Reactive Programming

<https://github.com/trungtrong/learn-rxjs>

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

https://www.tiepphan.com/rxjs-reactive-programming/

https://medium.com/@mohandere/rxjs-5-in-5-minutes-1c3b4ed0d8cc

https://gist.github.com/staltz/868e7e9bc2a7b8c1f754

|  |
| --- |
| 1 - { distinctUntilChanged } operator  import { from } from 'rxjs';  import { distinctUntilChanged } from 'rxjs/operators';  // only output distinct values, based on the last emitted value  const source$ = from([1, 1, 2, 2, 3, 3]);  source$  .pipe(distinctUntilChanged())  // output: 1,2,3  .subscribe(console.log);  // 1 => emit  // 1: trung lap, khong emit  // 2 (different) => emit  // 2: trung lap , khong emit |
| 2 - Like array.map: iterate through each item  import { map } from 'rxjs/operators';  //emit (1,2,3,4,5)  const source = from([1, 2, 3, 4, 5]);  //add 10 to each value  const example = source.pipe(map(val => val + 10));  //output: 11,12,13,14,15  const subscribe = example.subscribe(val => console.log(val)); |
| import { interval } from 'rxjs';  import { take } from 'rxjs/operators';  //emit value every 1s  const interval$ = interval(1000);  //take the first 5 emitted values  const example = interval$.pipe(take(5));  //output: 0,1,2,3,4  const subscribe = example.subscribe(val => console.log(val)); |
|  |
|  |