

---

Vorname	Nachname	Beruf	Dauer
Marten	Meißner	Fachinformatiker AE	07.22-07.24

---

## Wochenbericht KW 4. (2024.01.22. - 28.)

---

Für das aktuelle Projekt wurde sich das MVC (Modell-View-Controller) Pattern als Architekturart entschieden. Da hier für das Backend schon für Firebase entschieden, nutzen wir als Controller das Riverpod Package. Riverpod nutzt das Prinzip der "Dependency Injection" und erlaubt mir in der View, das anzeigen und Manipulieren von Daten.

Dafür gibt es in Riverpod verschiedene Provider.

- Provider
- StateProvider
- NotifierProvider
- StateNotifierProvider
- FutureNotifierProvider
- StreamNotifierProvider

Der Unterschied von dem ersten Provider zu den anderen ist das dieser keine Notifier Klasse erweitert. Dieser Provider eignet sich dafür einzelne Werte, die sich zur Laufzeit nicht ändern sollen, weiter zugeben, ähnlicher einer Globalen Konstanten.

Die anderen können ihren Rückgabewert zur Laufzeit anpassen. Wobei jeder seinem eigenen Spezialgebiet hat. So z. B. ist ein StateProvider, der schlichteste, da dieser mit einem Initialwert erstellt wird und dann angepasst werden kann. Dieser eignet sich aber nur für einfache Objekte, die beobachtet und manipuliert werden wie z. B. Generics oder Enums.

Alle anderen Provider erweitern einen separaten Notifier in den Methoden definiert werden können und dadurch viel mehr Umfang liefert.

Ich nutze in meinen Anwendungsfällen den NotifierProvider. Mit diesem ist es am einfachsten Anfragen an die Firebase API zu stellen. Dadurch dass die Methoden meistens als void also ohne Rückgabewert definiert werden, gibt es hier keine Umstände mit Asynchronen Abfragen in GUI. Die GUI reagiert nur auf Anpassung der Provider und nicht auf Aktivieren einer Methode.

In der GUI kann ich so sehr viel Stateless Widget nutzen, da die Provider bei Veränderung das Widget automatisch dazu anregen sich neu aufzubauen. Die einzige Anpassung in der GUI ist, dass die Widgets, die auf ein Provider reagieren, als „ConsumerWidget“ definiert werden müssen, der WidgetTree muss in ein Consumer gewrapt werden, um ein „WidgetRef“ Parameter nutzen zu können. Dann kann ich über den Parameter die „watch“ oder „read“ Methode und dann entweder den Provider zu beobachten oder den jeweiligen Notifier zu lesen, um eine Methode zu aktivieren, die wiederum den State im Provider ersetzt.

Kontrolliert am: \_\_\_\_\_ Unterschrift : \_\_\_\_\_