# Сервер. Node.js

### Цель работы

Познакомить со способом подключения внешнего файла с React-компонентами.

### Задания для выполнения

1. Для работы также необходимо установить [Node.js](https://nodejs.org/en/) (https://nodejs.org/en/)
2. Создайте в одной папке три файла:

**index.html**

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8" />

    <title>Hello React</title>

</head>

<body>

    <div id="app"></div>

    <script crossorigin src="<https://unpkg.com/react@16/umd/react.production.min.js>"></script>

    <script crossorigin src="<https://unpkg.com/react-dom@16/umd/react-dom.production.min.js>"></script>

    <script src="<https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/6.25.0/babel.min.js>"></script>

    <script type="text/babel" src="index.jsx"></script>

</body>

</html>

**index.jsx**

class Hello extends React.Component {

    render() {

        return <h1>Hello, React</h1>;

    }

}

ReactDOM.render(

    <Hello></Hello>,

    document.getElementById("app")

)

**package.json** (это файл конфигурации проекта, который в данном случае определяет название проекта - "helloapp", его версию - 1.0.0. Секция "devDependencies" устанавливает зависимости, которые будут использоваться при разработке проекта. В частности, нам необходим мини-вебсервер для запуска веб-страницы, поэтому здесь определена зависимость "lite-server".)

{

    "name": "helloapp",

    "version": "1.0.0",

    "scripts": {

        "start": "lite-server"

    },

    "devDependencies": {

        "lite-server": "^2.2.1"

    }

}

1. Откройте командную строку/терминал и перейдите с помощью команды cd к папке где лежат эти три файла. Затем для установки зависимости lite-server выполним команду: npm install и затем запустите приложение с помощью команды: npm start.
2. Замените компонент на список дел (<https://reactjs.org/>), добавив оформление.
3. Загрузить созданную страницу на GitHub в репозиторий Student, используя формат в названии Фамилия (латинскими буквами)\_5.

### Методические указания

В итоге компилятор Babel при запуске страницы подгрузит файл index.jsx и скомпилирует из него код JavaScript, который будет применяться к странице.

### Контрольные вопросы

1. Можно ли использовать другой веб-сервер?

### Дополнительные задания

## Добавьте другие компоненты со страницы (<https://reactjs.org/>)