

# Guruh Hapsara

Data Visualization Engineer at Bukalapak

I love Data & UI

24/7 on twitter @guruhhapsara

# Analyzing @jokowi's Word Usage on Twitter

Essay & Code by  
Guruh Hapsara

gusmugusmu  
pagi  
merapat

Eid Mubarak  
to All  
Fellow Muslim

# Going Reactive With RxJs



## What is RxJs ?

a library for composing asynchronous and event-based programs by using observable sequences.

It extends the observer pattern to support sequences of data and/or events and adds operators that allow you to compose sequences together declaratively while abstracting away concerns about things like low-level threading, synchronization, thread-safety, concurrent data structures, and non-blocking I/O.

~ ReactiveX(Rx) docs.

**TV**land  
Classic



# Lodash for Async/Events.

Ben Lesh - RxJs project lead.

# Lodash

```
import { chain } from 'lodash';

const data = [1, 2, 3, 4, 5, 6];

const result = chain(data)
  .filter(d => d % 2 === 0)
  .map(d => `${d}, genap ya`);

console.log(result.value());

/* output
[
  '2, genap ya',
  '4, genap ya',
  '6, genap ya'
]
```

# RxJs

3 main parts

## 1. Observable

A stream of values (numbers, clicks, ajax request, websocket) over time.

## 2. Operator

set of functions for transforming observable

## 3. Subscriber

The observable subscriber, observable wont do anything unless you subscribe to it.

# RxJs

```
● ● ●

import { of as of$ } from 'rxjs';
import { map, filter } from 'rxjs/operators';

const observable$ = of$(1, 2, 3, 4, 5, 6);

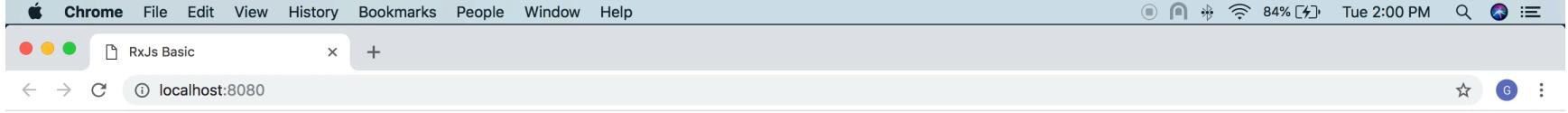
const operation$ = observable$.pipe(
  filter(d => d % 2 === 0),
  map(d => `${d},genap ya`),
);

operation$.subscribe(
  next => console.log(next),
  error => console.log('error', error),
  () => console.log('completed'),
);

/* output
'2, genap ya',
'4, genap ya',
'6, genap ya'
*/
```

Let's build a redux-like state  
management with RxJs





# Redux-like state management in RxJs

```
const counterElement = document.getElementById('counter-element');
const incrementEvent$ = fromEvent(document.getElementById('increment-button'), 'mousedown');
const decrementEvent$ = fromEvent(document.getElementById('decrement-button'), 'mousedown');

const increment$ = incrementEvent$.pipe(
  map(() => 1)
);

const decrement$ = decrementEvent$.pipe(
  map(() => -1)
);

const observable$ = merge(
  increment$,
  decrement$,
);

const counter$ = observable$.pipe(
  startWith(0),
  scan((curr, next) => curr + next)
);

counter$.subscribe(state => {
  counterElement.innerText = state;
});
```

142



Let's take it to the next level  
make it behaves like medium's claps

## Mouse Events

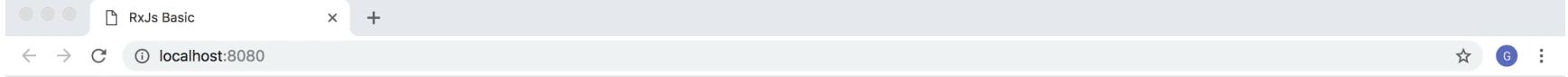
Event Name	Fired When
<code>mouseenter</code>	A pointing device is moved onto the element that has the listener attached.
<code>mouseover</code>	A pointing device is moved onto the element that has the listener attached or onto one of its children.
<code>mousemove</code>	A pointing device is moved over an element. (Fired continuously as the mouse moves.)
<code>mousedown</code>	A pointing device button is pressed on an element.
<code>mouseup</code>	A pointing device button is released over an element.
<code>auxclick</code>	A pointing device button (ANY non-primary button) has been pressed and released on an element.
<code>click</code>	A pointing device button (ANY button; soon to be primary button only) has been pressed and released on an element.
<code>dblclick</code>	A pointing device button is clicked twice on an element.
<code>contextmenu</code>	The right button of the mouse is clicked (before the context menu is displayed).
<code>wheel</code>	A wheel button of a pointing device is rotated in any direction.
<code>mouseleave</code>	A pointing device is moved off the element that has the listener attached.
<code>mouseout</code>	A pointing device is moved off the element that has the listener attached or off one of its children.
<code>select</code>	Some text is being selected.
<code>pointerlockchange</code>	The pointer was locked or released.
<code>pointerlockerror</code>	It was impossible to lock the pointer for technical reasons or because the permission was denied.

## Make it likely holding

```
const mouseUp$ = fromEvent(document, 'mouseup');

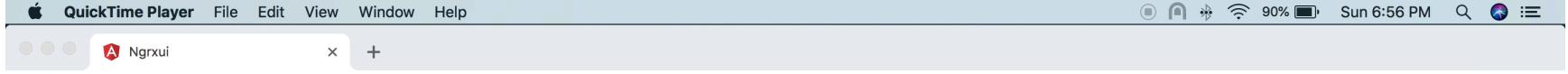
// previous code
const increment$ = incrementEvent$.pipe(
  map(() => 1),
);

// modification
const increment$ = incrementEvent$.pipe(
  switchMap(() => {
    return timer(0, 100).pipe(
      map(() => 1),
      takeUntil(mouseUp$)
    );
  }),
);
```



Let's make a pull to refresh simulation  
with Angular + RxJs





..... at 70% of window height .....

10%

drawer position

# Here is the business

1. Drag the drawer to bottom
2. Release mouse will pull the drawer up

```
const element = document.getElementById('pull-element');

const mouseDown$ = fromEvent<MouseEvent>(element, 'mousedown');
const mouseMove$ = fromEvent<MouseEvent>(document, 'mousemove');
const mouseUp$ = fromEvent<MouseEvent>(document, 'mouseup');

const animationFrame$ = interval(0, animationFrame);

const pullDown$ = mouseDown$.pipe(
  switchMap(() => {
    return mouseMove$.pipe(
      filter((d: MouseEvent) => d.y > this.minY),
      map((d: MouseEvent) => d.clientY),
      takeUntil(mouseUp$)
    );
  })
);

const pullUp$ = mouseUp$.pipe(
  exhaustMap(() => {
    return animationFrame$.pipe(
      filter(() => this.y > this.minY),
      map((t: number) => lerp(this maxY, this.minY, t / this.frameCount)),
      takeWhile((y: number) => y >= this.minY)
    );
  })
);

pullDown$.subscribe((y: number) => {
  // handle state
});

pullUp$.subscribe((y: number) => {
  // handle state
});
```

Let's build a NPM search in VueJs + RxJs



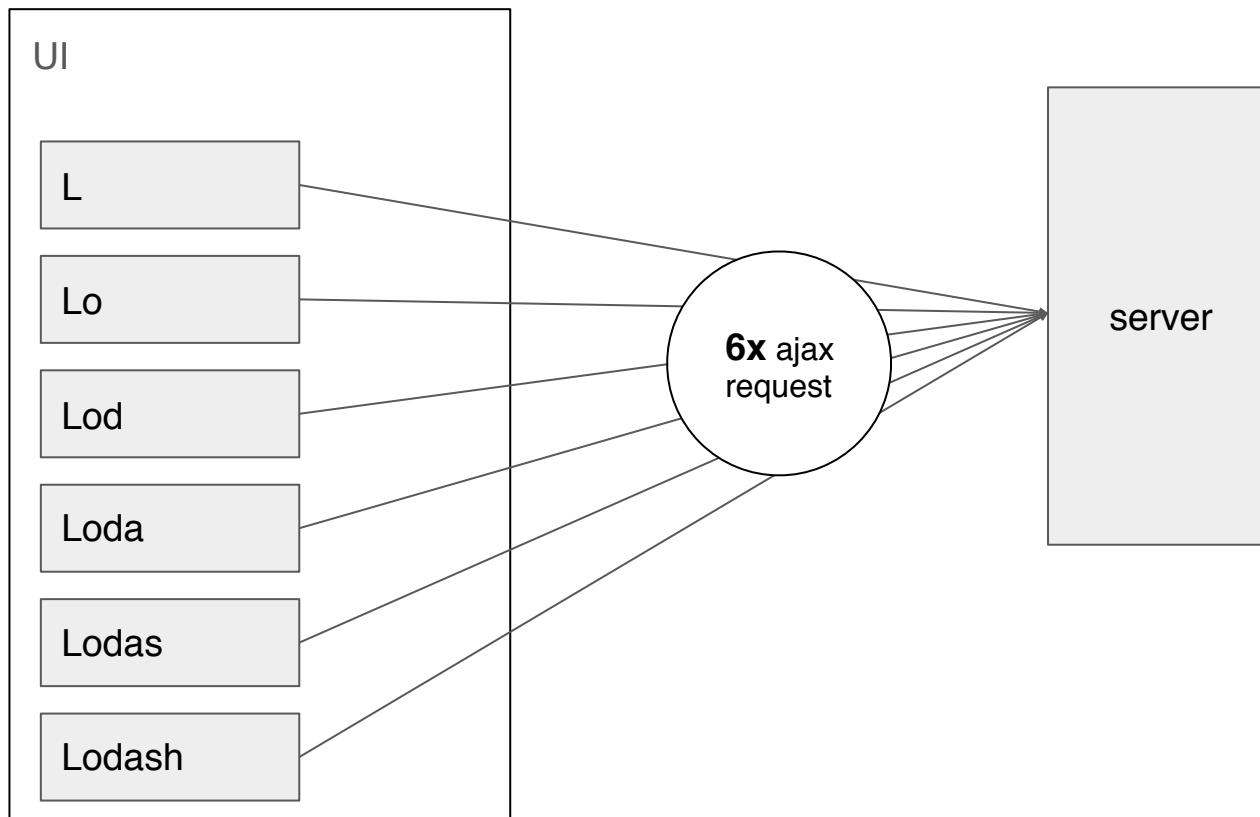
## Use Case

1. Only perform search request when user stops typing.
2. Cancel the ongoing search request when keywords are changed.

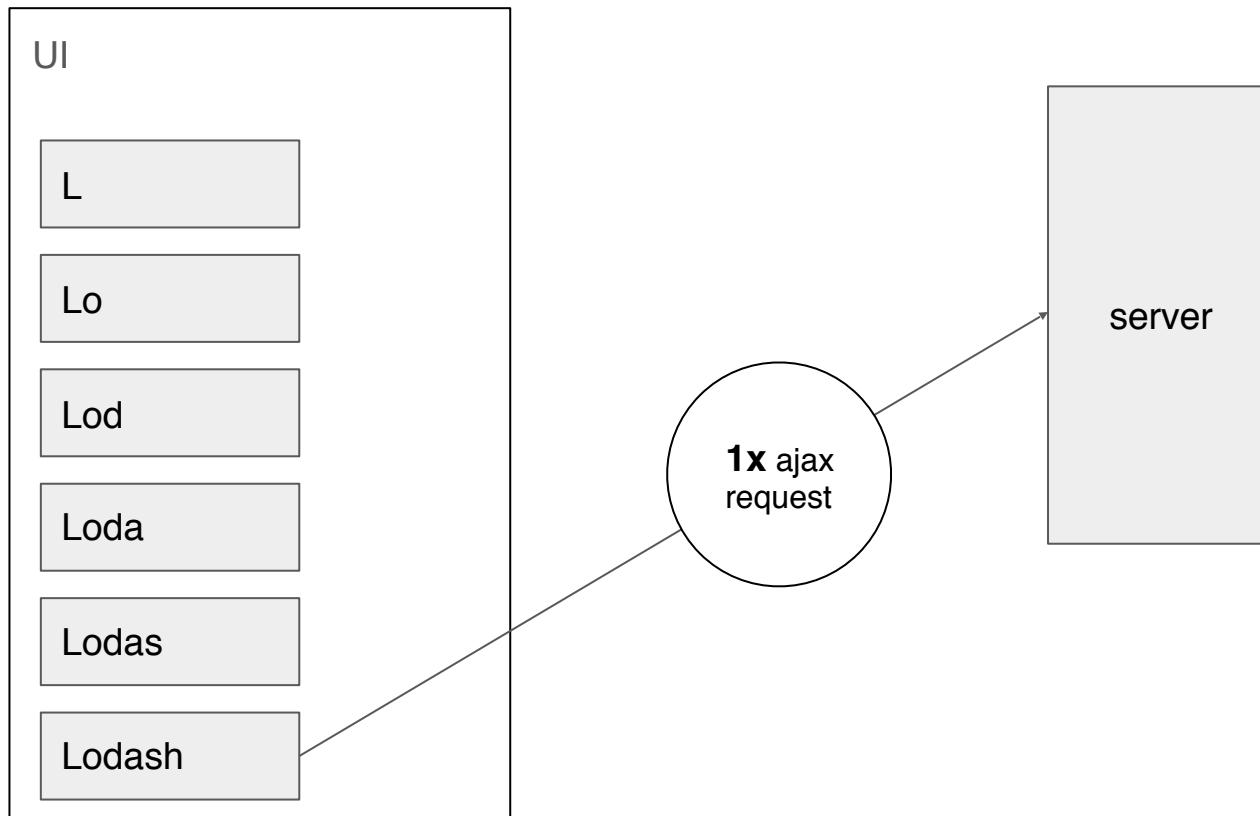
Only perform search request when user stops typing.

why?

## Search as keyword changed



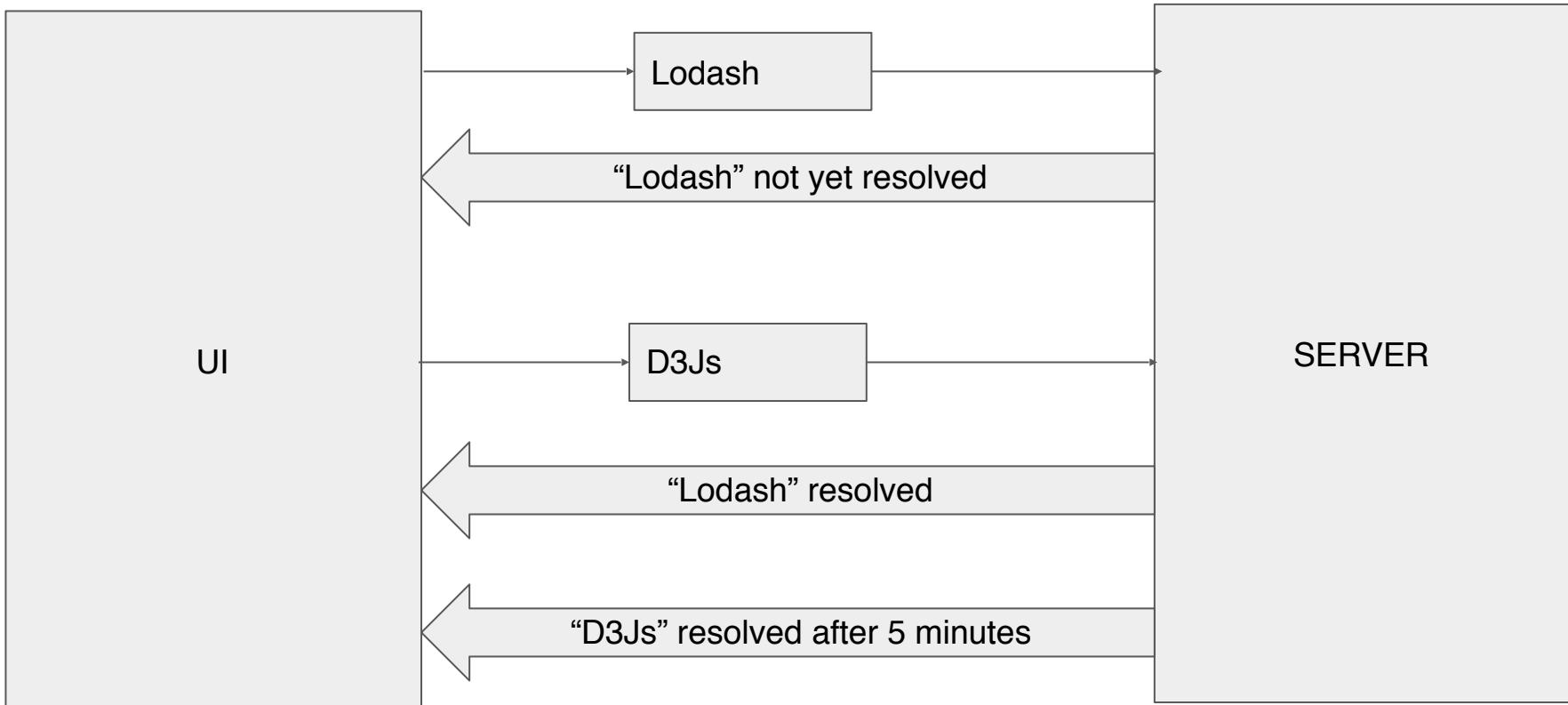
## Search as user stops typing



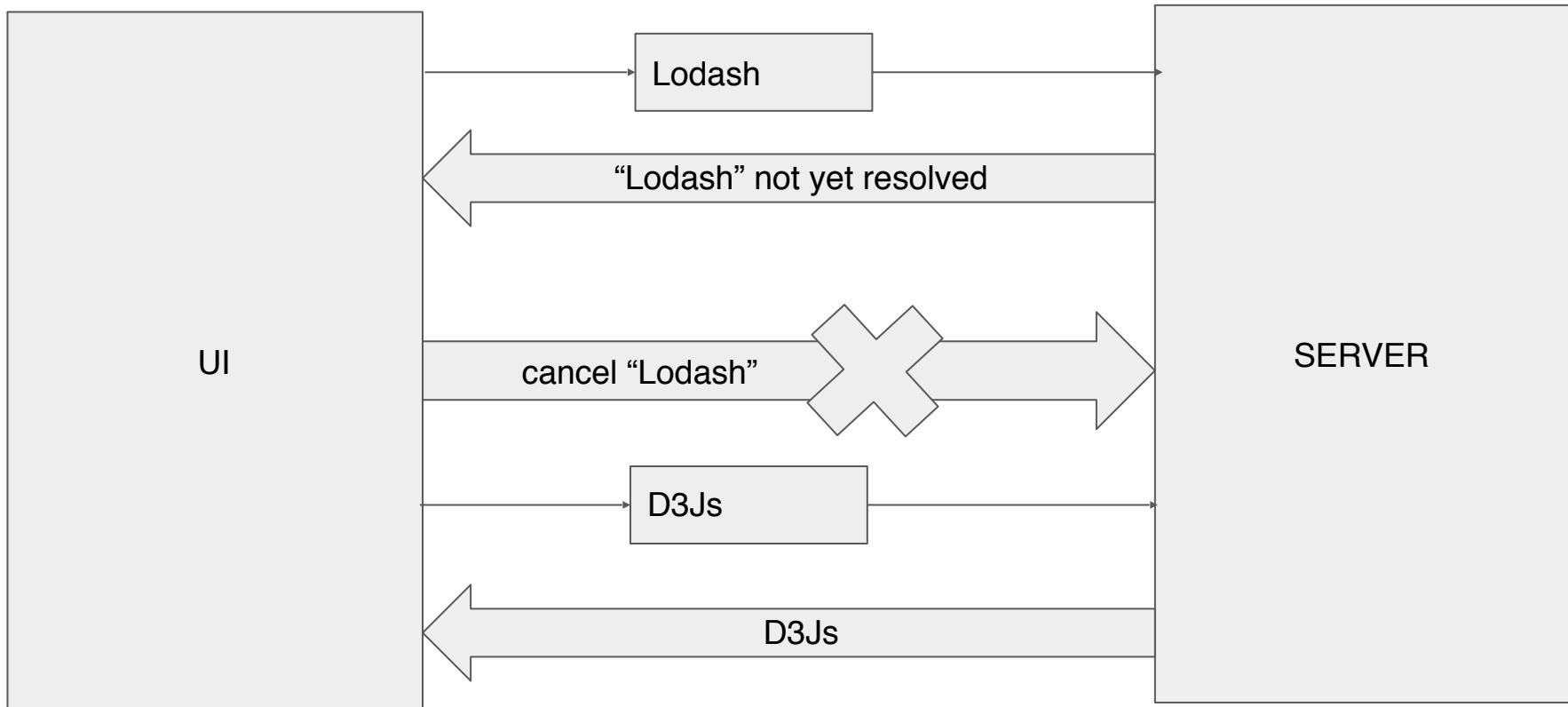
Cancel the ongoing search request when keywords are changed.

Why do we need ajax cancellation?

## Search keyword without cancellation



## Search keyword with cancellation



localhost:8080

G

## Package Explorer



rxjs @6.3.2

@rxjs/rx @4.1.0

rxjs-react @0.2.0

rxjs-ratelimiter @0.1.1

rxjs-reselect @1.1.1

rxjs-routing @1.0.1

rxjs-obd @0.0.8

rxjs-spy @7.2.1

rxjs-ts @5.0.0-alpha.10

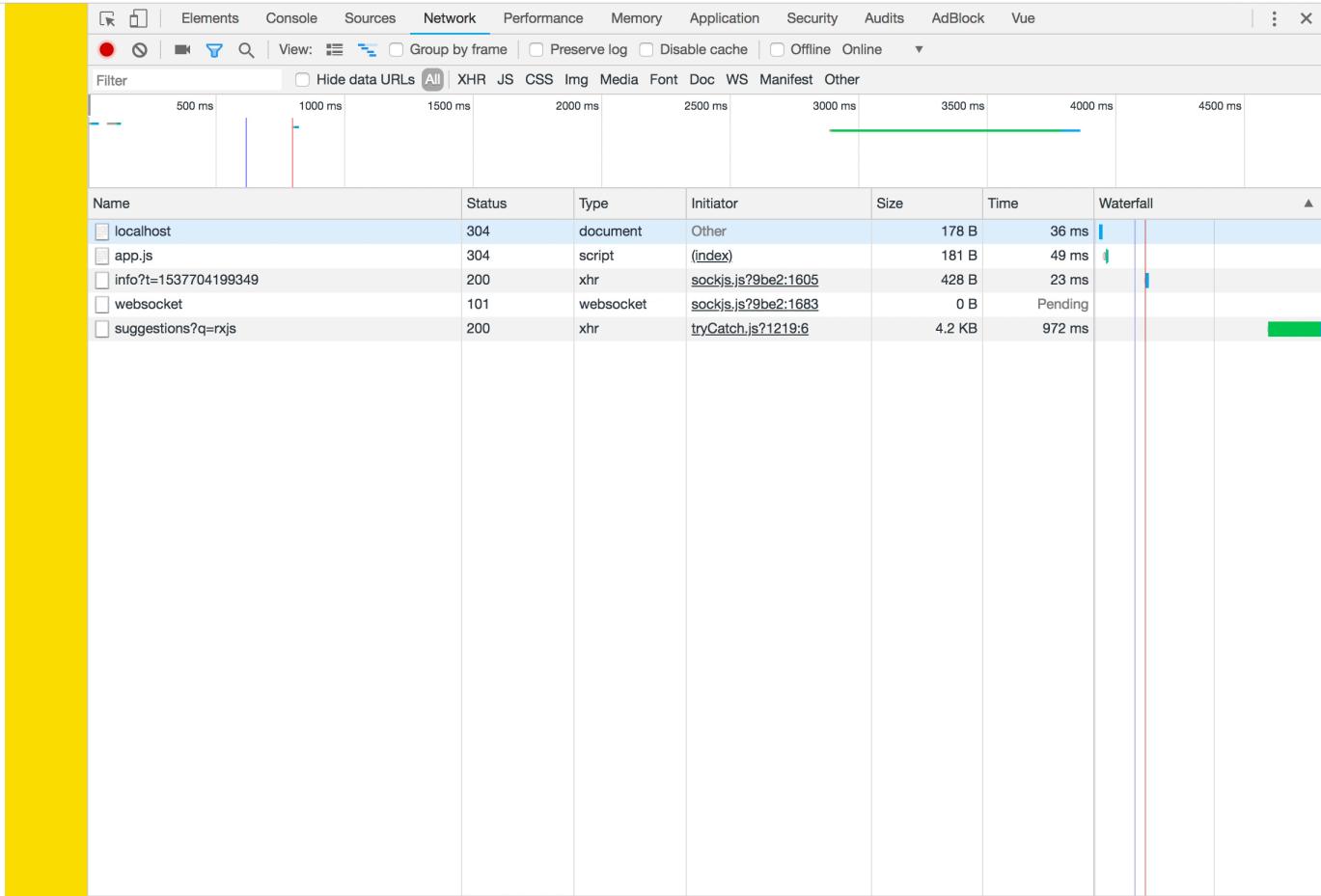
rxjs-wx @1.1.1

rxjs-aws @0.1.29

rxjs-csp @0.2.1

rxjs-es @5.0.0-beta.12

rxjs-dew @1.0.1



## Use Case

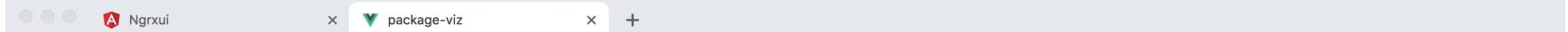
1. Only perform search request when user stops typing.
2. Cancel the ongoing search request when keywords are changed.

```
<template>
<div>
  <input class="search-input" placeholder="type package" v-stream:input="input$" />
  <!-- <FancyComponent /> -->
</div>
</template>

<script>
import { debounceTime, pluck, map, switchMap, filter, takeUntil } from 'rxjs/operators';
import { ajax } from 'rxjs/ajax';

export default {
  // ...vue stuffs
  domStreams: ['input$'],
  subscriptions() {
    const inputStream = this.input$.pipe(
      debounceTime(1000),
      pluck('event', 'target', 'value'),
      filter(keyword => keyword.length > 0),
      map(keyword => this.composeSearchUrl(keyword)),
      switchMap(url =>
        ajax
          .get(url)
          .pipe(
            map(r => this.mapSearchResult(r.response)),
            takeUntil(this.input$)
          )
      )
    );
    return {
      searchResult: inputStream
    };
  }
};
</script>

<style scoped>
/* fancy css */
</style>
```



## Package Explorer

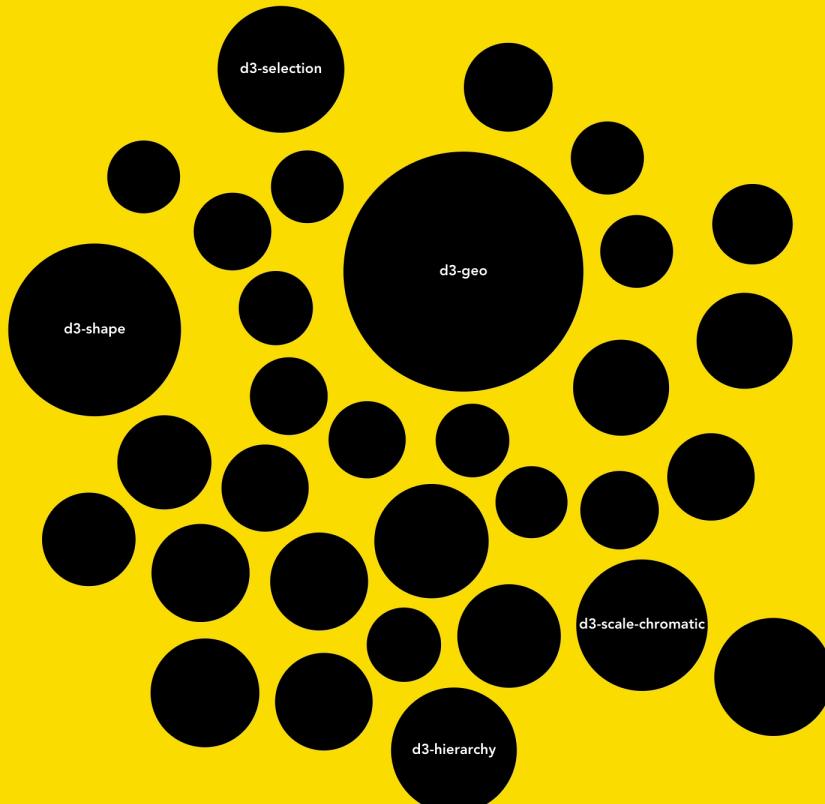
### d3

<b>d3</b>	@5.7.0
d3-3d	@0.0.9
d3-3	@0.0.0
d3-3.5.17	@1.0.0
d3-force-3d	@1.1.0
d3-dag	@0.1.2
d3-decompose	@1.0.2
d3-delaunay	@4.1.5
d3-dispatch	@1.0.5
d3-downloadable	@2.1.0
d3-dragrect	@0.0.2
d3-drag	@1.2.3
d3-dtree	@2.0.2
d3-donut	@1.0.0

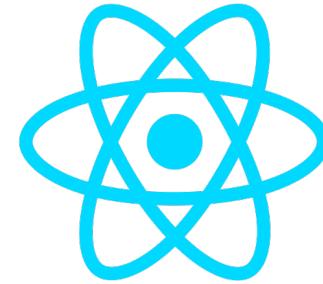
# d3

300.25 kB

31 dependencies



Animating 3000 elements in React



# Analyzing @jokowi's Word Usage on Twitter

Essay & Code by  
Guruh Hapsara

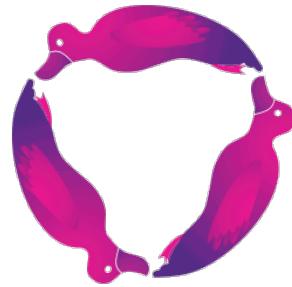
React Motion staggered motions

Canvas

WebGL

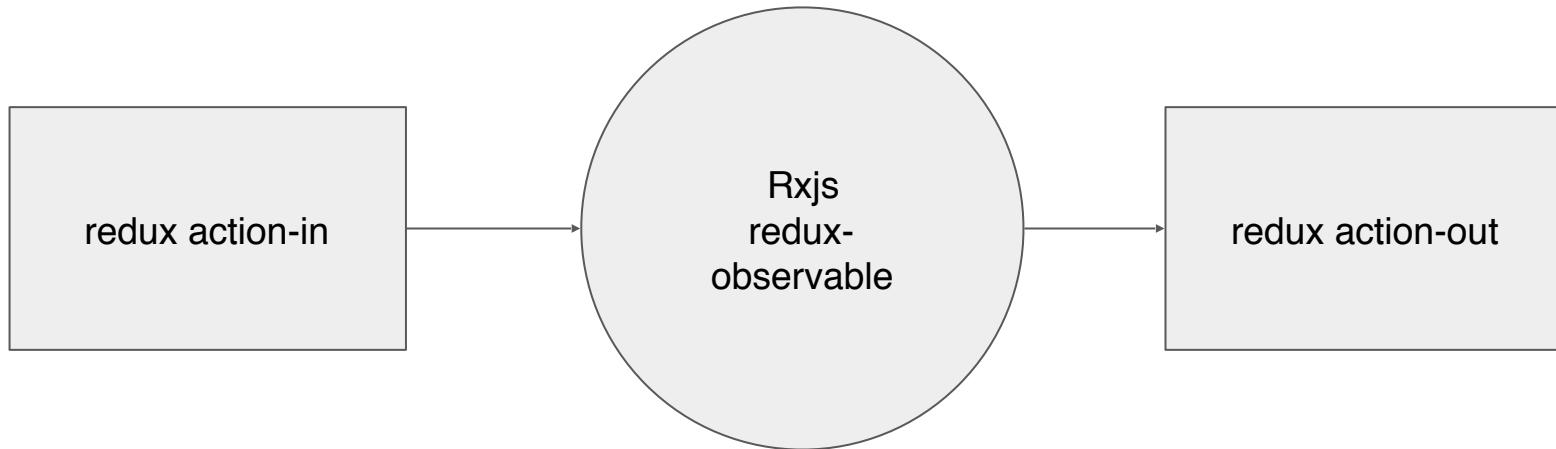
Using RxJs  
to control animation state

RxJs x React x Redux  
with redux-observable



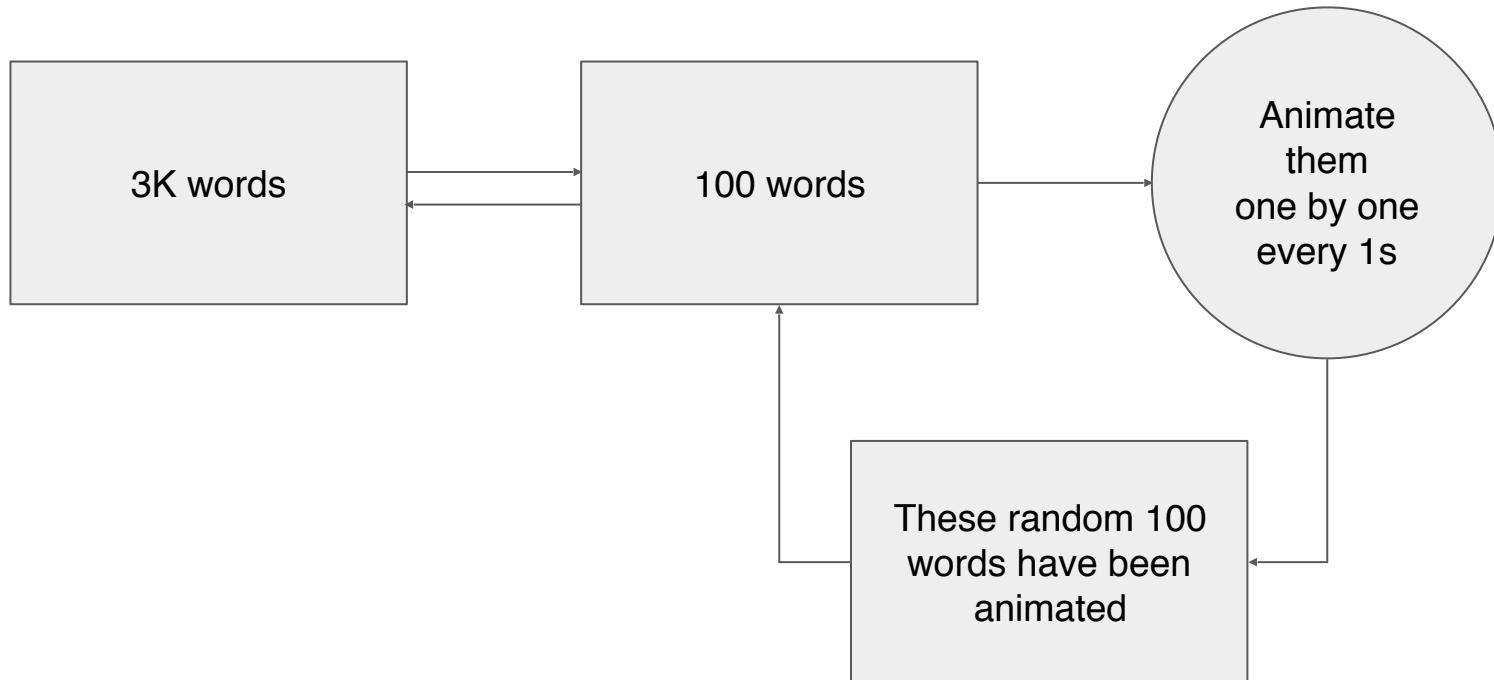
# redux-observable

crash course



## back to use case

animating 3K words



# Epic

1. Every 1s add 1 word to state.
2. Reset the index after it reach 100

```
export const startAddIndexEpic = (action$) => {
  return action$.pipe(
    ofType(types.START_ADD_INDEX),
    switchMap(() =>
      timer(2000, 1000).pipe(
        takeUntil(action$.pipe(
          ofType(types.STOP_ADD_INDEX)
        )),
        map(() => addIndex())
      )
    )
  );
};

export const addIndexEpic = (action$, state$) => {
  return action$.pipe(
    ofType(types.ADD_INDEX),
    filter(d => state$.index > 100),
    mergeMap(() =>
      Observable.of(
        stopAddIndex(),
        startAddIndex()
      )
    )
  );
};
```

Stop the animation  
when the section is  
not visible

&

Start when it's visible

```
● ● ●

export const stopAddIndexOnScrollEpic = (action$, state$) => {
  return action$.pipe(
    ofType(appTypes.ON_SCROLL),
    filter(action => {
      const { windowScrollY } = action.payload;
      const { sectionBoundary } = state$;

      const shouldStop = calculateWhenItShouldStop(windowScrollY, sectionBoundary);

      return shouldStop;
    }),
    map(() => stopAddIndex())
  );
};

export const startAddIndexOnScrollEpic = (action$, state$) => {
  return action$.pipe(
    ofType(appTypes.ON_SCROLL),
    filter(action => {
      const { windowScrollY } = action.payload;
      const { sectionBoundary } = state$;

      const shouldStart = calculateWhenItShouldStart(windowScrollY, sectionBoundary);

      return shouldStart;
    }),
    map(() => startAddIndex())
  );
};
```

## Redux observable middleware setup

```
const rootEpic = combineEpics(  
  addIndexEpic,  
  startAddIndexEpic,  
  startAddIndexOnScrollEpic,  
  stopAddIndexOnScrollEpic  
);  
  
// configure redux observable middleware  
const epicMiddleware = createEpicMiddleware();  
  
export default function configureStore(initialState) {  
  const store = createStore(  
    // redux stuffs  
    compose(  
      applyMiddleware(epicMiddleware)  
    )  
  );  
  
  epicMiddleware.run(rootEpic);  
  
  return store;  
}
```

Analyzing @jokowi's Word Usage



https://ghapsara.github.io/analyzing-jokowi-word-usage/



G

⋮

# Analyzing @jokowi's Word Usage on Twitter

Essay & Code by  
Guruh Hapsara

dimanapun

kinerja

tausiah

muara

roda

disertai

coba

Elements Console Sources Network Performance

```
<!doctype html>
<html lang="en">
  <head>...</head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root">
      ... <div> == $0
      <div>
        <div class="sc-VigVT TNKNo">
          <div>
            <div style="top: 798px; transform: rotate(0deg); position: absolute; width: 899px; display: flex; justify-content: center;">...
          </div>
          <div style="top: 798px; transform: rotate(0deg); position: absolute; width: 899px; display: flex; justify-content: center;">...
          </div>
          <div style="top: 798px; transform: rotate(0deg); position: absolute; width: 899px; display: flex; justify-content: center;">...
          </div>
          <div style="top: 798px; transform: rotate(0deg); position: absolute; width: 899px; display: flex; justify-content: center;">...
          </div>
          <div style="top: 798px; transform: rotate(0deg); position: absolute; width: 899px; display: flex; justify-content: center;">...
          </div>
          <div style="top: 798px; transform: rotate(0deg); position: absolute; width: 899px; display: flex; justify-content: center;">...
          </div>
          <div style="top: 798px; transform: rotate(0deg); position: absolute; width: 899px; display: flex; justify-content: center;">...
          </div>
          <div style="top: 798px; transform: rotate(0deg); position: absolute; width: 899px; display: flex; justify-content: center;">...
          </div>
          <div style="top: 798px; transform: rotate(0deg); position: absolute; width: 899px; display: flex; justify-content: center;">...
          </div>
          <div style="top: 797.986px; transform: rotate(0deg); position: absolute; width: 899px; display: flex; justify-content: center;">...
          </div>
          <div style="top: 797.985px; transform: rotate(0deg); position: absolute; width: 899px; display: flex; justify-content: center;">...
          </div>
```

html body div#root div

Styles Event Listeners DOM Breakpoints Properties Accessibility Adguard

Filter

:hov .cls +

## Why you might want to use RxJs ?

Make complex async/events more manageable.

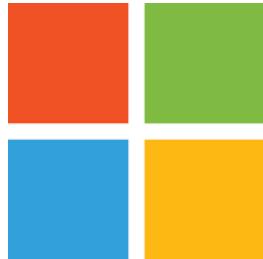
Easy to reason about code.

Transferable skill.

Reactive Extensions(Rx) is also available for these platform :

RxJava, RxKotlin, RxSwift, RxPython, RxGo, Rx.rb(ruby), reactivex (elixir).

# Who uses RxJs?



## Where do we go from here?

- [The introduction to Reactive Programming you've been missing](#) by Andre Stalz
- [learnrxjs.io](#)
- [So You Want to be a Functional Programmer](#) by Charles Scalfani

