Conclusion and Summary

And that's it for the final section of our course. Our Skypey app is ready for use. Since, it's been a long journey, you can find some helpful notes from this section below.

Building Skypey was a lot of fun.

Did you love it? We'd love to see your own version of Skypey. Change the colours, tweak the design, and build something better!

Here's a summary of some of the things we've learned so far:

- It is a good practice to **always plan your application development process** before jumping into code.
- In your state object, avoid nested entities at all cost. **Keep the state object normalized**.
- Storing your state fields as objects does have some advantages. **Be equally aware of the issues with using objects**, mainly the lack of order.
- The **Lodash utility** library comes very handy if you choose to use objects over arrays within your state object.
- No matter how little, always take some time to design the state object of your application.
- With Redux, you don't always have to pass down props. **You can access state values directly from the store**.
- Always **keep a neat folder structure** in your Redux apps e.g having all major Redux actors in their own folders. Apart from the neat overall code structure, this makes it easier for other people to collaborate on your project as they are likely conversant with the same folder structure.
- **Reducer composition** is really great especially as your app grows. This

increases testability and reducers the tendency for hard-to-track errors.

- For reducer composition, make use of combineReducers from the redux library.
- The object passed into the combineReducers function is designed to resemble the state of your application with each value gotten from the associated reducers.
- Always **break larger components into smaller manageable bits**. It's a lot easier to build your way up that way.

Be sure to check out the exercises in the next lesson. You'll do a few cool things with the Skypey app.

Catch you later!