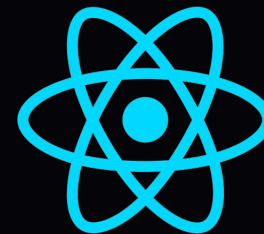


Знакомство с ReactJS. Первые компоненты

SPA. Webpack & Babel. Создание первого React-приложения.



Зачем изучать React?

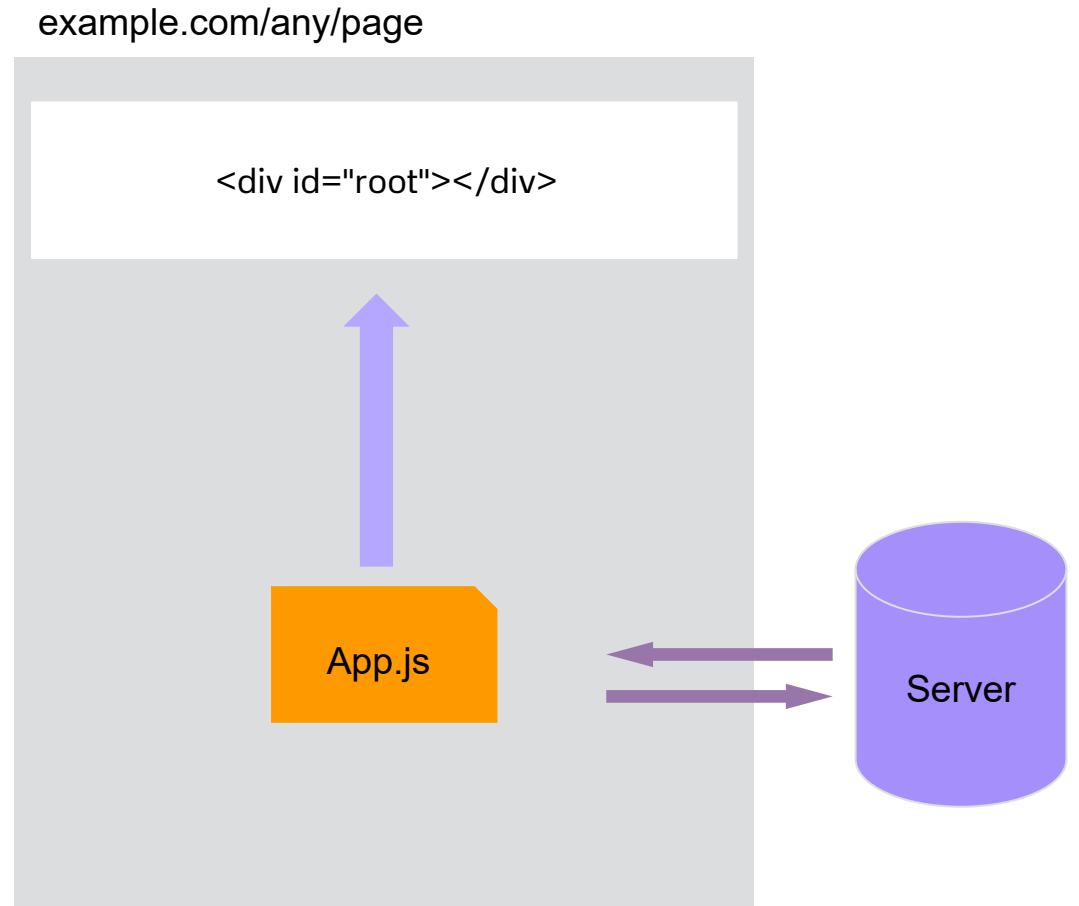
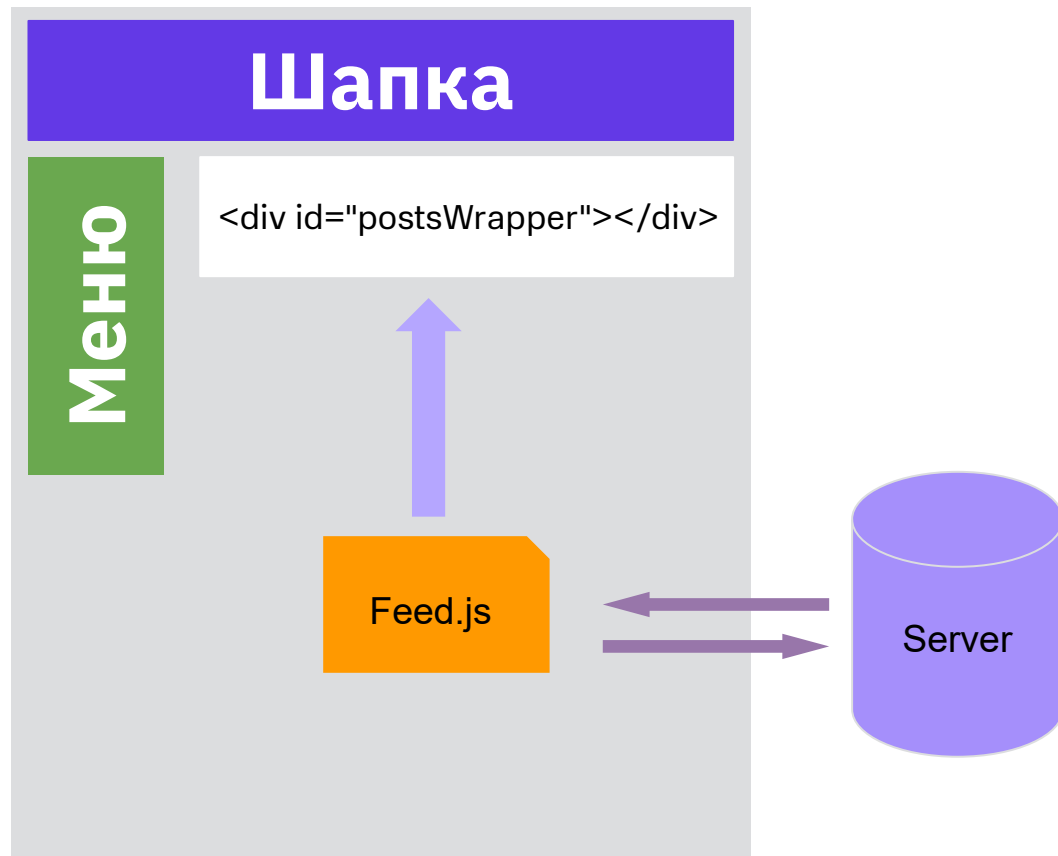
- Самый востребованный фреймворк в современной фронтенд-разработке.
- Кроссплатформенность: не только сайты, но и мобильные приложения.
- Аналоги: Vue и Angular.

Что будет на уроке?

- **Single Page Application**
- **Webpack**
- **Babel**
- **Первое приложение на React**



Стандартный подход vs SPA



Плюсы SPA

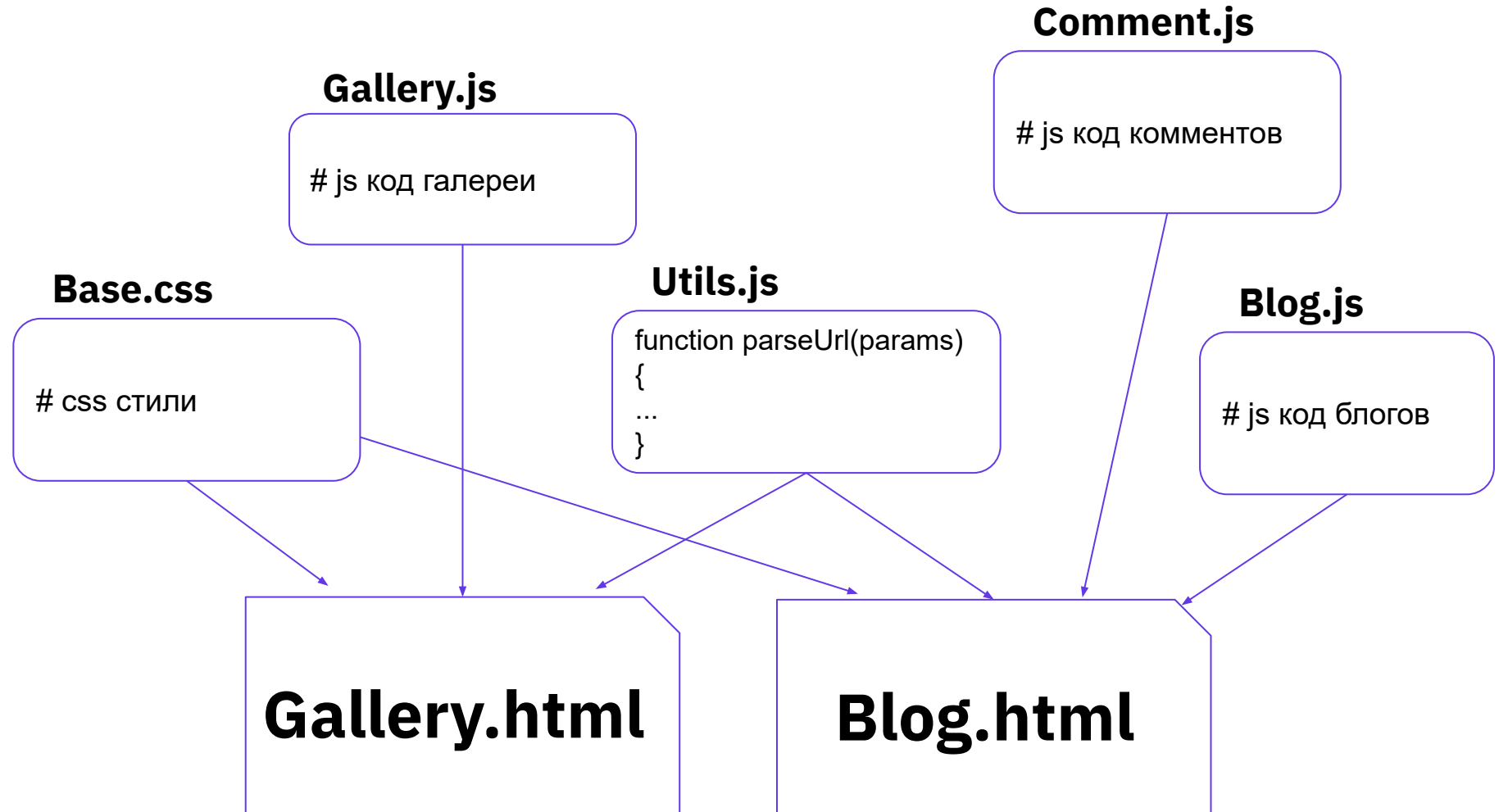
- Можно реализовать сложную логику
- Быстрая работа
- Экономия трафика
- Запросов на сервер больше, но они проще
- Простые способы реализовать кеширование

В чем подвох?

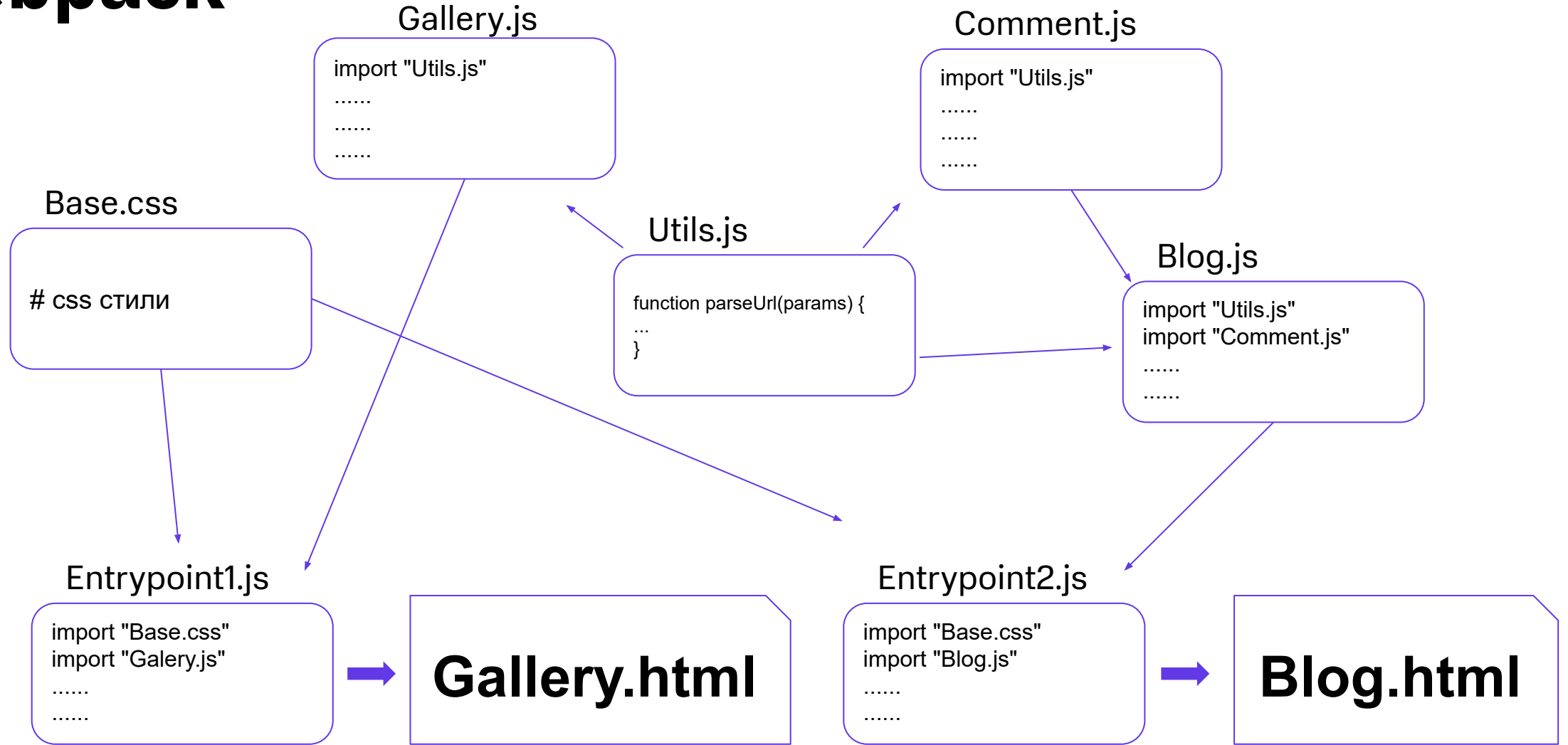
- Долгая первая загрузка
- Нужно хорошо знать не только HTML, но и JS
- Проблемы с SEO

Webpack & Babel

Обычный подход



Webpack



Плюсы webpack

- Генерация нескольких файлов (в том числе содержащих по несколько модулей сразу)
- Динамическая подгрузка файлов
- Масса плагинов для обработки кода
- Динамическое обновление кода во время его работы без потери состояния программы

Аналоги webpack

- Gulp
- Grunt
- Parcel
- Browserify

Babel

```
let a = () => {};  
let b = (c) => c;
```



```
var a = function () {};  
var b = function (c) {  
  return c;  
};
```

Create-react-app

Create-react-app (CRA) - библиотека, которая помогает быстро создать и настроить необходимые инструменты для начала разработки приложения на React. Она настраивает webpack, babel, jest (для запуска тестов) и пр.

прх

React. Пример кода

```
import React from 'react' ;
import ReactDOM from 'react-dom' ;

const element = React.createElement (
  'h1' ,
  { className : "element" },
  'Кажется, мы подключили React' ,
);

ReactDOM.render (
  element ,
  document.getElementById ( 'root' ) ,
);
```

JSX

```
import React from 'react' ;
import ReactDOM from 'react-dom' ;

const element = <h1 className ="element" >Кажется, мы подключили React </h1> ;

ReactDOM.render (
  element ,
  document.getElementById ( 'root' ),
);
```

create-react-app - App.js

```
1 import './App.css';
2
3 function App() {
4   return (
5     <div className="App">
6       <header className="App-header">
7         My First React App
8         <h3>Hello world!</h3>
9       </header>
10    </div>
11  );
12 }
13
14 export default App;
```


Функциональный и классовый компоненты

```
export function App() {  
  return (  
    <div className="App">  
      <header className="App-header">  
        My First React App  
        <h3>Hello world!</h3>  
      </header>  
    </div>  
  );  
}
```

```
export class App extends React.Component {  
  render() {  
    return (  
      <div className="App">  
        <header className="App-header">  
          My First React App  
          <h3>Hello world!</h3>  
        </header>  
      </div>  
    );  
  }  
}
```

Props

```
1 const myName = 'Alexander';
2
3 ReactDOM.render(
4   <React.StrictMode>
5     <App name={myName} />
6   </React.StrictMode>,
7   document.getElementById("root")
8 );
```

```
1 function App(props) {
2   return (
3     <div className="App">
4       <header className="App-header">
5         My First React App
6         <h3>Hello, {props.name}</h3>
7       </header>
8     </div>
9   );
10 }
```

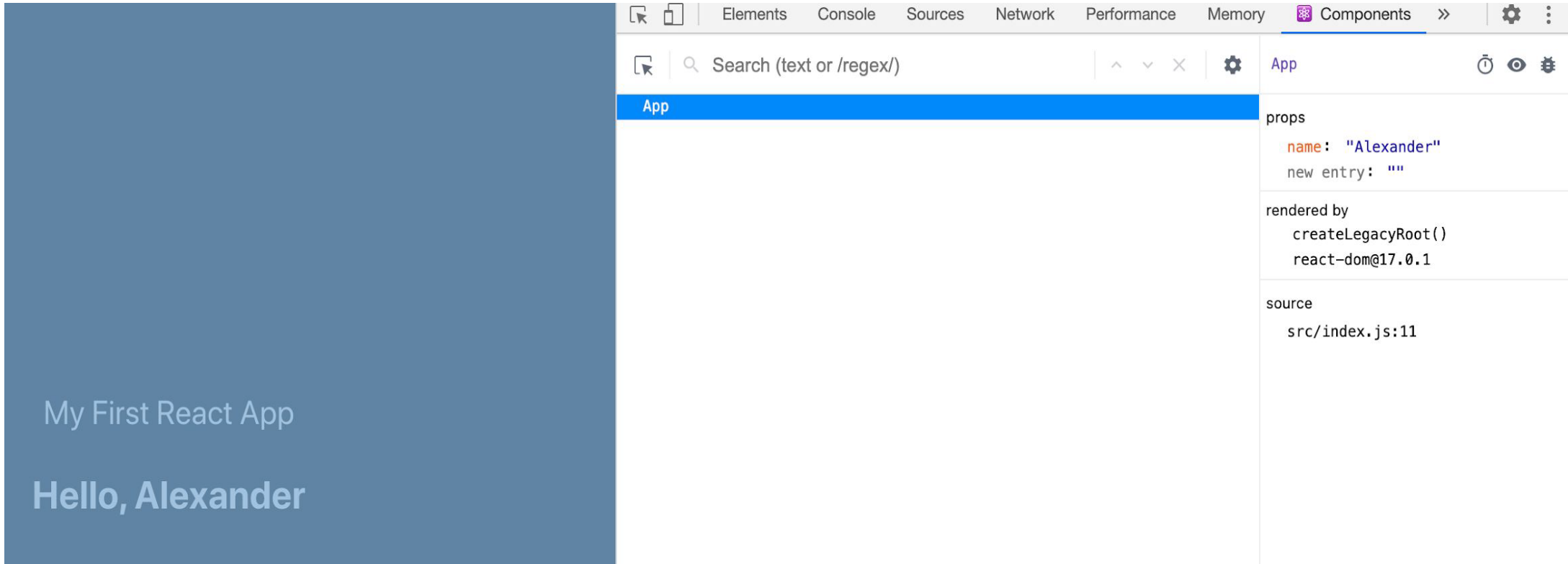
Стилизация КОМПОНЕНТОВ

1. inline styles (`style={{ marginTop: '20px' }}`)
2. className (css, sass, less)

Стилизация компонентов

```
1 import './App.css';
2
3 function App() {
4   return (
5     <div className="wrapper">
6       <div style={{paddingTop: '25px'}}>
7         Inline Styles
8       </div>
9     </div>
10  );
11 }
```

React devtools



The screenshot displays the React DevTools interface. On the left, a dark blue rectangular area represents the rendered application, containing the text "My First React App" and "Hello, Alexander". On the right, the Components panel is active, showing a tree view with a single component named "App" highlighted in blue. The right-hand pane provides details for the selected component:

- props**
 - name: "Alexander"
 - new entry: ""
- rendered by**
 - createLegacyRoot()
 - react-dom@17.0.1
- source**
 - src/index.js:11

Практическое задание

1. Развернуть новый проект с использованием create-react-app.
2. Создать компонент Message, отображающий переданный ему пропсом текст.
3. Изменить компонент App так, чтобы тот рендерил Message и передавал ему пропсом текст (константу).
4. Стилизовать компоненты через css (при желании можно использовать less или sass).
5. Установить расширение React Devtools.

Спасибо!

Задавайте вопросы

