

Sending Data

For express requests, there are four different types of requests.

- GET (default) - Retrieving data from the server. If you don't specify the method in the `fetch()`, this is the default.
- POST - Sending data to the server.
- PUT - Updating/replacing data on the server.
- PATCH - Partially updating data on the server. Similar to PUT, except you only want to modify certain fields in the data.
- DELETE - Removing data.

On the backend

On the server, we need to set up an endpoint for our data to go to. Let's say you want a method that handles login requests. Unfortunately, express isn't set up to handle data that isn't a JSON, so we're going to have to convert everything to a JSON. On our server, it might look something like this:

```
// After setting up express...

// This tells our server that we want to handle JSON requests
app.use(express.json());

app.post('/login', (req, res) => {
  const username = req.body.username;
  const password = req.body.password;
  res.send(`The username is: ${username}, and the password is: ${password}`);
});
```

The `app.post` will wait and listen for any 'POST' requests to `https://localhost:3000/login` from our frontend. Then, it will look in the request's body to find 'username' and 'password' variables. Then, it just sends them back to the frontend. In the future, we would want to do something else like putting the username and password into the database.

On the frontend

Now that we have our server set up to handle requests, we need to send that data to our server from the frontend when they log in. It would look something like this:

```
const [username, setUsername] = useState('Justin');
const [password, setPassword] = useState('Luce');

function login() {
  // Makes a request to our server's port, specifically to the 'login' endpoint
  fetch('https://localhost:3000/login', {
    method: 'POST', // Specifying that we're making a POST request (sending data)
```

```
// Headers is for metadata (extra information) about the request to help the
// browser/server understand how to handle it
headers: {
  'Content-Type': 'application/json', // Telling the server that the data
  // we're sending is a JSON
},
// The body is the main content that we want to send
body: JSON.stringify({ username: username, password: password }), // We want
// to send their username and password as a JSON string to our server
})
// Once the request completes, these .then statements will run
.then(responses => response.json()) // Turns the response from the server into a
// JSON
.then(data => console.log(data)) // Once it's turned into a JSON, we log it to
// the console
.catch(err => console.error("An error occurred: ", err)) // If there was an
// error, log that to the console
}

<button onClick={() => login()}>Login</button>
```

What about passwords?

- So in the above example we're sending a JSON object with a `username` and `password`. It's being sent as plain text. Is this a bad thing, if so why/why not?