стоит [Node.js](#): нужно зайти в консоль и выполнить проверку версии

```
node --version  
npm --version
```

v17.7.1

8.5.2

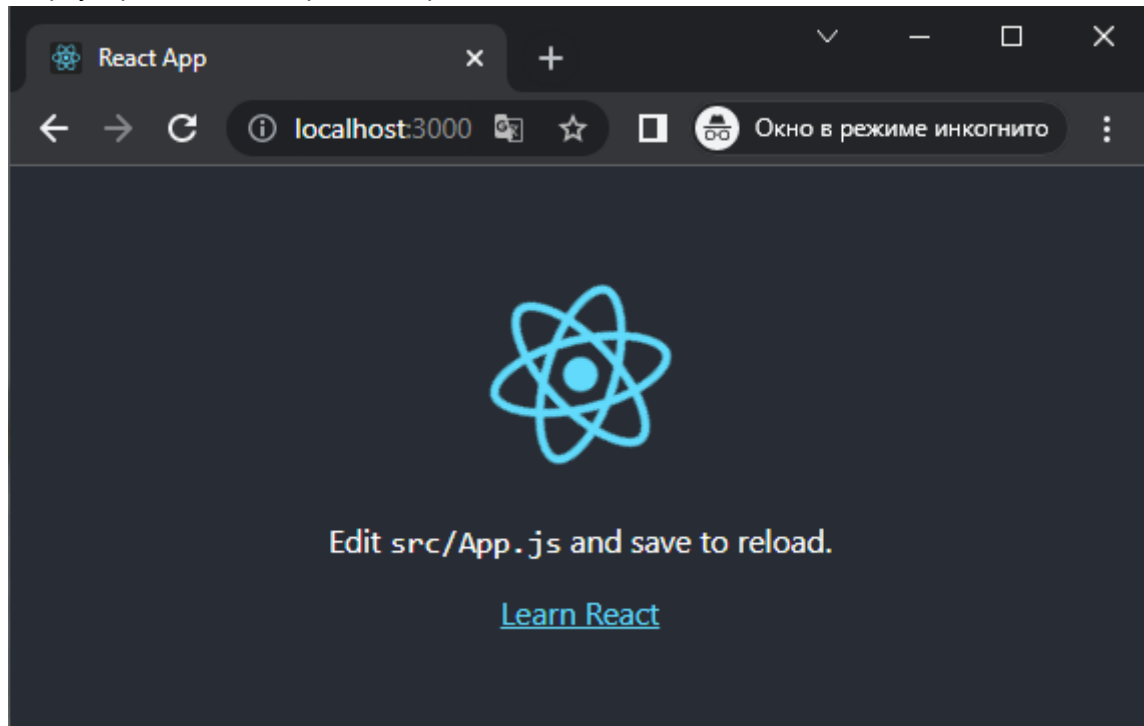
2. Установите инструмент

```
npx create-react-app my-app  
Need to install the following packages:  
  create-react-app  
Ok to proceed? (y) y
```

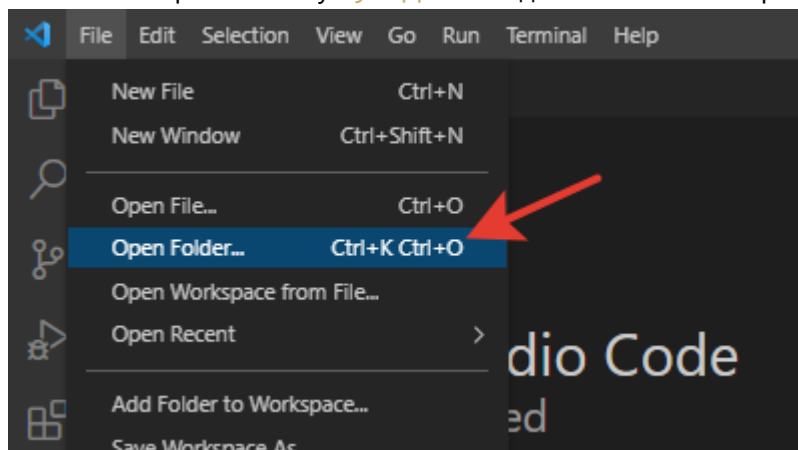
3. Перейдите в папку **my-app** и запустите приложение

```
cd my-app  
npm start
```

4. В браузере должна открыться страница подобная такой



5. В VSCode откройте папку `my-app` командой `Ctrl+O` или через пункт меню `File -> Open Folder`



6. Папка `my-app` будет выглядеть приблизительно так

```
my-app
├── README.md
├── node_modules
├── package.json
├── .gitignore
├── public
│   ├── favicon.ico
│   ├── index.html
│   ├── logo192.png
│   ├── logo512.png
│   ├── manifest.json
│   └── robots.txt
└── src
    ├── App.css
    ├── App.js
    ├── App.test.js
    └── index.css
```

```
|— index.js  
|— logo.svg  
|— serviceWorker.js  
|— setupTests.js
```

7. Откройте файл `index.js` - тут описывается подключение React и вызов отрисовки

```
import React from 'react';  
import ReactDOM from 'react-dom';  
import './index.css';  
import App from './App';  
import reportWebVitals from './reportWebVitals';  
  
ReactDOM.render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>,  
  document.getElementById('root')  
>);
```

8. Познакомьтесь с подсказками и убедитесь, что все понятно

- `import React from 'react';` - установка react
- `import './index.css';` - подключение CSS
- `import App from './App';` - подключение компонента `App` из текущей папки
- `ReactDOM.render()` - вызов отрисовки/рендера
- `<App />` - использование компонента `App` с синтаксисом `JSX`
- `document.getElementById('root')` - место где будем отрисовывать компонент `App`

9. Откройте файл `App.css` - тут будут описаны стили

```
.App {  
  text-align: center;  
}  
  
.App-logo {  
  height: 40vmin;  
  pointer-events: none;  
}  
  
...
```

10. Замените существующие стили в `App.css` следующими

```
* {  
  box-sizing: border-box;
```

```
font-family: sans-serif;
}
```

11. Откройте файл `App.js` - тут описывается компонент `App`

```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}

export default App;
```

12. Познакомьтесь с подсказками и убедитесь, что все понятно - `import './App.css';` - импортируется стиль компонента - `function App() { ... }` - описывается функциональный компонент - `return (<div>...</div>)` - возвращается JSX - специальный синтаксис

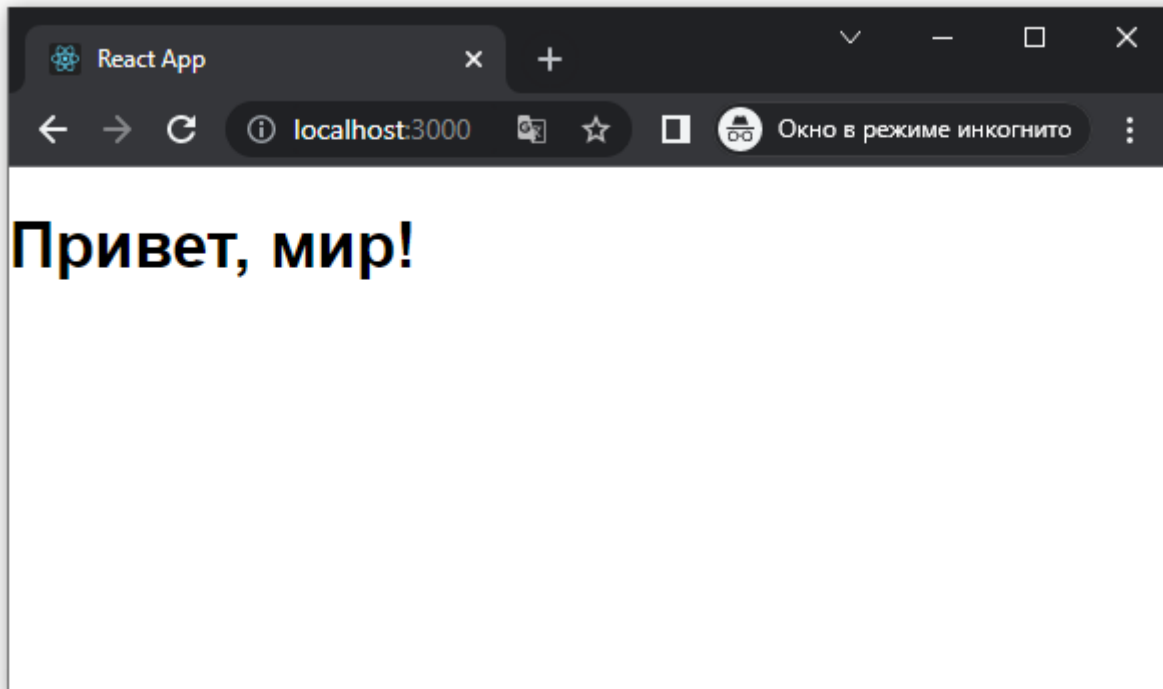
13. Измените содержимое компонента `App.js`

```
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <h1>Привет, мир!</h1>
      </header>
    </div>
  );
}

export default App;
```

14. Обновите страницу в браузере командой **Ctrl+F5**, должен быть результат похожий на следующий



15. Откройте файл **App.test.js** - тут описывается сам тест, измените его на следующее содержание

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders hello world heading', () => {
  render(<App />);
  const linkElement = screen.getByText(/Привет, мир!/i);
  expect(linkElement).toBeInTheDocument();
});
```

16. В консоли остановите приложение запуском **Ctrl+c**  
17. В консоли запустите тесты приложения

```
npm run test
```

18. Если появится предупреждение **Warning: ReactDOM.render is no longer supported in React 18. Use createRoot instead. Until you switch to the new API, your app will behave as if it's running React 17. Learn more: <https://reactjs.org/link/switch-to-createroot>**, то это означает что **CRA** использует версию меньшую чем React 18. Попробуйте **следовать инструкции**, либо **проигнорируйте**.  
19. Убедитесь, что тест выполнен

```
PASS   src/App.test.js
      ✓ renders hello world heading (115 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        3.048 s
Ran all test suites.
```

20. Для выхода нажмите **q** или два раза **Ctrl+c**

21. Попробуйте выполнить сборку приложения командой **npm run build**

```
C:\Users\administrator\Desktop\react.js\my-app>npm run build

> my-app@0.1.0 build
> react-scripts build

Creating an optimized production build...
Compiled successfully.

File sizes after gzip:

  46.05 kB   build\static\js\main.42483f1c.js
  1.78 kB   build\static\js\787.28cb0dcd.chunk.js
  284 B     build\static\css\main.4fbb31a4.css

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build

Find out more about deployment here:

  https://cra.link/deployment
```

22. Результатом сборки станет папка **build** с минимизированным HTML-файлом **index.html** и подпапка **static/js** с js-файлами внутри. Теперь можно попробовать выложить содержимое **build** на тестовый сервер (*не стоит делать это прямо сейчас*)

23. Измените содержание **index.js**

```
import React from 'react';
import { createRoot } from 'react-dom/client';
import './index.css';
```

```
import App from './App';
import reportWebVitals from './reportWebVitals';

const container = document.getElementById('root');
const root = createRoot(container);
root.render(<App />);
```

Теперь ваш код работает на свежем API отрисовки

#### 24. Выводы

- проверили версию `Node.js` и `npx`
- установили `CRA`
- познакомились с файловой структурой `CRA`
- познакомились с содержимым файлов приложения `index.html` и `index.js`
- вспомнили как в JS используются `import` и `export`
- познакомились с файлом `App.js` - он описывает функциональный компонент React
- столкнулись с синтаксисом `JSX`