React: fetch( )

Phase 2 // Week 2, Day 1

Sakib Rasul | July 10, 2023

Today's Objectives

Today, we'll answer the following questions:

1. What is **CRUD**? How do we implement it with `fetch()`?
2. How do we write `GET`, `POST`, `PATCH`, and `DELETE` requests in React?

Create, Read, Update, Delete

- In computer programming, **C**reate, **R**ead, **U**pdate, and **D**elete are the four basic operations of persistent storage.
- They are the four things we usually want to be able to do with data.
- How can we use `fetch()` to execute each of these operations? 🤔

Create, Read, Update, Delete

- In computer programming, **C**reate, **R**ead, **U**pdate, and **D**elete are the four basic operations of persistent storage.
- They are the four things we usually want to be able to do with data.
- How can we use `fetch()` to execute each of these operations? 🤔
  - To create data, we can write `POST` requests.
  - To read data, we can write `GET` requests.
  - To update data, we can write `PATCH` requests.
  - To delete data, we can write `DELETE` requests.

Writing a GET Request in React

```
export default function Museum() {
  const [galleries, setGalleries] = useState([]);
  useEffect(() => {
    fetch("https://museum.co/api/galleries")
    .then(response => response.json())
    .then(data => setGalleries(data));
  }, []);
  return (
    <main>
      {galleries.map(gallery => <Gallery key={gallery.id} gallery={gallery}/>)}
    </main>
  );
}
```

Writing a POST Request in React

```
export default function Museum() {
  const [galleries, setGalleries] = useState([]);
  useEffect(() => fetch("https://museum.co/api/galleries").then(res => res.json()).then(gs => setGalleries(gs)), []);
  function post(toBeAddedGallery) {
    fetch("https://museum.co/api/galleries", {
      method: "POST", headers: { "Content-Type": "application/json", "Accept": "application/json" },
      body: JSON.stringify(toBeAddedGallery)
    }).then(response => response.json())
      .then(addedGallery => setGalleries([…galleries, addedGallery]));
  }
  return (
    <main>
      {galleries.map(/*…*/)}
      <Form onSubmit={post} />
    </main>
  );
}
```

Writing a PATCH Request in React

```
export default function Museum() {
   const [galleries, setGalleries] = useState([]);
   useEffect(() => fetch("https://museum.co/api/galleries").then(res => res.json()).then(data => setGalleries(data)), []);
   function post(toBeAddedGallery) {/*…*/}
   function patch(toBeUpdatedGallery, id) {
      fetch(`https://museum.co/api/galleries/${id}`, {
         method: "PATCH", headers: { "Content-Type": "application/json", "Accept": "application/json" },
         body: JSON.stringify(toBeUpdatedGallery)
      }).then(response => response.json())
        .then(updatedGallery => setGalleries(galleries.map(gallery => gallery.id === id ? updatedGallery : gallery)));
   }
   return (
      <main>
         {galleries.map(gallery => <Gallery key={gallery.id} gallery={gallery} onPatch={patch} />)}
         <Form onSubmit={post} />
      </main>
   );
}
```

Writing a DELETE Request in React

```
export default function Museum() {
  const [galleries, setGalleries] = useState([]);
  useEffect(() => fetch("https://museum.co/api/galleries").then(r => r.json()).then(gs => setGalleries(gs)), []);
  function post(toBeAddedGallery) {/*…*/}
  function patch(toBeUpdatedGallery, id) {/*…*/}
  function remove(id) {
    fetch(`https://museum.co/api/galleries/${id}`, { method: "DELETE" })
    .then(response => response.ok && setGalleries(galleries.filter(gallery => gallery.id !== id)));
  }
  return (
    <main>
      {galleries.map(gallery => <Gallery key={gallery.id} gallery={gallery} onPatch={patch} onRemove={remove} />)}
      <Form onSubmit={post} />
    </main>
  );
}
```

Let’s write some requests!