

---

## Надсилання відповіді

Для надсилання відповіді в express у об'єкта response можна використовувати низку функцій. Найпоширеніший спосіб надсилання відповіді представляє функція **send()**. Як параметр ця функція може приймати об'єкт Buffer, рядок, у тому числі з html-кодом, об'єкт javascript або масив.

```
1 const express = require("express");
2 const app = express();
3
4 app.use(function (request, response) {
5     response.send("<h2>Hello</h2>");
6 });
7
8 app.listen(3000);
```

Надсилання об'єкта:

```
1 response.send({id:6, name: "Tom"});
```

Відправлення масиву:

```
1 response.send(["Tom", "Bob", "Sam"]);
```

Надсилання об'єкта:

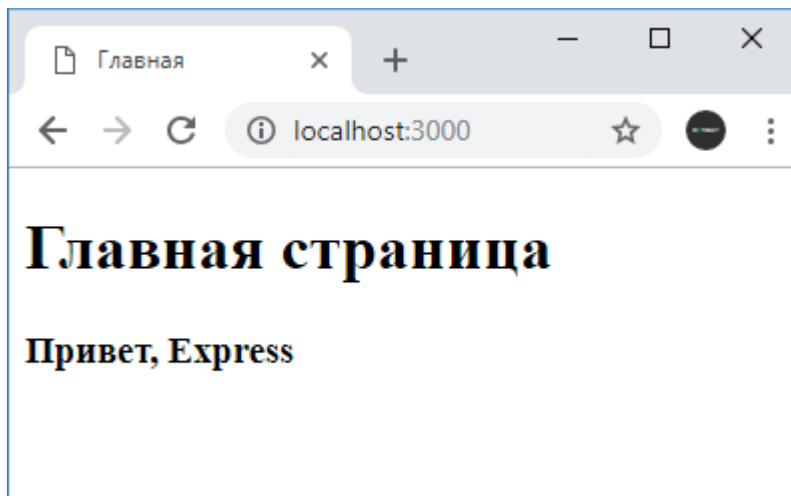
```
1 response.send(Buffer.from("Hello Express"));
```

Об'єкт Buffer формально представляє деякі бінарні дані. Так, у разі вище під час виконання коду за замовчуванням браузер завантажить файл, у якому буде рядок "Hello Express".

### sendFile

Метод send удобен для отправки строк, некоторого кода html небольшой длины, однако есть отправляемый код html довольно большой, то соответственно код приложения тоже становится громоздким. Например, мы можем написать так:

```
1 const express = require("express");
2 const app = express();
3
4 app.use(function (request, response) {
5   response.send(`<!DOCTYPE html>
6   <html>
7   <head>
8     <title>Главная</title>
9     <meta charset="utf-8" />
10  </head>
11  <body>
12    <h1>Главная страница</h1>
13    <h3>Привет, Express</h3>
14  </body>
15  <html>`);
16 });
17
18 app.listen(3000);
```



Однако гораздо лучше определять код html в отдельных файлах и затем эти файлы отправлять с помощью функции **sendFile()**.

Например, определим в папке проекта новый файл **index.html**:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Главная</title>
5     <meta charset="utf-8" />
6   </head>
7   <body>
8     <h1>Главная страница</h1>
9     <h3>Привет, Express</h3>
10  </body>
11 </html>
```

Отправим этот файл с помощью функции `sendFile`:

```
1 const express = require("express");
2 const app = express();
3
4 app.use(function (request, response) {
5   response.sendFile(__dirname + "/index.html");
6 });
7
8 app.listen(3000);
```

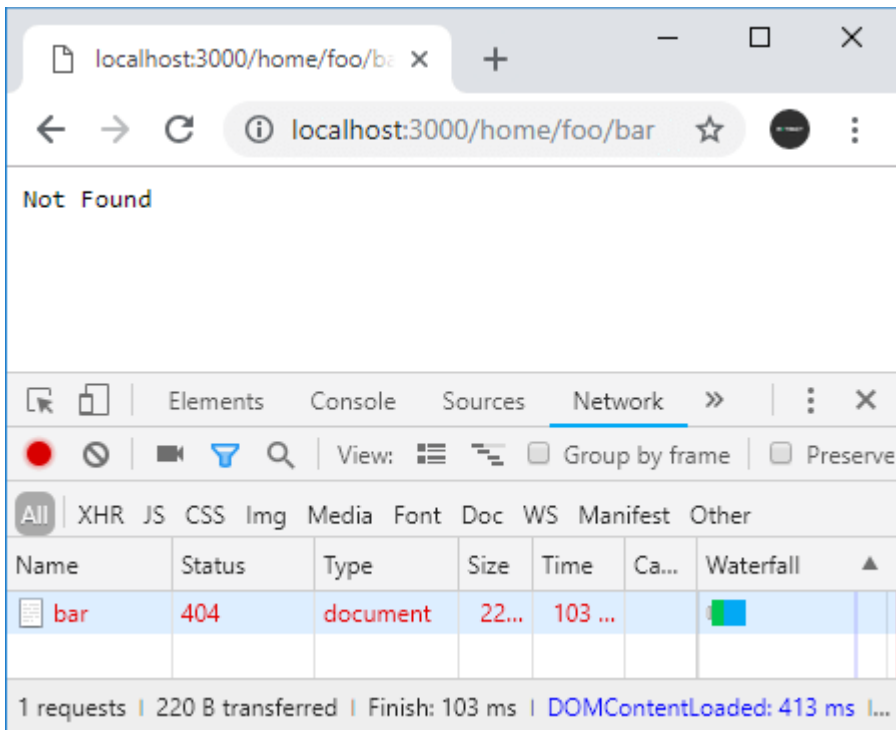
В итоге мы получим тот же самый результат.

Следует учитывать, что в функцию `sendFile` необходимо передавать абсолютный путь к файлу, именно для этого с помощью `__dirname` получаем абсолютный путь к текущему проекту и затем добавляем к нему путь к файлу в рамках текущего проекта.

## Отправка статусных кодов

Функція **`sendStatus()`** надсилає користувачеві певний статусний код з деяким стандартним повідомленням. Наприклад, відправимо статусний код 404, який каже, що ресурс не знайдено:

```
1 const express = require("express");
2 const app = express();
3
4 app.use("/home/foo/bar", function (request, response) {
5   response.sendStatus(404)
6 });
7
8 app.listen(3000);
```



Як видно зі скріншота, при надсиланні статусного коду 404 також надсилається повідомлення "Not Found". Але, можливо, ми захочемо надсилати якісь свої інформативніші повідомлення. У цьому випадку можна використовувати комбінацію функції **status()** , яка також надсилає статусний код, та функції **send()** :

```
1 const express = require("express");
2 const app = express();
3
4 app.use("/home/foo/bar", function (request, response) {
5   response.status(404).send(`Ресурс не найден`);
6 });
7
8 app.listen(3000);
```

[Назад](#). [Зміст](#). [Вперед](#)

