



# Agenda - Beispiel

Dies ist unsere empfohlene Agenda für das Thema RxJS. Wir verfügen darüber hinaus über weiteres Kursmaterial, um Themen einfach auszutauschen.

## Reaktive Programmierung mit RxJS

Reaktive Programmierung ist event-basiertes Programmieren wozu meistens das Observable-Pattern genutzt wird. Dieses Pattern ermöglicht es Streams von Daten durch verschiedene Operatoren zu managen, ohne das wir uns zu viel um low-level Operationen selber kümmern müssen.

- Event-Basiertes Programmieren
- synchron vs. asynchron
- Vergleich zu anderen Reaktiven Bibliotheken
- Observables
- Observables vs. Promises
- Fehlerbehandlung

## Subjects

Ein Subject ist ein Proxy Objekt welches sowohl ein Observable als auch Observer implementiert. Dies ist eine sinnvolle Abstraktion, da ein Observable selber auf mehrere Observable subscriben kann. Je nach Anwendungsfall gibt es verschiedene Standardverhalten von Subjects welche du kennen solltest.

- Subjects Einführung
- Cold vs. Hot Observables
- Operatoren im Detail
- Observables vs. Subjects
- Different Types of Subjects

## Operatoren

Für den effizienten Umgang mit Observables liefert RxJS von Haus aus einige Operatoren mit. Diese helfen uns Daten auf unserem Stream zu erzeugen, transformieren, filter oder auch zu kombinieren.

- Einführung in die gängigsten operatoren
- Verkettung von Operators
- Verschiedene Kategorien von Operatoren

- Erstellen und Transformieren
- Filtern und Kombinieren von Stream
- Error Handling
- Utility Operators

### **RxJS am Beispiel Angular**

Um RxJS in einem echten Projekt einzusetzen, nutzen wir hierbei das Framework Angular. Angular basiert sehr stark auf RxJS und bietet somit eine gute Basis um RxJS zu lernen. RxJs ist jedoch eine unabhängige Bibliothek, andere Frameworks wie React oder Vue nutzen ebenfalls RxJS. Das Wissen lässt sich auch auf diese übertragen.

- RxJS in Angular
- EventEmitter für User-Events
- Umgang mit Http Events
- Reaktive Formulare
- Reaktive State Management