

# StateNotifierProvider

---

## Einführung

In der letzten BMI-App mit StateProvidern haben wir festgestellt, dass selbst für eine einfache App mindestens vier Provider erstellt werden mussten. Bei größeren Apps würde dies zu einer erheblichen Zunahme an Komplexität führen. Außerdem wurden die Funktionen direkt über `provider.notifier` im GUI definiert, was die Trennung zwischen GUI und Logik beeinträchtigt. Um diese Probleme zu vermeiden, benötigen wir ein besser geeignetes Werkzeug.

## Defintion

Der StateNotifierProvider ist ein Provider in Riverpod, der es ermöglicht, den Zustand mit einer StateNotifier-Klasse zu verwalten. Er eignet sich, um komplexe Logiken und Zustandsänderungen sauber und testbar zu implementieren.

## Aufbau Schritte:

Der Aufbau von `StateNotifierProvider` ist nicht so einfach wie bei den anderen Providern. Er erfordert eine zusätzliche Klasse (`StateNotifier`-Klasse).

### 1. Der Bau der `StateNotifier`-Klasse:

Die `StateNotifier`-Klasse ist eine Klasse, die von der eingebauten `StateNotifier`-Klasse in Riverpod erbt. Beim Erstellen dieser Klasse muss der Zustandstyp (egal ob es sich um einen einfachen Typ oder eine erstellte Klasse handelt) angegeben werden. Der Konstruktor der `StateNotifier`-Klasse wird durch `spuer` Konstruktor erstellt. Sobald die `StateNotifier`-Klasse fertig ist, können darin alle erforderlichen Methoden erstellt werden. So wird die Logik vom GUI getrennt, und der Bau der `StateNotifier`-Klasse ist abgeschlossen.

### 2. Der Bau des `StateNotifierProvider`:

Der `StateNotifierProvider` wird wie folgt erstellt:

```
provider = StateNotifierProvider<StateNotifier, State>((ref) =>
    StateNotifier());
```

## Wichtiger Hinweis:

Die `StateNotifier`-Klasse bereitet ein `state`-Objekt vor, das die aktuellen Werte enthält. Durch dieses `state`-Objekt können die Methoden der Klasse aufgebaut werden. Wenn man eine Methode wie `increment()` baut, reicht es nicht aus, einfach `state + 1` zu schreiben. Stattdessen muss die Änderung wie folgt gespeichert werden:

```
state = state + 1;
```

## Manipulieren im GUI durch Button

Um eine Funktion aus der `StateNotifier`-Klasse aufzurufen, kann man dies ganz einfach mit dem folgenden Code erreichen:

```
ref.read(providerName.notifier).methodeName();
```

## Wichtige Hinweise:

1. Die `State`-Klasse soll **immutable** sein.
2. Alle Methoden, außer `copyWith`, sollen nicht in der `State`-Klasse, sondern in der `StateNotifier`-Klasse erstellt werden.