

# Updating the Message State

The SEND\_MESSAGE action will now be handled in the messages reducer. This will be a simple challenge for now.

We're nearing the end of our task. Hang in there!

In this section, we will figure out how to update the message state to allow messages to appear in the chat window when they are submitted.

Here's a list of all the reducers we've got at this point:

- activeUserId.js
- contacts.js
- messages.js
- typing.js
- user.js

Which of these do you think should be concerned with updating the messages in a user conversation?

Yes, the messages reducer.

Here's the current content of the messages reducer:

```
import { getMessages } from "../static-data";
export default function messages(state = getMessages(10), action) {
  return state;
}
```



messages.js

Not so much going on in there.

Import the **SEND\_MESSAGE** action type, and let's begin to handle that in this messages reducer.

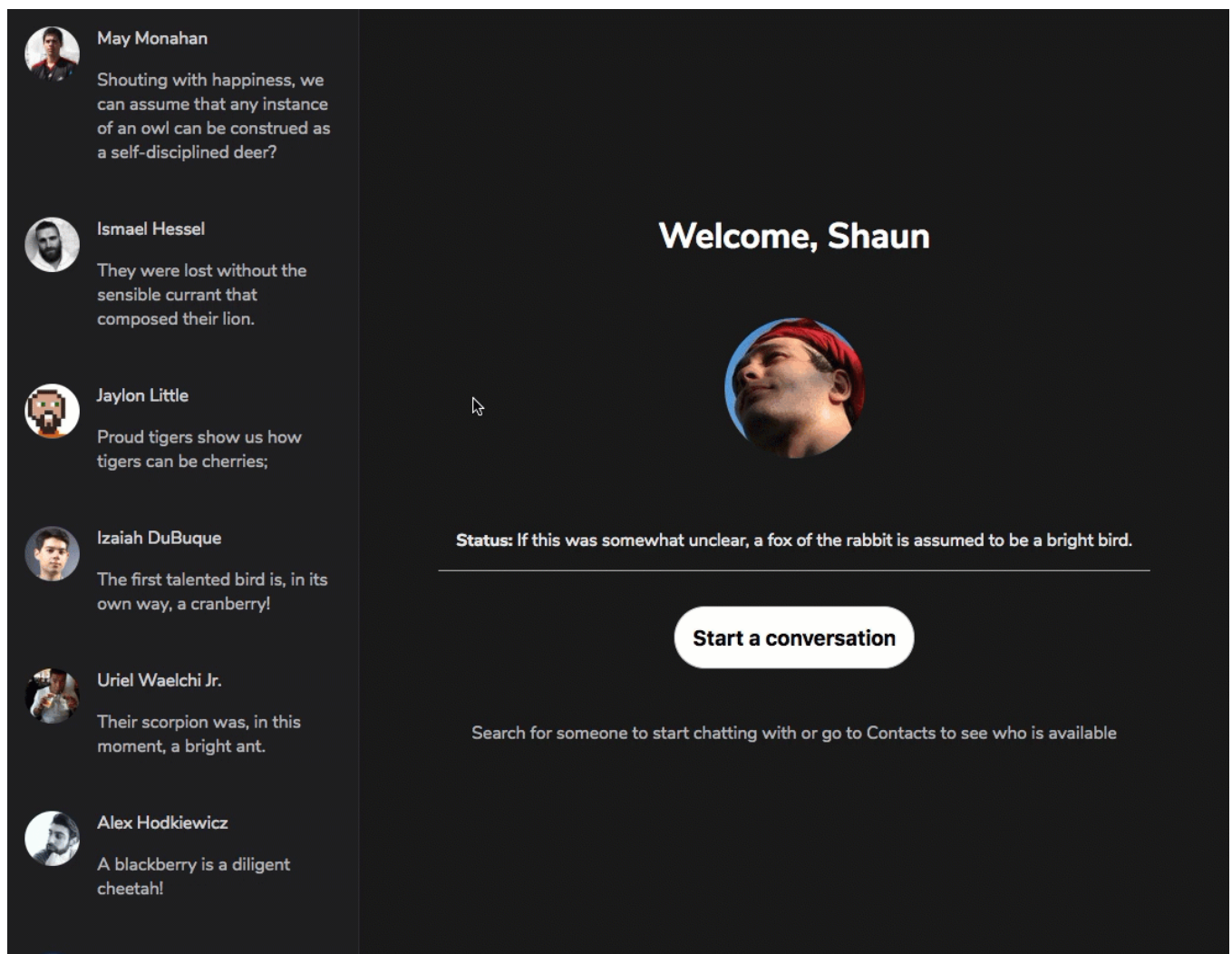
```
import { getMessages } from "../static-data";
```

```
import { getMessages } from "../static-data";
import { SEND_MESSAGE } from "../constants/action-types";
export default function messages(state = getMessages(10), action) {
  switch (action.type) {
    case SEND_MESSAGE:
      return "";
    default:
      return state;
  }
}
```

Now, we are handling the the action type, `SEND_MESSAGE` but an empty string is returned.

This isn't what we want, but we'll build this up from here. In the mean time, what do you think is the consequence of returning an empty string here?

Let me show you.



All the messages disappear!

But why? That's because as soon as we hit enter, the `SEND_MESSAGE` action is dispatched. As soon as this action reaches the reducer, the reducer returns an empty string ""

Thus, there are NO messages in the state object. It's all gone!

This is definitely unacceptable.

What we want is to retain whatever messages are in state.

However, we want to add a new message ONLY to the messages of the active user.

Okay. But how?

Let's talk about it in the next lesson.