POST

SAKIB RASUL

AUGUST 4, 2023

These slides were first presented on August 4, 2023 at Flatiron School.

Today's Objectives

Today, we'll answer the following questions:

1. How do prepare data to be sent to an **external resource**?
2. How do we send data?
3. What can we do after we've successfully sent data?

Preparing Data

- When we want a user to send data to an **external resource** like **JSON Server**, we often have them fill out a **form**.
- Once the user submits the form, we construct an object with what the user submitted.

```
<form>
  <label>
    Name <input name="name" />
  <label/>
  <label>
    Address <input name="address" />
  <label/>
<form>
```

```
form.addEventListener("submit", () => {
  const person = {
    name: event.target.name,
    address: event.target.address
  };
};
```

Sending Data With POST

- Next, we send the object by calling `fetch(URL, OPTIONS)`.

```
form.addEventListener("submit", () => {
  const formData = { /* … */ };
  fetch("http://localhost:3000/objects", {
    method: _______,
    headers: ____________________________,
    body: ______________________________
  });
};
```

Sending Data With POST

- Next, we send the object by calling `fetch(URL, OPTIONS).`

```
fetch("http://localhost:3000/objects", {
  method: "POST",
  headers: {
    Content-Type: "application/json",
    Accept: "application/json"
  },
  body: JSON.stringify(object)
};
```

This defaults to `GET` when it's not specified.

This is where we specify the type of data we're about to send.

This is where we specify the type of data we hope to receive.

This is where we specify the data we want to send.

Handling the Response

- Finally, we attach one or more `.then()` clauses to handle a successful response.
- Our first `.then()` will always be reserved for parsing a successful response.
- Our second `.then()` will always be reserved for using the newly sent data, i.e. the object we just added to our database.
- Optionally, we can attach a `.catch()` clause to handle errors.

```
fetch(URL, { method: POST, headers: { /*…*/ }, body: /*…*/ } )
.then(response => response.json())
.then(newlyCreatedObjectInDatabase => { /* … */ })
.catch(errorObject => { /* … */ });
```

Let’s Try It!

THANKS!