The Message Component

Just as with all our previous components, we'll first implement the initial set up for the Message component. Make an action, create a reducer to facilitate the action, and check if the message field show up in the Log.

We've had to build more difficult components.

This one won't be difficult to build. However, there's one point to consider.

The Input component will be a controlled component.

Therefore we will be storing the input value in the application state object.

For this, we'll need a new field called **typing** in the state object.

Let's get that in there.

For our considerations, whenever a user types, we will dispatch a `SET_TYPING_VALUE` action type. Be to sure add this constant in the **constants/action-types.js** file.

```
export const SET_TYPING_VALUE = "SET_TYPING_VALUE";
```

Also, the shape of the dispatched action will look like this:

```
{
  type: SET_TYPING_VALUE,
    payload: "input value"
}
```

Above, the payload of the action is the value typed in the input. Let's create an action creator to handle the creation of this action:

actions/index.js:

```
import {
  SET_ACTIVE_USER_ID,
  SET_TYPING_VALUE
} from "../constants/action-types";
```

```
...
export const setTypingValue = value => ({
  type: SET_TYPING_VALUE,

  payload: value
})
```

Now, let’s create a new typing reducer - one that will take this created action into consideration.

reducers/typing.js

```
import { SET_TYPING_VALUE } from "../constants/action-types";
export default function typing(state = "", action) {
  switch (action.type) {
    case SET_TYPING_VALUE:
      return action.payload;
    default:
      return state;
  }
}
```

The default value for the typing field will be set to an empty string:

However, when an action with type, `SET_TYPING_VALUE` is dispatched, the value in the payload will be returned.

handle the SET_TYPING_VALUE type

Otherwise, the default state, “” will be returned.

Be sure to include this newly created reducer in the **combineReducers** function call.

reducers/index.js:

```
...
import typing from "./typing";
export default combineReducers({
  user,
  messages,
  typing,
  contacts,
  activeUserId
});
```

Be sure to inspect the logs and ascertain that a typing field is indeed attached to the state object.

Once we’ve done that, we can finally start building the actual component.