**Week 9+10: State Management**

**State Management**

State management is the process of tracking and updating the state of your application. In Flutter, state is anything that can change over time, such as the current user, the contents of a shopping cart, or the position of a map marker.

There are a number of different state management techniques available in Flutter, each with its own advantages and disadvantages. Some of the most popular state management techniques include:

* **Inherited Widget:**This technique allows you to share state between widgets that are not directly connected to each other.
* **StatefulWidget:** This technique allows you to create widgets that can change their state over time.
* **Provider:** This is a popular state management package that makes it easy to share state between widgets.
* **MobX:** This is another popular state management package that uses observables to track state changes.
* **RiverPod:** Riverpod is a state management library for Flutter that uses providers to manage state. Providers are objects that encapsulate state and allow listening to changes in that state. This makes it easy to update the UI when state changes. Riverpod is also very efficient, only rebuilding the UI when state actually changes.

The best state management technique for your application will depend on the specific needs of your app. If you are just starting out with Flutter, I recommend using the Inherited Widget or StatefulWidget techniques. Once you have a better understanding of how state management works, you can explore other options such as Provider or MobX.

Here are some of the benefits of using state management in Flutter:

* Easier to maintain: State management can help you keep your code organized and easier to maintain.
* Fewer bugs: State management can help you avoid bugs that are caused by changes to state.
* Better performance: State management can help improve the performance of your application by reducing the number of times the UI needs to be redrawn.