

What are pipe and tap methods in Angular tutorial? [closed]

I'm following the tutorial at <https://angular.io>, and I'm having trouble finding documentation; specifically for the methods `pipe` and `tap`. I can't find anything on <https://angular.io> or <http://reactivex.io/rxjs/>.

99

My understanding is that `pipe` and `tap` are both methods of `Observable`, which is being imported from RxJS, correct? What are they supposed to do?



Where should I be looking for documentation of methods?

18



edited Nov 14 '17 at 5:29



Daniel Kucal

3,788 1 19 41

asked Nov 13 '17 at 23:45



Ben Rubin

2,872 2 16 31

closed as off-topic by [Ken White](#), [HPierce](#), [www](#), [Kirk Beard](#), [Hovercraft Full Of Eels](#) Nov 14 '17 at 1:18

This question appears to be off-topic. The users who voted to close gave this specific reason:

- "Questions asking us to **recommend or find a book, tool, software library, tutorial or other off-site resource** are off-topic for Stack Overflow as they tend to attract opinionated answers and spam. Instead, [describe the problem](#) and what has been done so far to solve it." – [Ken White](#), [HPierce](#), [www](#), [Kirk Beard](#), [Hovercraft Full Of Eels](#)

If this question can be reworded to fit the rules in the [help center](#), please [edit the question](#).

21 I find it strange when users ask proper questions, receive a proper answer, and yet for the mods are clueless what OP is asking :D - why the hell is this "off-topic"? – [Paul Strupeikis](#) Jan 29 at 14:49

1 Answer

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



66 long to paste here) and [pipe](#) operators:

```
/**
 * Used to stitch together functional operators into a chain.
 * @method pipe
 * @return {Observable} the Observable result of all of the operators having
 * been called in the order they were passed in.
 *
 * @example
 *
 * import { map, filter, scan } from 'rxjs/operators';
 *
 * Rx.Observable.interval(1000)
 *   .pipe(
 *     filter(x => x % 2 === 0),
 *     map(x => x + x),
 *     scan((acc, x) => acc + x)
 *   )
 *   .subscribe(x => console.log(x))
 */
```

Pipe: Used to stitch together functional operators into a chain.

Tap: Can perform side effects with observed data but **doesnot modify** the stream in any way.

edited May 18 at 5:50



samlu

156 1 11

answered Nov 14 '17 at 0:49



Daniel Kucal

3,788 1 19 41

-
- 3 Thank you for the answer and the links. Part of my problem is that I'm new to Angular, and I'm not sure which methods are part of core JavaScript or Node.js or RxJS or Angular. Your answer helped me to clarify that. Thank you. — [Ben Rubin](#) Nov 14 '17 at 0:57
-
- 3 @BenRubin I would recommend that you start by learning native JS properly before starting to learn the tools. It will make it much easier to understand the tools and what it actually does (and to know what parts is native vs the tool). — [Magnus Eriksson](#) Nov 14 '17 at 6:44
-
- 15 since 5.5 and the introduction of pipeable (once known as lettable) operators, do as been renamed as `tap` ... In short, it's a mess. github.com/ReactiveX/rxjs/blob/master/doc/pipeable-operators.md — [Luca](#) Mar 12 '18 at 14:22 ✎
-

Can you explain me with an example what every line (filter, map, scan) exactly do? — [Andrea Martinelli](#) Aug 10 '18 at 7:48

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



Facebook

Got a question that you can't ask on public Stack Overflow? [Learn more](#) about sharing private information with Stack Overflow for Teams.



Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google

Facebook 