

## Практическая работа №1

### Создание простого сервера

**Цель:** Познакомится со средой выполнения Node JS, а также разработать простой сервер, способный принимать запросы пользователя.

#### Задачи:

1. Установить среду выполнения JS кода Node JS;
2. Реализовать простой сервер;
3. Разработать простой одностраничный сайт;
4. Объединить серверную и клиентские части приложения.

### Ход работы

#### Предисловие

Все материалы по данной работе вы можете найти вот тут: [GitHub](#)

В сообщении вы увидите еще одну инструкцию как запустить сервис скопировав файлы из репозитория. Если вам удобнее копировать оттуда, то не стесняйтесь клонировать его из источника 😊

#### Установка Node JS

Для достижения поставленной цели в первую очередь нам нужно установить среду выполнения Node JS. Для этого необходимо перейти на сайт проекта: <https://nodejs.org/en>. Для выполнения работы скачаем и установим версию 20.11.0 LTS

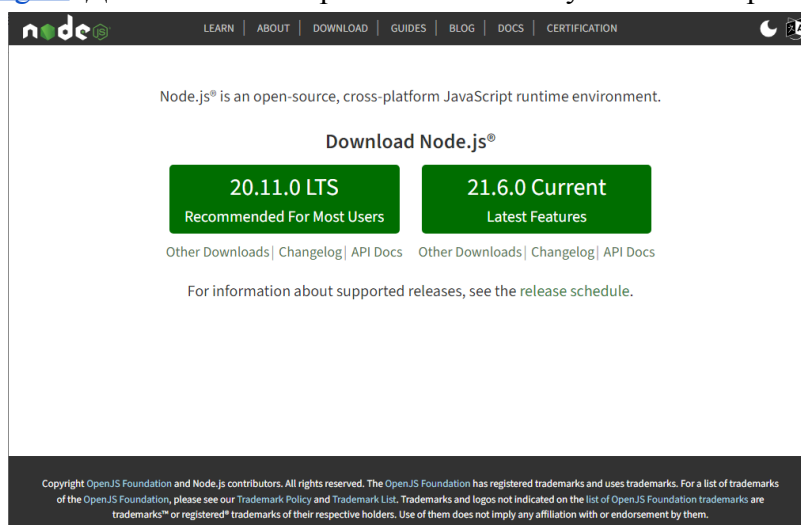


Рисунок 1. Главная страница проекта

В зависимости от вашей операционной системы процесс установки может отличаться. Вот ссылка на примеры установки:

<https://byshep.com/ru/articles/kak-ustanovit-node-js-i-npm-v-windows-macos-i-linux/>

После установки откройте командную строку и выполните команду, после чего у вас должен появиться примерно следующий результат:

```
node -v
```

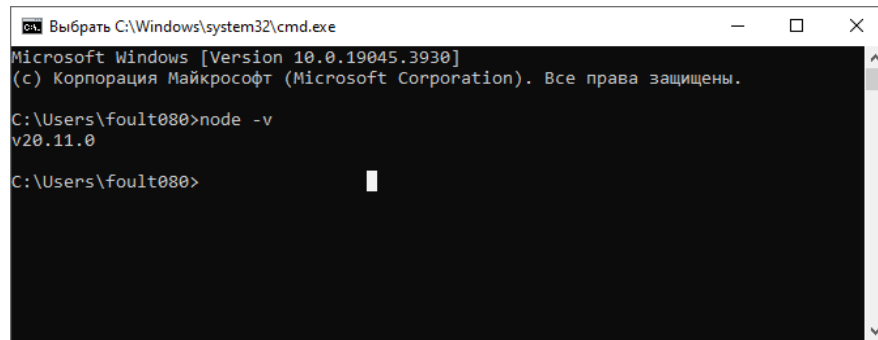


Рисунок 2. Результат установки Node JS.

### Создание простого сервера

Следующим шагом давайте создадим простой веб-сервер, который будет отвечать на запросы пользователя. Для этого создайте папку, после чего скопируйте путь до этой директории. В качестве примера я создал папку в каталоге Документы.

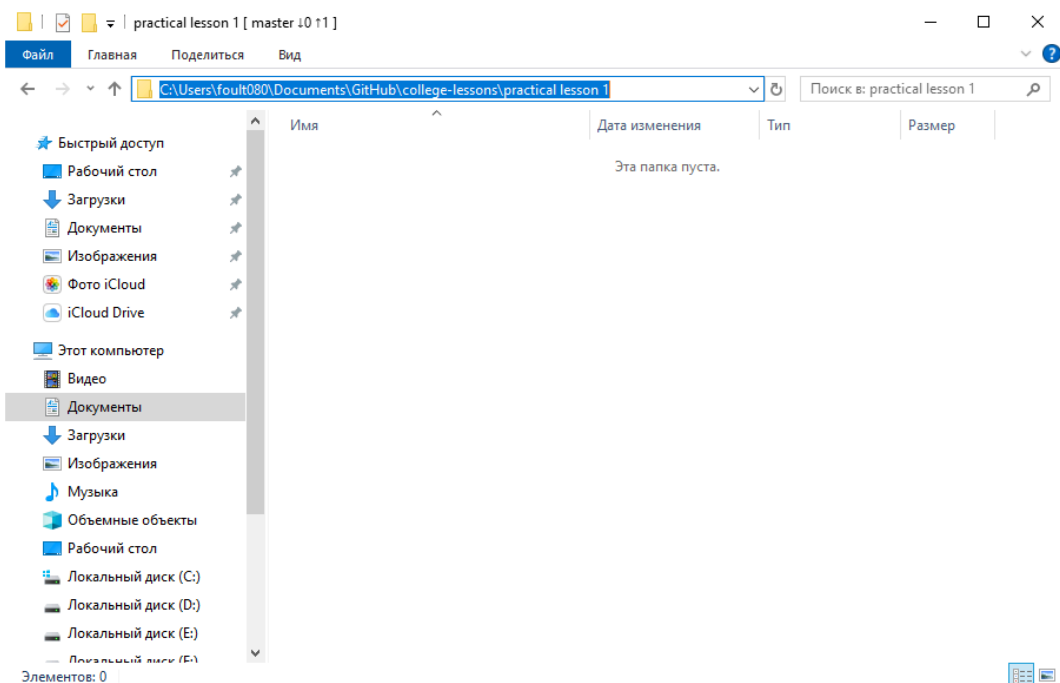


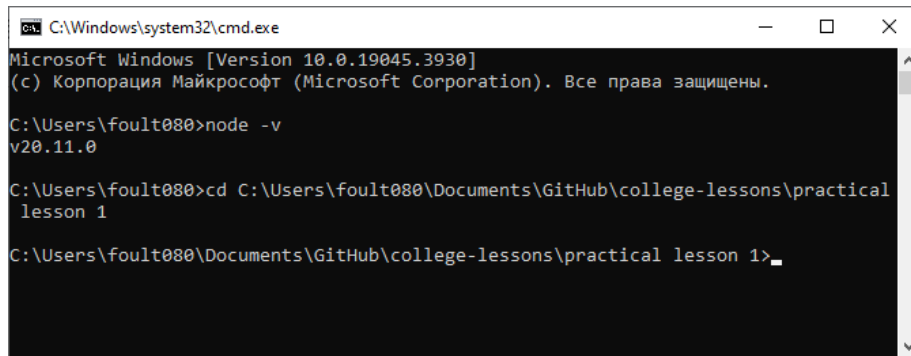
Рисунок 3. Создание пустой директории

Следующим шагом нам нужно внутри нашей консоли попасть в эту директорию. Для этого выполните следующую команду:

*cd \*Путь до директории\**

В моем случае путь будет следующим у вас он может отличаться:

*cd C:\Users\foult080\Documents\GitHub\college-lessons\practical lesson 1*



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.3930]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\foult080>node -v
v20.11.0

C:\Users\foult080>cd C:\Users\foult080\Documents\GitHub\college-lessons\practical lesson 1

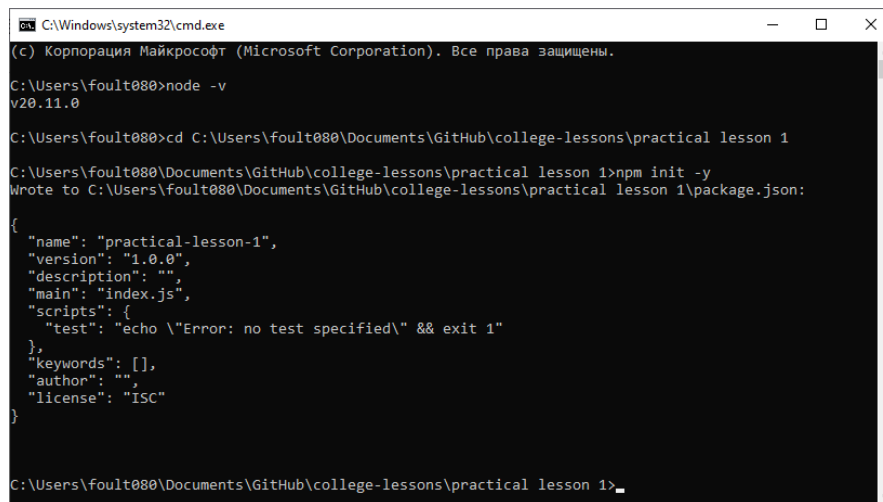
C:\Users\foult080\Documents\GitHub\college-lessons\practical lesson 1>
```

Рисунок 4. Результат перехода в директорию через консоль

Следующим шагом нам нужно проинициализировать наш новый проект. Для этого выполните следующую команду в консоли:

*npm init -y*

Данная команда создаст файл package.json внутри которого будут настройки и информация о нашем проекте.



```
C:\Windows\system32\cmd.exe
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\foult080>node -v
v20.11.0

C:\Users\foult080>cd C:\Users\foult080\Documents\GitHub\college-lessons\practical lesson 1

C:\Users\foult080\Documents\GitHub\college-lessons\practical lesson 1>npm init -y
Wrote to C:\Users\foult080\Documents\GitHub\college-lessons\practical lesson 1\package.json:

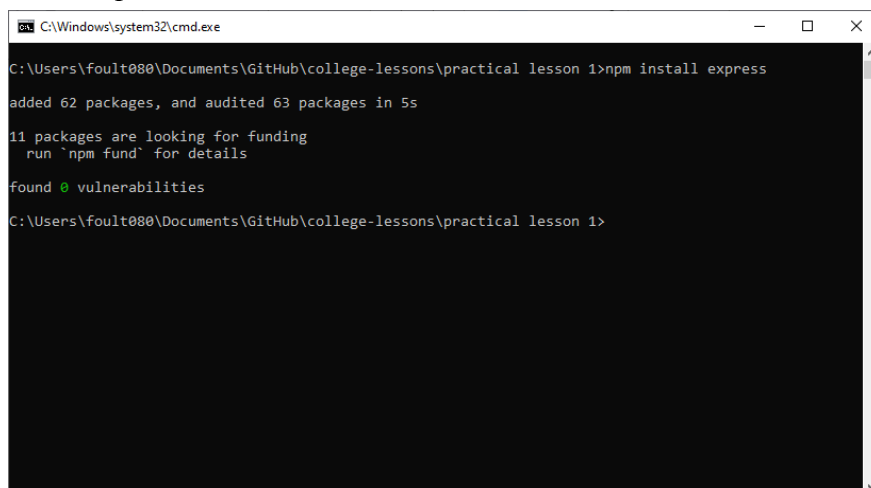
{
  "name": "practical-lesson-1",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

C:\Users\foult080\Documents\GitHub\college-lessons\practical lesson 1>
```

Рисунок 5. Результат инициализации пустого проекта.

Теперь давайте установим пакет для создания веб-сервера. Для этого выполните следующую команду в консоли:

*npm install express*



```
C:\Windows\system32\cmd.exe

C:\Users\foult080\Documents\GitHub\college-lessons\practical lesson 1>npm install express

added 62 packages, and audited 63 packages in 5s

11 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities

C:\Users\foult080\Documents\GitHub\college-lessons\practical lesson 1>
```

Рисунок 6. Результат установки Express

Следующим шагом давайте откроем нашу директорию в вашем любимом редакторе. Вы можете использовать любой редактор кода включая блокнот. Важно следить за наименованием файла и строками, которые вы добавляете в скрипт. Все новые строки будут подсвечены цветом ниже.

Давайте создадим файл с именем `app.js`. После чего поместим в него следующий код:

```
const path = require('path');
const express = require('express');
const app = express();

app.use(express.json({ extended: false }));

// проверка работы сервиса
app.get('/api/', async (req, res) => {
  console.log('Пришел GET запрос');
  return res.status(200).json({ msg: 'Привет Мир!' });
});

// отправка сообщения
app.post('/api/', async (req, res) => {
  const { name } = req.body;
  console.log(`Запрос прислал ${name}`);
  return res.status(201).json({ msg: `Привет ${name}!` });
});

// константа с портом
const PORT = 5000;

// server app
app.listen(PORT, () => console.log(`Сервер запущен на порту: ${PORT}`));
```

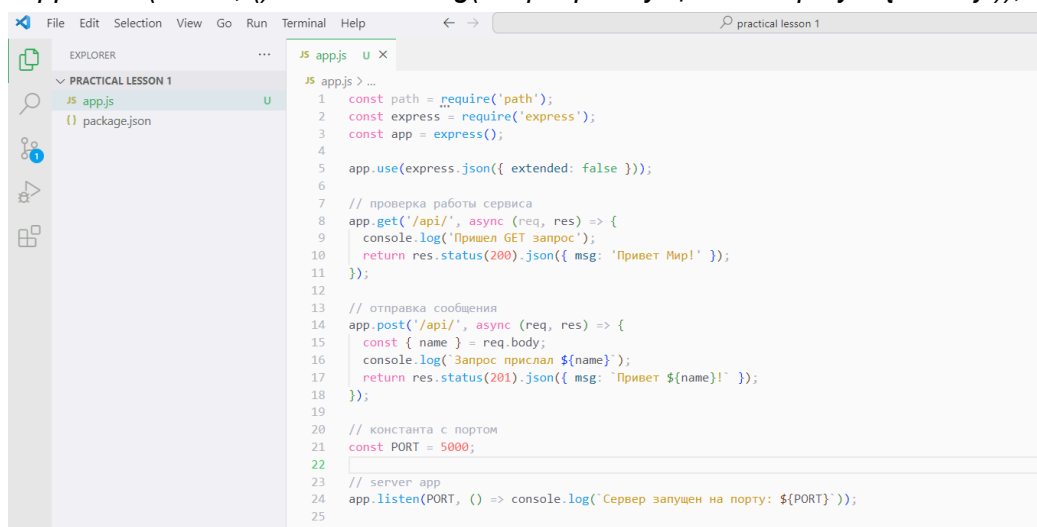
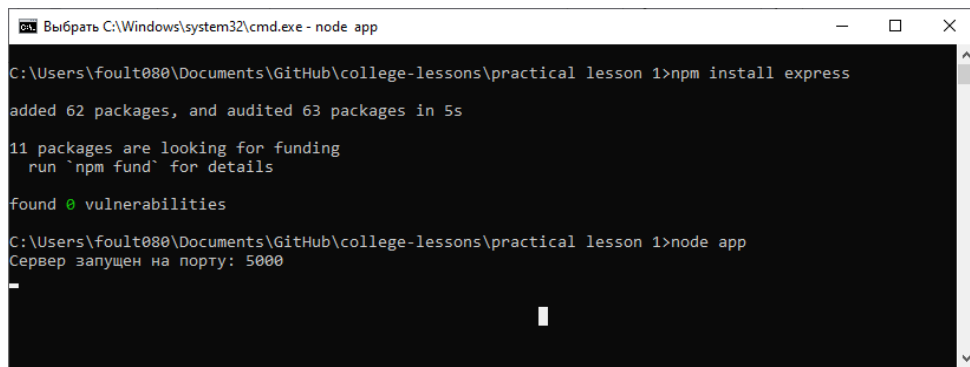


Рисунок 6. Скрипт в редакторе VS Code

Теперь давайте запустим его и попробуем отправить простой запрос. Для этого обязательно сохраните файл, после чего выполните команду в консоли:

*node app*



```
Выбрав C:\Windows\system32\cmd.exe - node app

C:\Users\foult080\Documents\GitHub\college-lessons\practical lesson 1>npm install express

added 62 packages, and audited 63 packages in 5s

11 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\foult080\Documents\GitHub\college-lessons\practical lesson 1>node app
Сервер запущен на порту: 5000
```

Рисунок 7. Результат запуска скрипта в консоли

Теперь не закрывая консоль откройте браузер и перейдите по адресу: <http://localhost:5000/api>

Вы должны увидеть сообщение как на рисунке 8.

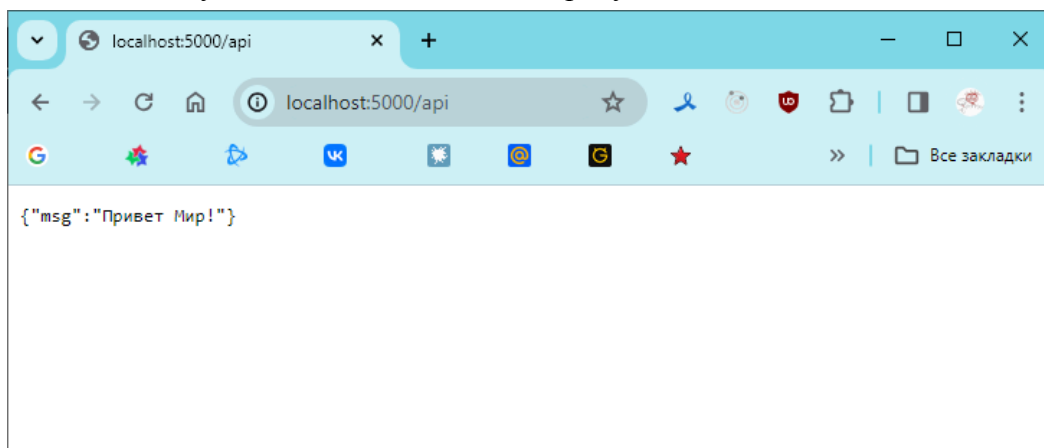


Рисунок 8. Ответ сервиса в браузере

А теперь давайте вернемся в нашу консоль и посмотрим что произошло там.

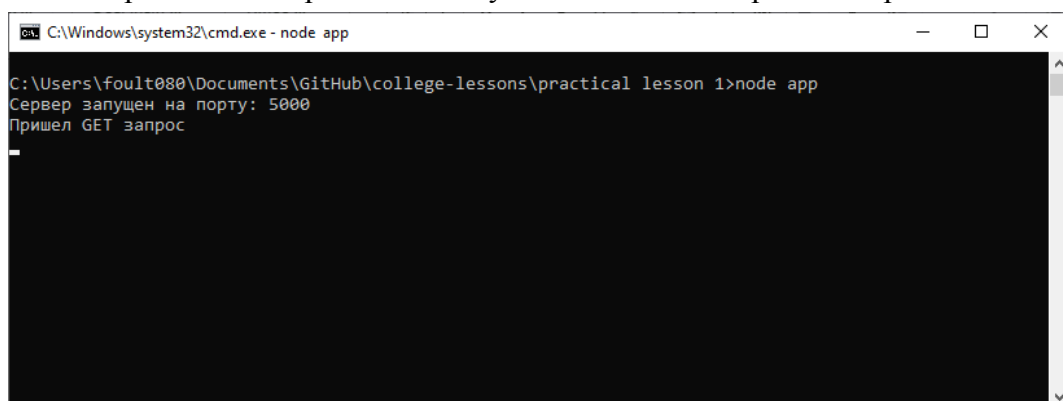


Рисунок 9. Результат командной строки.

Попробуйте обновить страницу несколько раз и посмотрите на то что происходит в консоли.

Давайте подведем итог того что у нас сейчас есть:

Мы установили среду выполнения Node JS, а также написали свой первый веб-сервер, который может отвечать нам простыми сообщениями. Давайте теперь создадим страничку, которая будет взаимодействовать с вашим сервером.

## Создание клиентской части

В нашей директории создайте папку public. После чего давайте поместим в нее ранее использованную верстку с “Приветствием”. Вы можете найти верстку тут: [GitHub](#) или скопировать отсюда:

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>First page</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
      rel="stylesheet"

integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUV
M"
      crossorigin="anonymous"
    />
    <link rel="stylesheet" type="text/css" href="./assets/style.css" />
  </head>
  <body>
    <div class="container my-5">
      <h1>Базовая страничка для знакомства с JavaScript</h1>

      <div class="col-lg-8 px-0" class="input-el">
        <h3>Взаимодействие с Input</h3>
        <h4>Как тебя зовут?</h4>
        <input
          id="user-name"
          type="text"
          class="form-control col-4"
          placeholder="Имя"
          aria-label="Имя пользователя"
          aria-describedby="addon-wrapping"
        />
        <button class="btn btn-success mt-2" onclick="sayHello()">Поприветствовать</button>
      </div>

      <hr />

    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"

integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz
"
      crossorigin="anonymous"
    ></script>
    <script src="./assets/main.js"></script>
  </body>
</html>
```



Отлично! Теперь вернемся к нашему app.js и добавим туда следующие команды:

```
app.use(express.static(path.join(__dirname, 'public')));
app.get('*', (req, res) => res.sendFile(path.resolve(__dirname, 'public', 'index.html')));
```

На рисунке 12 зеленым цветом подсвечены изменения.



Рисунок 12. Содержимое app.js

Теперь вернемся в консоль и остановим работу нашего сервера. Для этого нажмите сочетание клавиш CTRL + C.

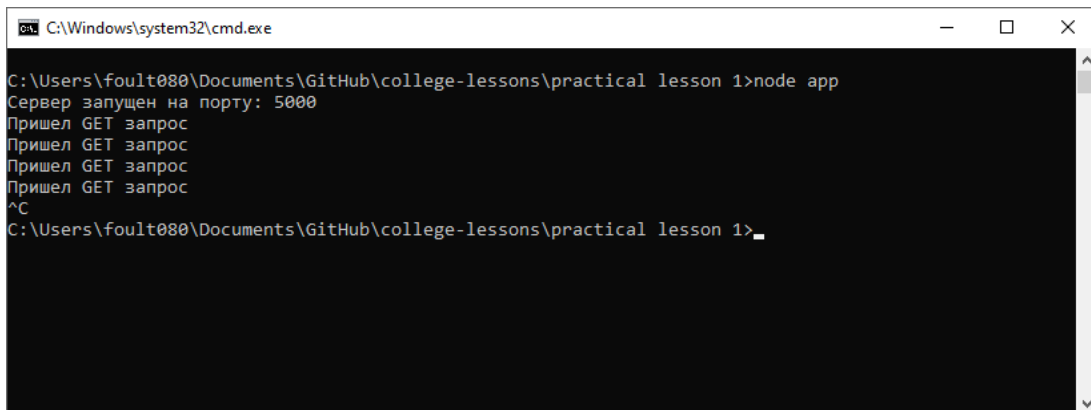


Рисунок 13. Остановка сервера

После чего повторно выполните команду:

```
node app
```

Если вы все выполнили верно, то при переходе по адресу: <http://localhost:5000/> вы должны увидеть следующее изображение:



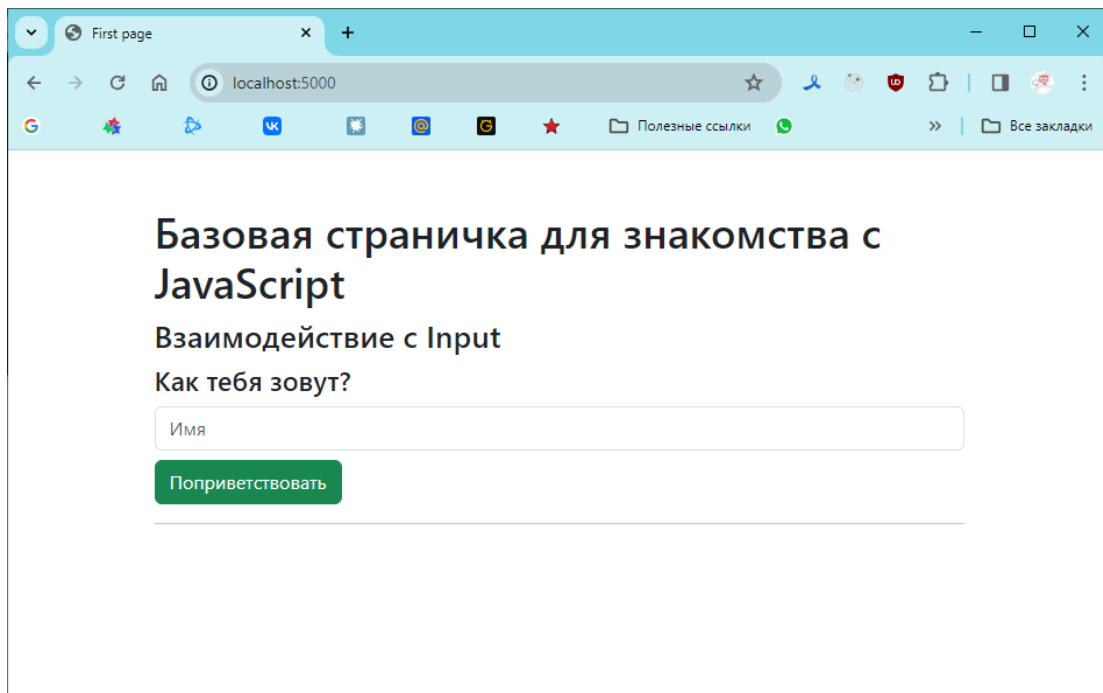


Рисунок 14. Ответ сервера

Попробуйте ввести имя и посмотрите на сообщение во всплывающем окне.

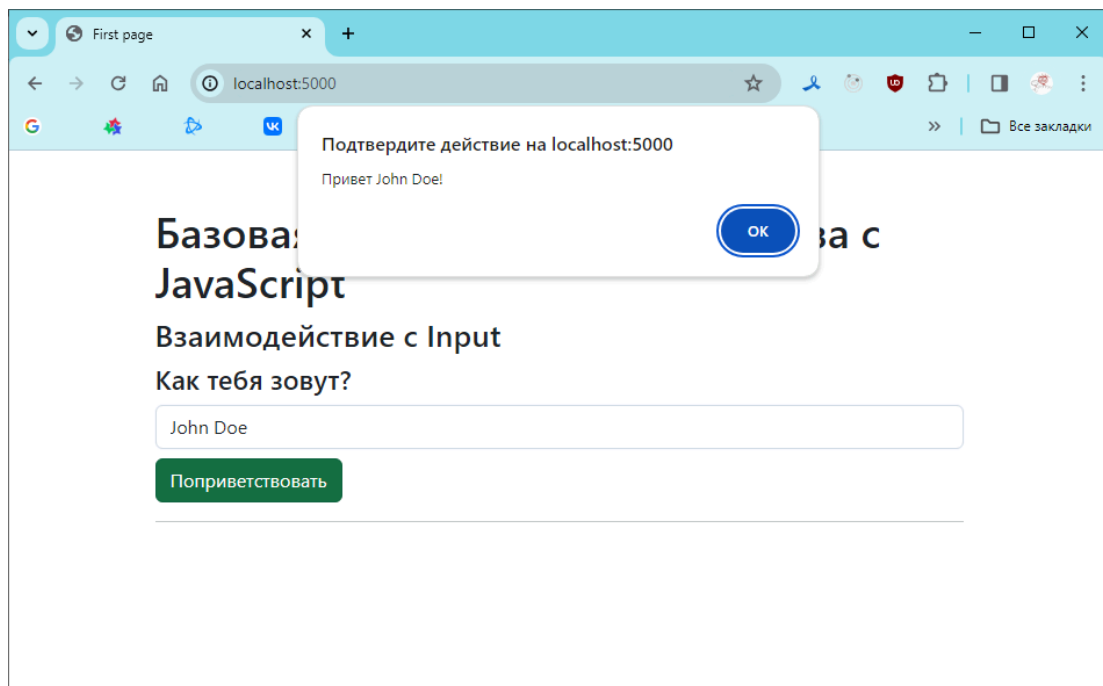


Рисунок 15. Отправка запроса с именем пользователя.

Поздравляю!

Вы только что сделали свой первый простой сервис, который отправляет HTTP запрос на сервер и получает ответ от него.

Если в ходе выполнения у вас что-то не получилось, то перейдите по вот этой ссылке [GitHub](#) и сравните ваши файлы с исходником. В случае если это не помогло, то напишите об этом в Телеграмм канале.