



Angular Advanced short recap – day #2



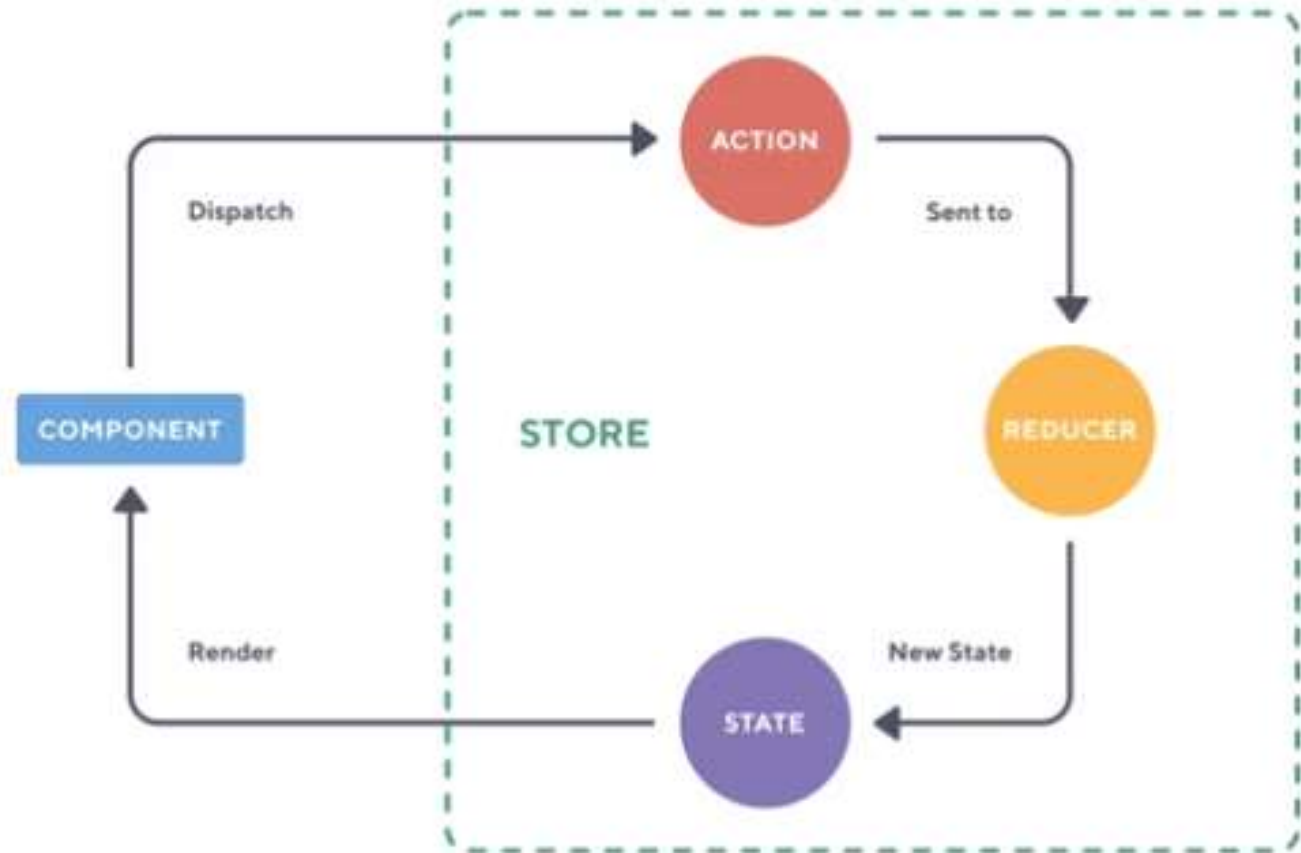
Peter Kassenaar
info@kassenaar.com

Short recap day 2 –

- Routing events
- Observables – from scratch
 - The observable design pattern
 - `.subscribe()`, `| async`
 - Operators: `map`, `tap`, `filter`, `catchError`, ...
- State Management
 - Concepts – store, state, actions, reducers, dispatch

REDUX ARCHITECTURE

One-way dataflow



<https://platform.ultimateangular.com/courses/ngrx-store-effects/lectures/3788532>



Daniele Ghidoli

About me

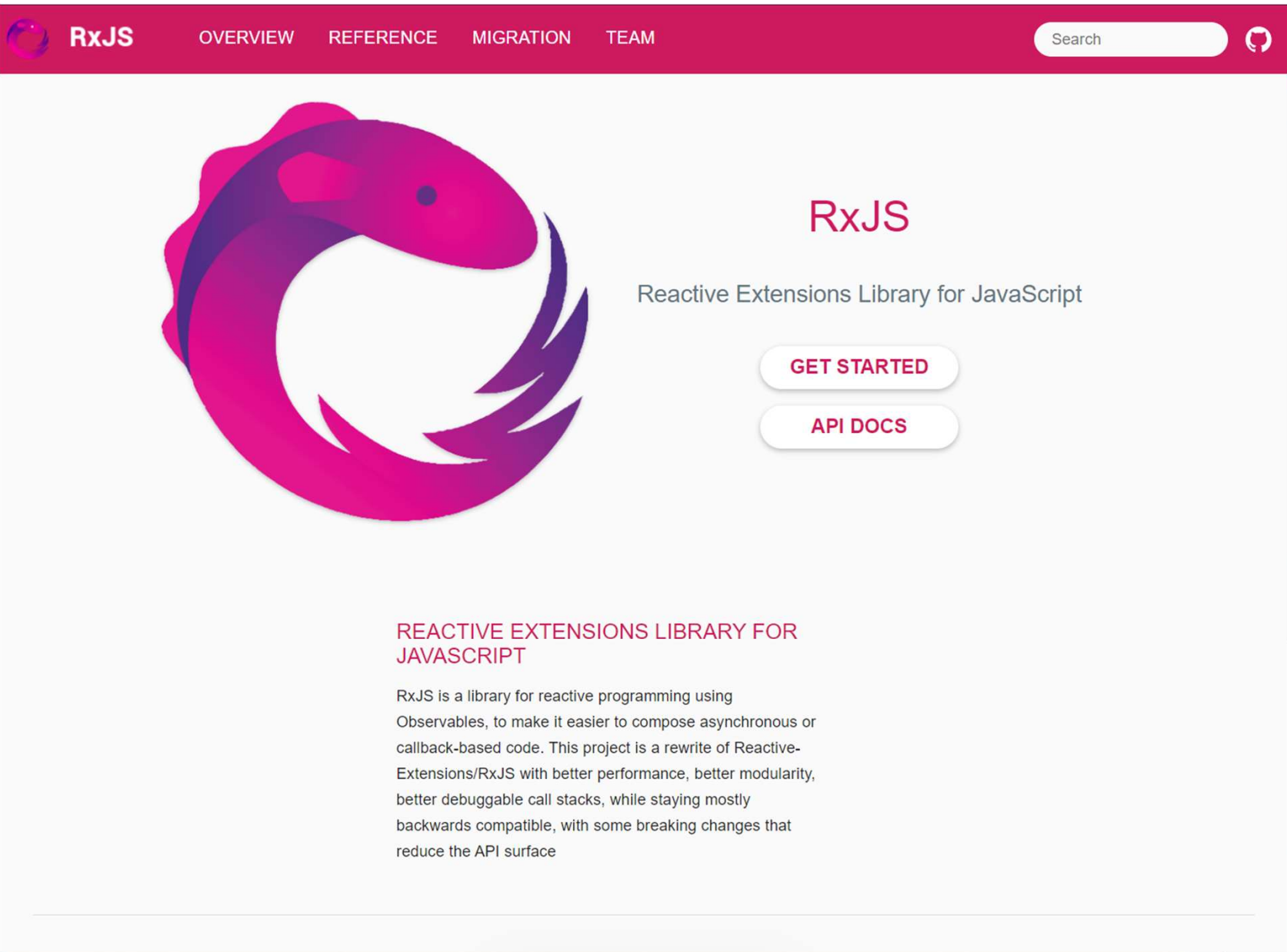


October 22, 2016

Combining multiple Http streams with RxJS Observables in Angular2



<http://blog.danieleghidoli.it/2016/10/22/http-rxjs-observables-angular/>



<https://rxjs-dev.firebaseio.com/>

[reactive.how](#) 2.0-alpha.4

[Launchpad for RxJS →](#)

Follow Stay tuned

ESSENTIALS	REDUCING	TAKING	SKIPPING	COMBINING	TIME	CONCATENATING
map	reduce	take	skip (soon)	combineLatest	delay	startWith (soon)
filter	max	takeWhile	skipWhile	zip	debounceTime	endWith (soon)
merge	count	first	takeLast		throttleTime	concat (soon)
scan			last			

Learn RxJS operators and Reactive Programming principles

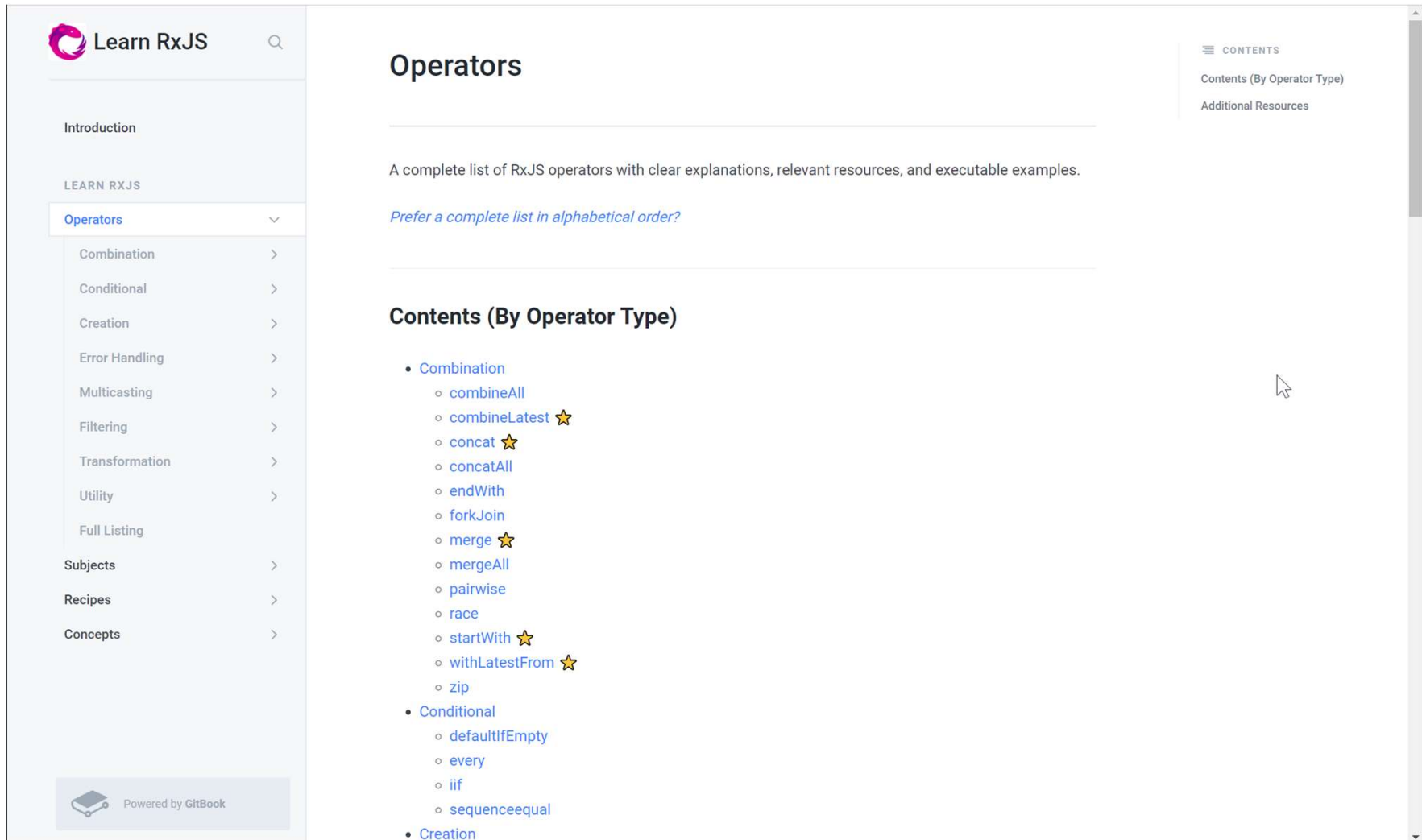
Launchpad for RxJS

debounceTime vs throttleTime

Decks of cards for RxJS: learn, compare and memorize

<https://reactive.how/>

More info on Observables



The screenshot shows the 'Learn RxJS' website. The left sidebar contains a navigation menu with the following items: Introduction, LEARN RXJS (with a sub-menu open showing Operators, Combination, Conditional, Creation, Error Handling, Multicasting, Filtering, Transformation, Utility, and Full Listing), Subjects, Recipes, and Concepts. The main content area is titled 'Operators' and includes a description: 'A complete list of RxJS operators with clear explanations, relevant resources, and executable examples.' Below this is a link: 'Prefer a complete list in alphabetical order?'. The 'Contents (By Operator Type)' section lists three categories: Combination (with sub-items: combineAll, combineLatest ★, concat ★, concatAll, endWith, forkJoin, merge ★, mergeAll, pairwise, race, startWith ★, withLatestFrom ★, zip), Conditional (with sub-items: defaultIfEmpty, every, iif, sequenceEqual), and Creation.

Learn RxJS

Introduction


LEARN RXJS

- Operators
- Combination
- Conditional
- Creation
- Error Handling
- Multicasting
- Filtering
- Transformation
- Utility
- Full Listing

Subjects

Recipes

Concepts

Powered by  GitBook

Operators

A complete list of RxJS operators with clear explanations, relevant resources, and executable examples.

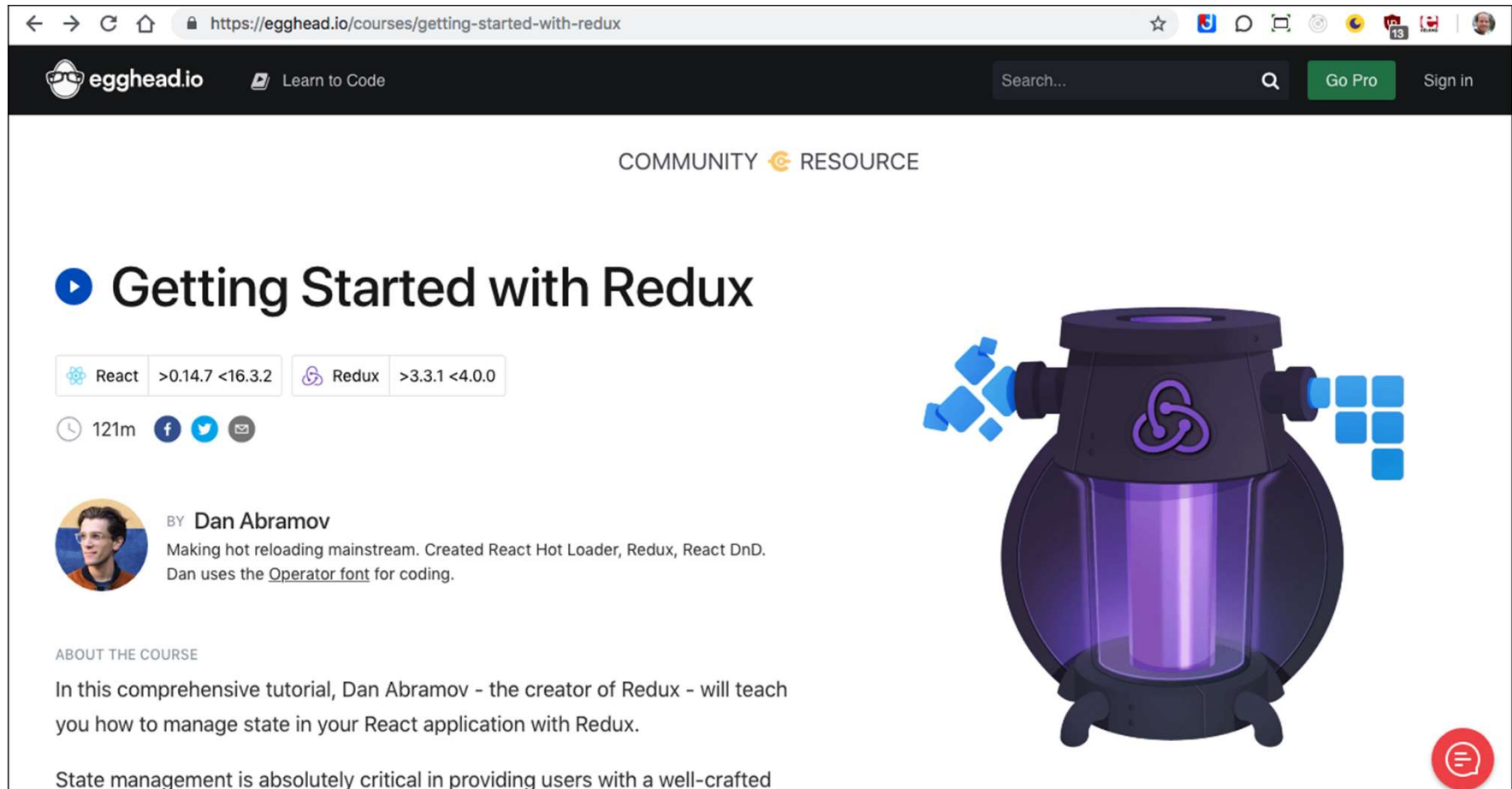
[Prefer a complete list in alphabetical order?](#)

Contents (By Operator Type)

- Combination
 - [combineAll](#)
 - [combineLatest](#) ★
 - [concat](#) ★
 - [concatAll](#)
 - [endWith](#)
 - [forkJoin](#)
 - [merge](#) ★
 - [mergeAll](#)
 - [pairwise](#)
 - [race](#)
 - [startWith](#) ★
 - [withLatestFrom](#) ★
 - [zip](#)
- Conditional
 - [defaultIfEmpty](#)
 - [every](#)
 - [iif](#)
 - [sequenceEqual](#)
- Creation

<https://www.learnrxjs.io/>

Extra info - Free Redux-lessons/video



The screenshot shows a web browser window with the URL <https://egghead.io/courses/getting-started-with-redux>. The page features the egghead.io logo and a navigation bar with a search bar, a 'Go Pro' button, and a 'Sign in' link. Below the navigation bar, there are links for 'COMMUNITY' and 'RESOURCE'. The main content area displays the course title 'Getting Started with Redux' with a play button icon. It also shows the course's compatibility with React (>0.14.7 <16.3.2) and Redux (>3.3.1 <4.0.0). The course is by Dan Abramov, with a brief bio mentioning his work on React Hot Loader, Redux, and React DnD. A large, stylized illustration of a purple and black blender with blue cubes is positioned on the right side of the page. At the bottom left, there is a section titled 'ABOUT THE COURSE' which describes the tutorial as comprehensive and taught by the creator of Redux. A red chat bubble icon is located in the bottom right corner.

egghead.io Learn to Code Search... Go Pro Sign in

COMMUNITY RESOURCE

Getting Started with Redux

React >0.14.7 <16.3.2 Redux >3.3.1 <4.0.0

121m

BY Dan Abramov

Making hot reloading mainstream. Created React Hot Loader, Redux, React DnD.
Dan uses the Operator font for coding.

ABOUT THE COURSE






In this comprehensive tutorial, Dan Abramov - the creator of Redux - will teach you how to manage state in your React application with Redux.

State management is absolutely critical in providing users with a well-crafted

<https://egghead.io/courses/getting-started-with-redux>

Creating an 'ng add' schematic



Nitay Neeman's Blog



Contents

- The Goal
- Initializing a Workspace
- Internal Library
 - Creating the Library
 - Using the Library
- Schematics Project
 - Creating the Project
 - Adding a Schematic
 - Running a Schematic
- Publishing Process
 - Adding npm Scripts
 - Copying Schematics to "dist"
 - Publishing the Package
- Demo Application
- Summary


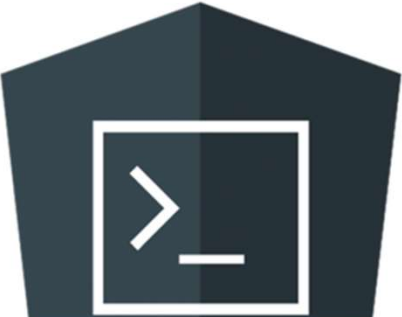

Making an Addable Angular Package Using Schematics

August 11, 2018 | 14 min read |  

Let's try to create an Angular library that's consumable easily by making an "ng add" schematic. This library will provide a simple exported Angular Element.

"**Angular CLI** is the best thing which happened to Angular" - that's how I was thinking for a long time. Simply, it lets you do a lot of stuff with no effort.


But then, Angular announced a new idea, called Angular Labs, which has a purpose of initiating new explorations. Basically, that's the place that **Angular Elements** and **Schematics** came from. In this way, the Angular team made my heart skip a beat once again. 😊




Subscription

Only new content, no spam.

Subscribe

 Follow 718 followers

 YouTube 157

<https://nitayneeman.com/posts/making-an-addable-angular-package-using-schematics/>

Any Questions?

Today - day 3 – **Miscellaneous**

- Store from scratch with `BehaviorSubject()`
- Dynamic Forms
- Reusable/configurable Components, Services
- Angular monorepo's / micro frontends?
- Shared libraries in projects/multiple applications
- ...

Wrap up

- Angular CLI tips & tricks
- NG applications with multiple modules
 - Routing and Lazy loading modules
 - Loading strategies
- Design patterns
- Observables, RxJS operators
- State Management
- Dynamic forms
- Reusable components

Let's connect

duursporten (lopen, zwemmen, fietsen, roeien, schaatsen).
Laatste jaar in combinatie met studie Bewegingswetenschappen, Rijksuniversiteit Groningen (RUG).

[See more education](#) ▾

Featured Skills & Endorsements [Add a new skill](#) ✎

Teaching · 3 Youri Pattynama and 2 connections have given endorsements for this skill

TypeScript · 3 Youri Pattynama and 2 connections have given endorsements for this skill

Angular · 2 Momen El Kamri and 1 connection have given endorsements for this skill

[See 26 more skills](#) ▾

Recommendations [Ask to be recommended](#) ✎

[Add new profile section](#)

[Try Premium for Free](#)

Messaging ✎ ⚙

<https://www.linkedin.com/in/peterkassenaar/>



www.angulartraining.nl