

# rxAngular

Zwischenbericht | Call 18 | Projekt ID 6798

Lizenz: CC BY-SA



# Inhalt

1	Einl	eitung	3	
2	Stat	us der Arbeitspakete	3	)
	2.1	Arbeitspaket 1 - Detailplanung und Formales am Projektstart	3	
	2.2	Arbeitspaket 2 - Entwicklung Modul Frame-Budget-Optimierung Arbeitspaket 3 - Entwicklung Modul Task Abortion		
	2.4	Arbeitspaket 4 - Entwicklung Modul Coalescing mit Scoping-Mechanismus	6	)
	2.5 2.6	Arbeitspaket 5 - Projektmanagement und Zwischenbericht	7 7	,
3	Um	setzung Förderauflagen	8	-
4	7us	ammenfassung Planaktualisierung	8	
5	Öffe	ntlichkeitsarbeit/ Vernetzung	8	3
6	Fige	ne Projektwebsite	C	
_	-18	The fire jetter exerce		



# 1 Einleitung

Das RxAngular-Team war sehr produktiv und hat mit den letzten Releases neue Funktionen sowie Verbesserungen der bestehenden Lösung bereitgestellt, einschließlich der nativen Unterstützung für Signals in den Kerndirektiven:

#### Feinkörnige Reaktivität

Entwickelt und optimiert wurden RxAngular-Direktive für eine feinkörnige Reaktivität bis hinunter zu einzelnen Template-Schnipseln (="EmbeddedViewRef"). Das bedeutet, dass bei Datenänderungen nur die notwendigen Teile von Angular-Templates aktualisiert werden, was zu einer wesentlich besseren Leistung führt.

#### **Kontextvariablen und Templates**

Mit unseren neuen RxAngular-Direktiven ist es nun möglich, intuitiv mit Kontextvariablen und Templates zu arbeiten. Dies erleichtert die Verwaltung komplexer Datenstrukturen und Logiken in Angular-Templates.

#### Lazy template creation

Unsere RxAngular-Direktiven wurden so entwickelt, um Lazy-Template-Erstellung (vgl. Lazy Loading¹) zu ermöglichen, was zur Performance-Verbesserungen beiträgt, indem die Erstellung von Template-Elementen aufgeschoben wird, bis sie tatsächlich benötigt werden.

#### **Concurrent Mode**

Benutzer:innen möchten 60 fps und reibungslose Interaktionen mit Anwendungen, und RxAngular-Direktives sind dafür optimiert. Sie stellen sicher, dass die Anwendungen auch bei großen Mengen an Änderungen reaktionsfähig und leistungsfähig bleibt, indem der Concurrent Mode (intelligente Aufteilung u Priorisierung von Datenströmen) ermöglicht wird.

### 2 Status der Arbeitspakete

#### 2.1 Arbeitspaket 1 - Detailplanung und Formales am Projektstart

#### Haupttätigkeiten:

- Verfeinerung der Entwicklungs-Roadmap
- 1. Blogbeitrag verfassen
- 1. Förderrate beantragen

<sup>&</sup>lt;sup>1</sup> Lazy Loading ist eine Webentwicklungstechnik, die das Laden von z.B. Bildern verhindert, bis sie benötigt werden.



Hauptergebnis: 1. Blogbeitrag

https://www.netidee.at/rxangular/rxangulars-roadmap-after-funding

Abweichungen zum Plan: 1 neues Team-Member

Mit Enea Jaholari kam ein weiterer Entwickler zum Team, dessen Onboarding-Phase erst nach Antragsabgabe abgeschlossen war. Mit seinem Spezialwissen im Bereich des performantem List-Diffing kann er einen wichtigen Beitrag zum Projekt leisten.

#### 2.2 Arbeitspaket 2 - Entwicklung Modul Frame-Budget-Optimierung

#### Haupttätigkeiten:

- Wesentliche Verbesserungen zur Virtual-Scrolling-Funktion
- Erster Entwurf zu einem Reconciliation-Algorithmus für effizienteres, asynchrones Listen-Diffing

Wir haben erkannt, dass die Virtual-Scrolling-Funktion (= Benutzer:innen wird nur ein kleiner Teil der Daten zu einem bestimmten Zeitpunkt angezeigt) mit Stand der Technik nur ineffizient genutzt wird:

- 1) Die Anzeige der sichtbaren Elemente hatte zur Folge, dass die Position anderer Elemente invalidiert wurde und bei jedem Update neu berechnet werden musste.
- 2) Ineffizienz, da die Updates beim Scrollen auf dem Main-Thread ausgeführt wurde.

Unsere Verbesserungen beruhen auf einer Kombination von 3 verschiedenen Ansätzen:

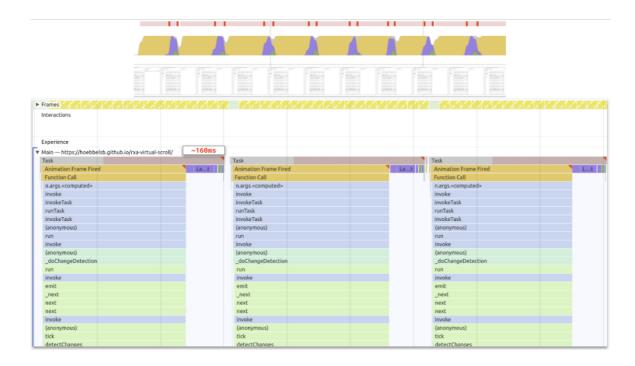
- 1) Gänzliches Weglassen aller Elemente, die nicht sichtbar sind
- 2) Bei Einblendung neuer Elemente die Anzeigestruktur alter Elemente recyclen, anstatt neu zu erzeugen
- 3) Feinkörnige Reaktivität: Nur Elemente, welche tatsächlich verändert werden müssen, werden auch neu dargestellt.
- 4) Concurrent Mode: Anwendung des Concurrent Modes auf sichtbare Elemente.

#### Messergebnisse:

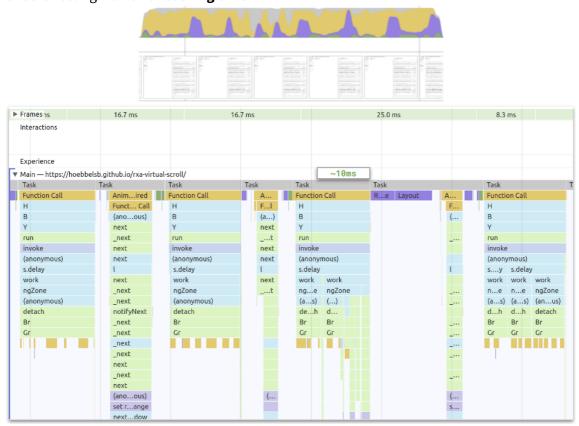
https://www.rx-angular.io/docs/template/api/virtual-scrolling#comparison-with-angular-cdk

Existierende Virtual-Scrolling-Lösung (@angular/cdk/virtual-scrolling): 160ms blocking time





#### Unsere Lösung: 10ms no blocking time



Demo-Applikation zum Vergleich verschiedener Scroll-Strategien: <a href="https://hoebbelsb.github.io/rxa-virtual-scroll/">https://hoebbelsb.github.io/rxa-virtual-scroll/</a>

Abweichungen zum Plan: Ausfall zweier Team-Member; Verlängerung um 2 Monate



#### 2.3 Arbeitspaket 3 - Entwicklung Modul Task Abortion

Haupttätigkeiten: Weiterentwicklung für Native PostTask-Scheduling-Support

Unsere Entwicklungstätigkeiten zielten darauf ab, die bestehenden Nachteile von benutzerdefinierten Scheduler zu erstellen, um Aufgaben innerhalb der von Ihnen kontrollierten Codebasis zu priorisieren, verwenden Skripte von Drittanbietern möglicherweise nicht denselben Scheduler. Folglich wird die Priorisierung von Aufgaben in Umgebungen schwierig, in denen Sie nicht die Kontrolle über den gesamten Code auf der Seite haben. Die Alternativen beschränken sich auf das Aufteilen von Aufgaben oder das explizite Nachgeben gegenüber Benutzerinteraktionen. rxAngular zielt darauf ab, diese Nachteile zu überwinden, indem wir die native scheduler.postTask-API einbinden (in Arbeit).

#### Weitere Arbeiten:

- Task Aborting: Erweiterung der Implementierung zur Unterstützung des Aufgabenabbruchs, so dass geplante Aufgaben abgebrochen werden können.
- Task Prioritization Configuration: Untersuchung von Optionen zur Konfiguration der Aufgabenpriorisierung auf der Grundlage spezifischer Anforderungen oder Anwendungsbedürfnisse.
- Testing: Implementierung von Tests zur Sicherstellung der Zuverlässigkeit und Kompatibilität der nativen scheduler.postTask-API-Integration in verschiedenen Browsern und Umgebungen.

Abweichungen zum Plan: Ausfall zweier Team-Member; Verlängerung um 2 Monate

#### 2.4 Arbeitspaket 4 - Entwicklung Modul Coalescing mit Scoping-Mechanismus

<u>Haupttätigkeiten:</u> Weiterentwicklungen zur Einbindung von Angular Signals.
Signals sind insbesondere für Junior Angular-Entwickler\*innen ein hilfreiches Tool, da sie das reaktive Programmieren entscheidend erleichtern. Im Grunde sind sie ein Signalgeber mit einem Wert, der alle verbundenen Konsumenten benachrichtigt, wenn sich dieser Wert ändert.

Wir haben erkannt, dass es mit Angular Signals zu unerwarteten Problemen (konkret: Over Rendering) mit einer unserer bisherigen Render-Strategie kommt, wenn Flags zur Markierung von zu rendernden Objekten eingesetzt werden. Als mögliche Lösungsansätze haben wir untersucht, ob und wie sich die Deaktivierung von Parent-Flags auswirkt.

Ergebnis: Wir haben diese Diskussion lange geführt und können nun endlich damit beginnen, das Parent-Flag in den Render-Strategien und als Eingabe für unsere Direktiven zu verwerfen. Mit den neuen Signal Queries ist das Parent-Flag überflüssig, es verursacht nur unnötige Änderungserkennungszyklen. Dies wurde mit neuen Tests und einer neuen Demo bestätigt.



deprecate parent flag: <a href="https://github.com/rx-angular/rx-angular/pull/1710">https://github.com/rx-angular/rx-angular/pull/1710</a> drop parent flag: <a href="https://github.com/rx-angular/rx-angular/pull/1734">https://github.com/rx-angular/rx-angular/pull/1734</a> signal compatability: <a href="https://github.com/rx-angular/rx-angular/pull/1714">https://github.com/rx-angular/rx-angular/pull/1714</a>

Abweichungen zum Plan: Ausfall zweier Team-Member; Verlängerung um 2 Monate

#### 2.5 Arbeitspaket 5 - Projektmanagement und Zwischenbericht

#### Haupttätigkeiten:

- Zwischenbericht erstellen
- 2. Blogbeitrag verfassen
- 2. Förderrate beantragen

Hauptergebnis: 2. Blogbeitrag

https://www.netidee.at/rxangular/new-features-rxangular

Abweichungen zum Plan: keine

#### 2.6 Arbeitspaket N - Dokumentation und Formales am Projektende

#### Haupttätigkeiten:

- Homepage, Dokumentation und Tutorials (neue Features) aktualisieren
- APIs entwickeln
- Weiterentwicklung auf Version 18 aufgrund Update des zugrunde liegenden Frameworks (Angular 18)

Problem mit Meta-Tags gelöst, wenn unsere Homepage auf Social Media geteilt wird. <a href="https://github.com/rx-angular/rx-angular/pull/1732">https://github.com/rx-angular/rx-angular/pull/1732</a>

Angular18 Update (https://github.com/rx-angular/rx-angular/pull/1730)

- Deprecated API-Verwendungen verhindern
- Demo App wurde an neue Angular-Features angepasst
- Version bump & Anpassung von Peer-Dependencies:
  - o @rx-angular/cdk
  - o @rx-angular/template

netidee Call 18 Zwischenbericht Projekt-ID 6798

- @rx-angular/state
- o @rx-angular/isr

Abweichungen zum Plan: keine



# 3 Umsetzung Förderauflagen

Es wurden keine speziellen Förderauflagen festgelegt.

# 4 Zusammenfassung Planaktualisierung

Aufgrund des Ausfalls zweier Team-Member (1 krankheitsbedingt; 1 aus persönlichen Gründen) wurde das planmäßige Projektende von 10/2024 auf 12/2024 verschoben.

# 5 Öffentlichkeitsarbeit/Vernetzung

2024 fanden bereits einige Events mit Talks und Workshops des RxAngular-Teams statt:

#### 10.02.2024: ngIndia, Delhi/India

10.02. Talk: 3 Reactive Primitives - Architecture in Modern Angular

#### 24.02.2024: ngAsia

24.02. Performance at Scale: Non-trivial Optimization Techniques in Angular at NG Asia Meetup

#### 18.03.-21.03.2024: ngConf, Salt Lake City/USA

18.03. Workshop: Nx for Scalable Angular Architectures (with Julian)

19.03. Workshop: Performance at Scale: High-Speed Angular Applications On Any Device (with Julian)

20.03. Talk: 3 Cornerstones ineffable in future Angular architecture

#### 06.05.-08.05.2024: EnterJS, Mainz/Germany

06.05. Workshop: Modern Angular at Scale – Performance and Architecture

07.05. Talk: Modern Angular, Change Detection and Signals

08.05. Talk: How To Migrate Into a Monorepo: Strategies and Best Practices

#### 24.05.2024: ngBelgrade, Belgrade/Serbia

24.05. Talk: Angular Performance and Core Web Vitals in 2024

#### 25.06.–27.06.2024: ngRome, Rome/Italy

25.06. Workshop: High-speed Angular applications on any device 27.06. Talk: Angular Performance and Core Web Vitals in 2024



#### Upcoming Events mit Beiträgen des RxAngular-Teams:

#### Monorepo World: 7 Oct 2024

Monorepo World is the ultimate conference where platform architects and tool builders come together to advance the state of software development.

#### ng-DE

NG-DE is the first community-driven Angular conference in Germany. It will take place in Bonn, October 2024.

https://push-based.io/event/high-speed-angular-apps-on-any-device https://push-based.io/event/going-deep-into-signals-change-detection https://push-based.io/event/cut-my-task-into-pieces-this-is-concurrent-mode

#### ng-TLV: November 27th, 2024

Angular TLV stands as the cornerstone of Israel's thriving Angular community. Ng-TLV 2024 is the first ever Angular conference Israel has ever seen

#### ng-BE: December 5 & 6, 2024

NG-BE is a 2-day event in Ghent, Belgium, that brings together Angular developers and experts from all over the world to share ideas, news and opinions about Angular.

#### ngPoland November 2024 (Warsaw & Online)

The biggest Angular Conference in Europe.

# 6 Eigene Projektwebsite

https://www.rx-angular.io/