
POST-запити та відправлення форм

Під час оправлення якихось складних даних зазвичай використовуються форми. Розглянемо, як отримувати надіслані дані в Express.

Спочатку визначимо у папці проекту новий файл **index.html** із формою введення даних:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Регистрация</title>
5   <meta charset="utf-8" />
6 </head>
7 <body>
8   <h1>Введите данные</h1>
9   <form method="post">
10    <label>Имя</label><br>
11    <input type="text" name="userName" /><br><br>
12    <label>Возраст</label><br>
13    <input type="number" name="userAge" /><br><br>
14    <input type="submit" value="Отправит" />
15  </form>
16 </body>
17 </html>
```

Тут визначено два поля для введення імені та віку користувача.

У файлі **app.js** визначимо наступний код:

```
1 const express = require("express");
2
3 const app = express();
4
5 // создаем парсер для данных application/x-www-form-urlencoded
6 const urlencodedParser = express.urlencoded({extended: false});
7
8 app.get("/", function (request, response) {
9   response.sendFile(__dirname + "/index.html");
```

```
10 });
11 app.post("/", urlencodedParser, function (request, response) {
12     if(!request.body) return response.sendStatus(400);
13     console.log(request.body);
14     response.send(`${request.body.userName} - ${request.body.userAge}`);
15 });
16
17 app.listen(3000, ()=>console.log("Сервер запущен..."));
```

Насамперед для отримання відправлених даних необхідно створити парсер:

```
1 const urlencodedParser = express.urlencoded({extended: false});
```

Поскольку данные отправляются с помощью формы, то для создания парсера применяется функция **urlencoded()**. В эту функцию передается объект, устанавливающий параметры парсинга. Значение `extended: false` указывает, что объект - результат парсинга будет представлять набор пар ключ-значение, а каждое значение может быть представлено в виде строки или массива.

При переходе по адресу "/" будет срабатывать метод `app.get`, который отправит пользователю файл `index.html`.

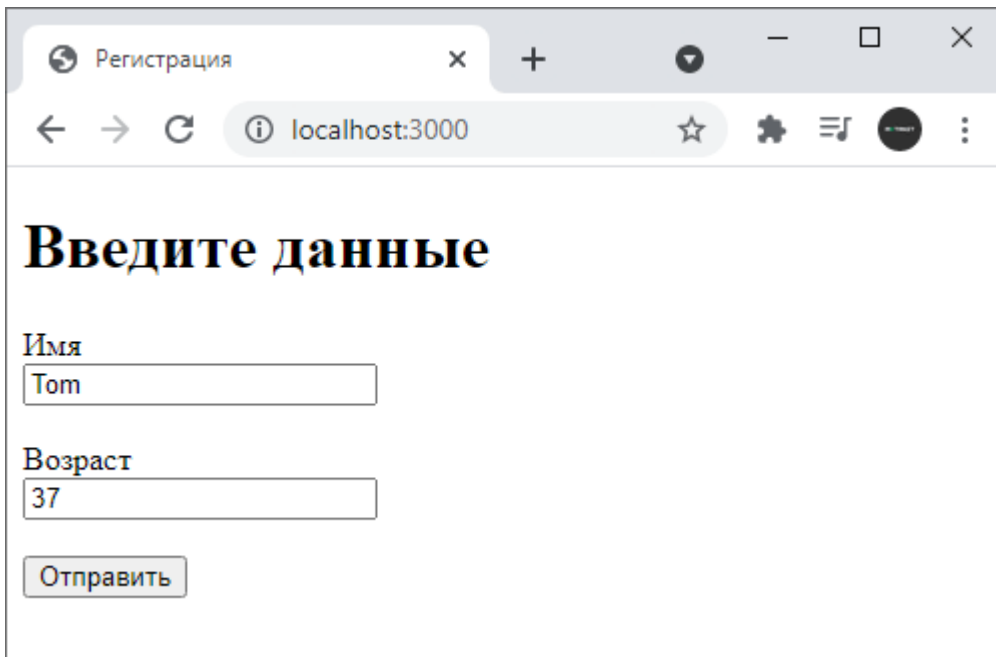
Так как данные отправляются с помощью метода POST, то для обработки определяем функцию **app.post("/",...)**. Первый параметр функции - адрес, на который идет отправка - "/". В коде формы в файле `index.html` мы не указали адрес для отправки, поэтому по умолчанию данные будут отправляться по тому же адресу, по которому доступна веб-страница `index.html` - "/". Таким образом, с одним адресом "/" связаны две функции, только одна обрабатывает запросы `get`, а другая - запросы `post`. Второй параметр - выше созданный парсер. Третий параметр - обработчик:

```
1 app.post("/", urlencodedParser, function (request, response) {
2     if(!request.body) return response.sendStatus(400);
3     console.log(request.body);
4     response.send(`${request.body.userName} - ${request.body.userAge}`);
5 });
```

Для отримання самих відправлених даних використовуємо вирази типу **request.body.userName**, де `request.body` інкапсулює дані форми, а `userName` - ключ даних, який відповідає значенню атрибута `name` поля введення на сторінці `html`:

```
1 <input type="text" name="userName" />
```

Запустимо програму та звернемося до сторінки `index.html`:



Регистрация

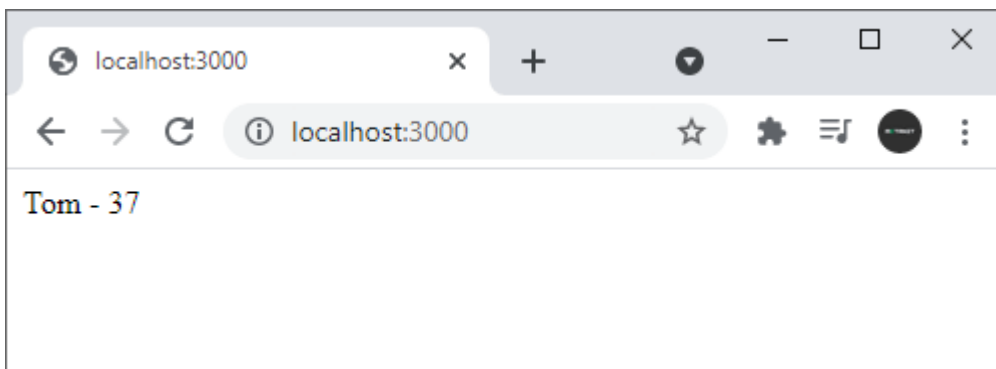
localhost:3000

Введите данные

Имя

Возраст

Після надсилання введені дані будуть отримані у функції `app.post("/...", ...)`, яка виведе результат обробки:



localhost:3000

localhost:3000

Том - 37

[Назад](#) [Зміст](#) [Вперед](#)

