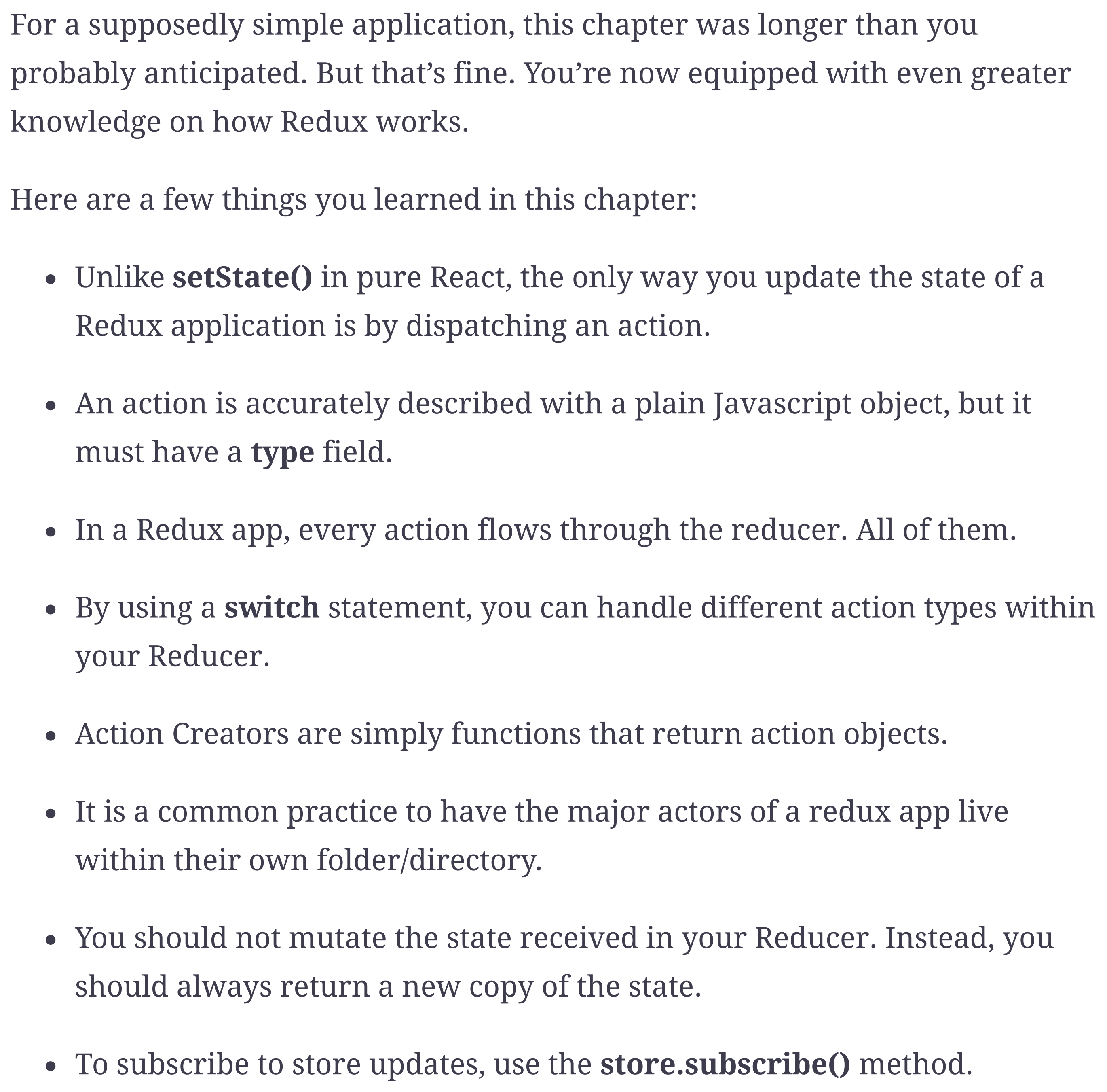
Conclusion and Summary

This chapter was slightly longer. Below we have some of the fundamental topics that we covered.



For a supposedly simple application, this chapter was longer than you probably anticipated. But that’s fine. You’re now equipped with even greater knowledge on how Redux works.

Here are a few things you learned in this chapter:

- Unlike **setState()** in pure React, the only way you update the state of a Redux application is by dispatching an action.
- An action is accurately described with a plain Javascript object, but it must have a **type** field.
- In a Redux app, every action flows through the reducer. All of them.
- By using a **switch** statement, you can handle different action types within your Reducer.
- Action Creators are simply functions that return action objects.
- It is a common practice to have the major actors of a redux app live within their own folder/directory.
- You should not mutate the state received in your Reducer. Instead, you should always return a new copy of the state.
- To subscribe to store updates, use the **store.subscribe()** method.