

Actions and reducers 101

It's time to have our store handle changing data. We've dispatched a fake action but now let's create a reducer and some actions that actually do some useful work.

1. Add a file in the store folder called reducers.js. Move the reducer out of store.js into reducers.js and export it properly. Don't forget that you'll need to import it in store.js now.
2. Add a new line inside the body of your reducer function:

```
console.log("In reducer", state, action);
```

3. Run and test to make sure your refactor hasn't broken anything. You should be seeing your DUMMY_ACTION being dispatched.

Putting in some safeties

4. Add a switch statement[†] to your reducer, switching on action.type. (Remember, action.type is always required and always a string).
5. Put in a *default* case that simply returns the old state unchanged.
6. Again, run and test. Your refactor is now safer but still doesn't do much. Let's fix that now.

Adding in your first case

7. Create a case in the switch to say that if the user dispatches a "SET_FILMS" action, we read the films payload and set that in state. Something like this should do the trick:

```
case "SET_FILMS":
```

```
  return {...state, films: action.films}
```

8. Now edit App.js. Find the useEffect. Add this inside of it:

```
fetch("/api/films")
```

```
.then(res=>res.json())
```

```
.then(films=>dispatch({type:"SET_FILMS",films}));
```

What we just did was send off a request to the API server for some data; a list of films. Then, when we get a response, convert it from a stream of JSON and dispatch an action to set those films.

9. Run and test. If your component is still console.log()ging state, you should see that we've loaded a bunch of films into state.films. Cool, right?

Bonus!! Showing them on the page

10. It'd be a lot cooler if we could see all of the films on the page. Find the <main> element and add this inside it:

```
<section>
```

```
  {state.films?.map(film => <div key={film.id}>{film.title} - {film.tagline}</div>)}
```

```
</section>
```

This is just some React which will be covered elsewhere. But for the curious, this will iterate through all of the state's films and show each title and tagline in a <div>.

[†] <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/switch>

Adding in some new actions

11. This seems like a good time to add in reducer actions that will be needed in the future. Go ahead and add these:

action.type	action.payload
SET_CURRENT_DATE	date
SET_CURRENT_FILM	film
SET_CURRENT_SHOWING	showing
SET_TABLES	tables
SET_THEATERS	theaters
SET_SEATS	seats
SET_SHOWINGS	showings
SET_USER	user

12. Bonus! Whenever you're ready, you can delete that fake dispatch that we added earlier for testing along with any `console.log()`s that are distracting you.