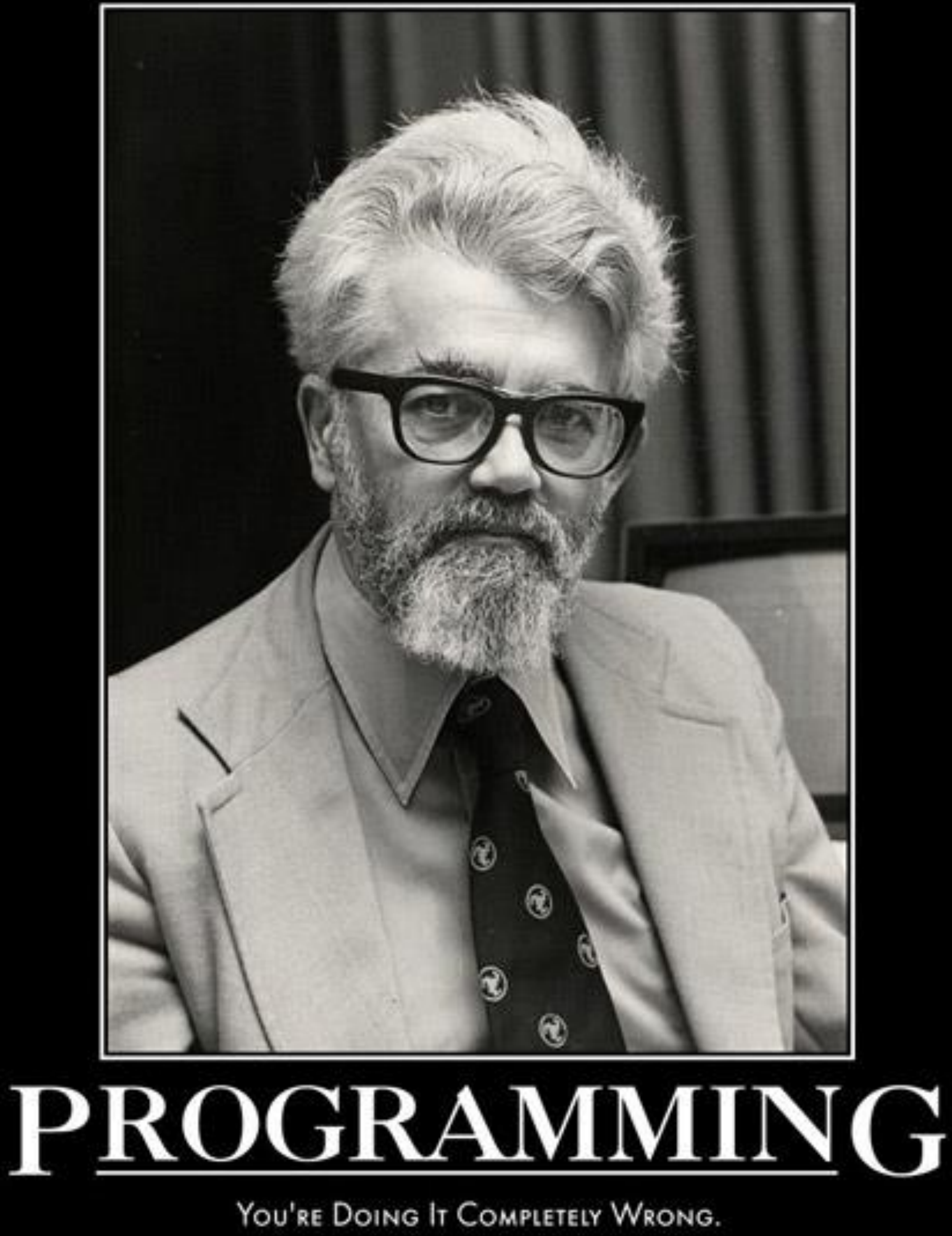


Learn You Some Reactive Programing for Great Good!

Matthew Podwysocki @mattpodwysocki

github.com/mattpodwysocki/brooklynjs

OR:
HOW I LEARNED TO STOP WORRYING ABOUT ASYNCHRONOUS AND EVENT-BASED PROGRAMMING AND LOVE THE OBSERVABLE

Or "I thought I had a problem. I thought to myself, "I know, I'll solve it with callbacks and events!". have Now problems. two I

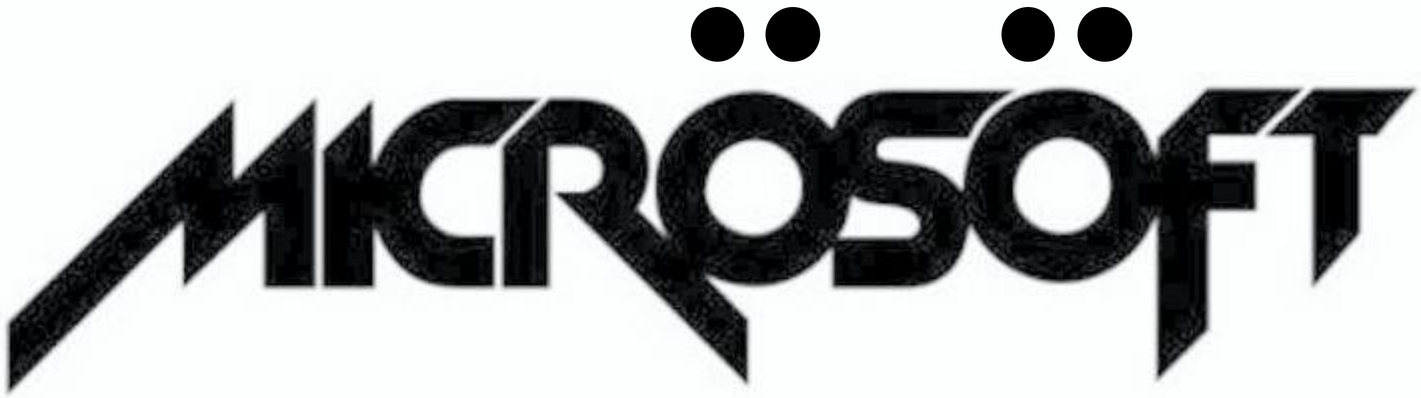
trapd in Monad tutorl
plz help

Software Engineer

Open Sourcerer

@mattpodwysocki

github.com/mattpodwysocki


MICRÖSÖFT

Reactive Extensions (Rx)

@ReactiveX

http://reactivex.io

streem prosesing

Real-Time is Everywhere...

“We choose to go to solve asynchronous programming and do the other things, not because they are easy, but because they are hard”

Former US President John F. Kennedy - 1962
[citation needed]

Callback Hell

```
function play(movieId, callback) {
    var movieTicket, playError,
        tryFinish = function () {
            if (playError) {
                 callback(playError);
            } else if (movieTicket && player.initialized) {
                 callback(null, ticket);
            }
        };
    if (!player.initialized) {
        player.init(function (error) {
            playError = error;
            tryFinish();
        }
    }
    authorizeMovie( function (error, ticket) {
        playError = error;
        movieTicket = ticket;
        tryFinish();
    });
});
```

culturepub.fr
next
ad ⇨

Events and the Enemy of the State

```
var isDown = false, state;

function mousedown (e) {
  isDown = true;
  state = { startX: e.offsetX,
            startY: e.offsetY; }
}

function mousemove (e) {
  if (!isDown) { return; }
  var delta = { endX: e.clientX - state.startX,
                endY: e.clienyY - state.startY };
  // Now do something with it
}

function mouseup (e) {
  isDown = false;
  state = null;
}
```

```
function dispose() {
  elem.removeEventListener('mousedown', mousedown, false);
  elem.removeEventListener('mouseup', mouseup, false);
  doc.removeEventListener('mousemove', mousemove, false);
}

elem.addEventListener('mousedown', mousedown, false);
elem.addEventListener('mouseup', mouseup, false);
doc.addEventListener('mousemove', mousemove, false);
```

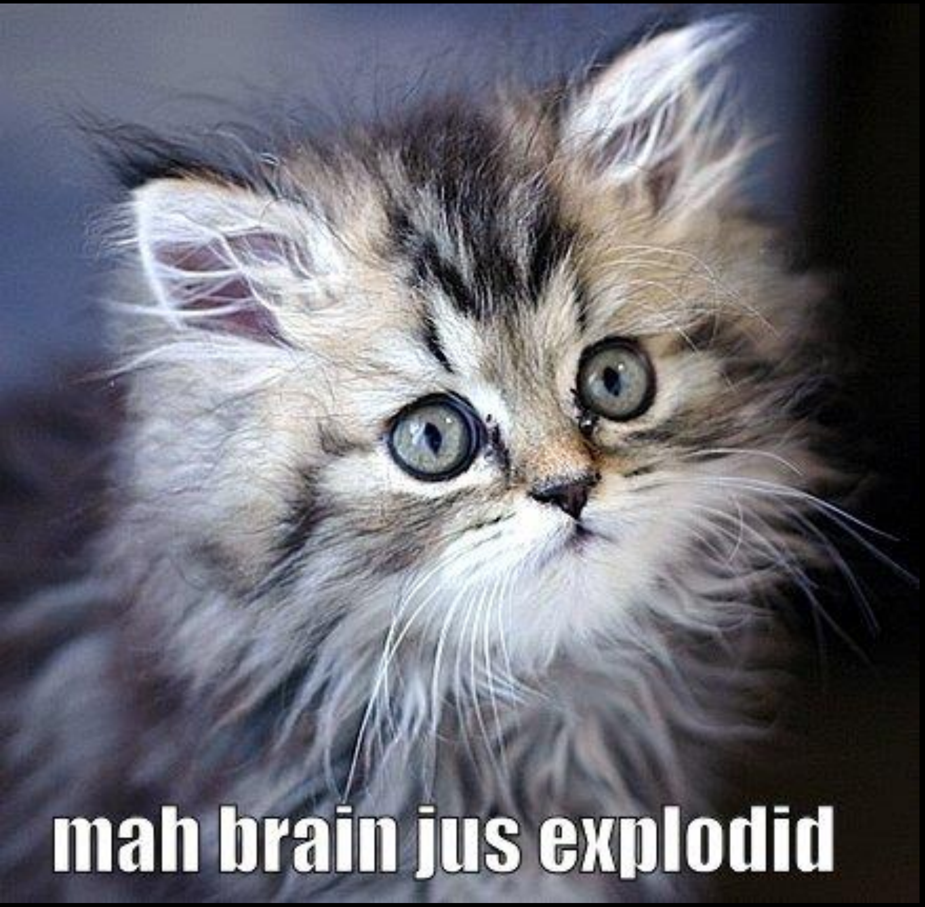
Top-rated Movies Collection

```
var getTopRatedFilms = function (user) {
  return user.videoLists
    .flatMap(function (videoList) {
      return videoList.videos
        .filter(function (v) { return v.rating === 5; });
    });
}

getTopRatedFilms(me)
  .forEach(displayMovie);
```

What if I told you…

…that you could create a drag event…

…with the almost the *same code*



Mouse Drags Collection

```
var getElementDrags = function (elmt) {
  return dom.mousedown(elmt)
    .flatMap(function (md) {
      return dom.mousemove(document)
        ~~.filter~~  .takeUntil(dom.mouseup(elmt));
    });
};

getElementDrags(image)
  .forEach(moveImage)
```

Everything is a stream

Crossing the streams…

Oh yes we did!

```
var connectEvent = fromEvent(e, 'connect'),
    dataEvent = fromEvent(e, 'data'),
    errorEvent = fromEvent(e, 'error'),
    endEvent = fromEvent(e, 'end');

var stream = dataEvent
  .merge(errorEvent.flatMap(throwError))
  .skipUntil(connectEvent.amb(errorEvent))
  .takeUntil(endEvent.amb(errorEvent));
```

You already know how to do this….

INTERACTIVE

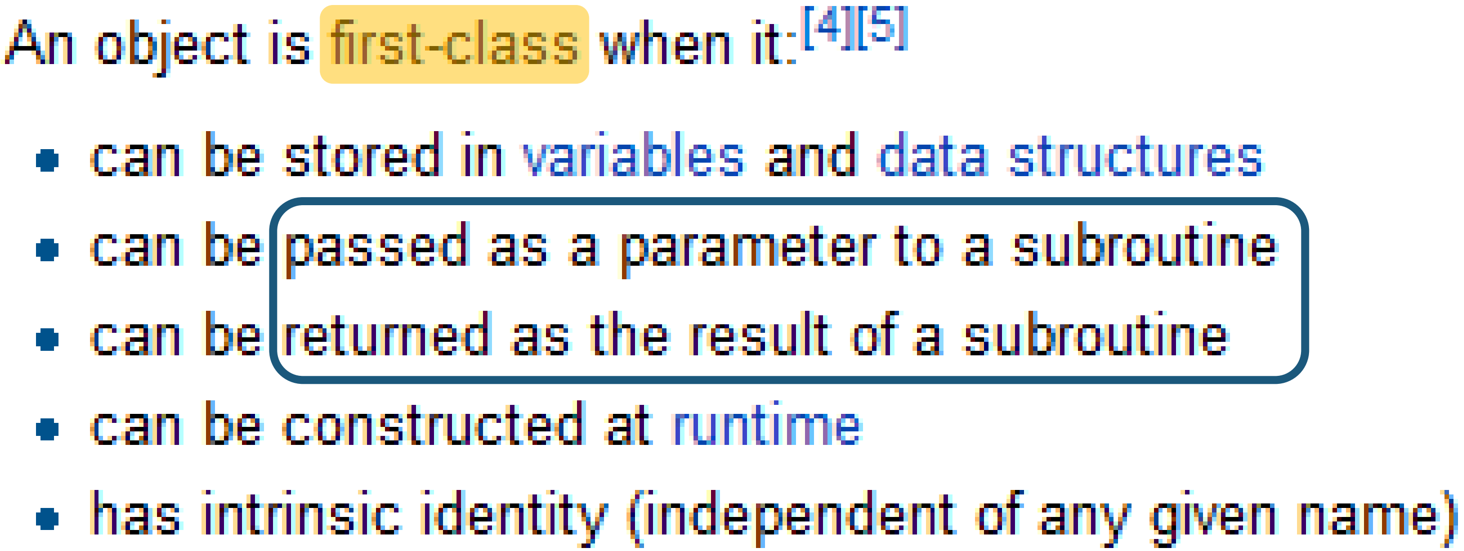
```
var source = getStockData();

source
  .filter(function (quote) {
      return quote.price > 30;
  })
  .map(function (quote) {
      return quote.price;
  })
  .forEach(function (price) {
    console.log(‘Higher than $30: $' + price);
  });
```

REACTIVE

```
var source = getStockData();

source
  .filter(function (quote) {
      return quote.price > 30;
  })
  .map(function (quote) {
      return quote.price;
  })
  .forEach(function (price) {
    console.log(‘Higher than $30: $' + price);
  });
```

First-Class Async and Events

Observables to the rescue



An object is first-class when it:[4][5]

- can be stored in variables and data structures
- can be passed as a parameter to a subroutine
- can be returned as the result of a subroutine
- can be constructed at runtime
- has intrinsic identity (independent of any given name)

WIKIPEDIA
The Free Encyclopedia

What is Reactive Programming Anyhow?

Merriam-Webster defines reactive as *“readily responsive to a stimulus”*,
i.e. its components are “active” and always ready to receive events.

Wanna really know what Reactive Programming Is?

Real Time Programming: Special Purpose or General Purpose Languages
Gerard Berry

http://bit.ly/reactive-paper

Why Promises Aren’t Enough

Problems in Promiseland

- How do I handle cancellation?
- What if I don’t care about the return value ala Autocomplete?

```
var promise;

input.addEventListener('keyup', function (e) {

  if (promise) {
    // Um, how do I cancel?
  } else {
    promise = getData(e.target.value).then(populateUI);
  }
}, false);
```

Observables - Composing Events and Promises

```
var words = input.keyup
            .map(function() { return input.value; })
            .throttle(500)
            .distinctUntilChanged()
            .flatMapLatest(
               function(term) { return search(term); }
            );

words.subscribe(function(data) {
  // Bind data to the UI
});
```

DOM events as a sequence of strings

Reducing data traffic / volume

Latest response as word arrays

Web service call returns single value sequence

Binding results to the UI

Functional Reactive Programming (FRP) is…

A concept consisting of

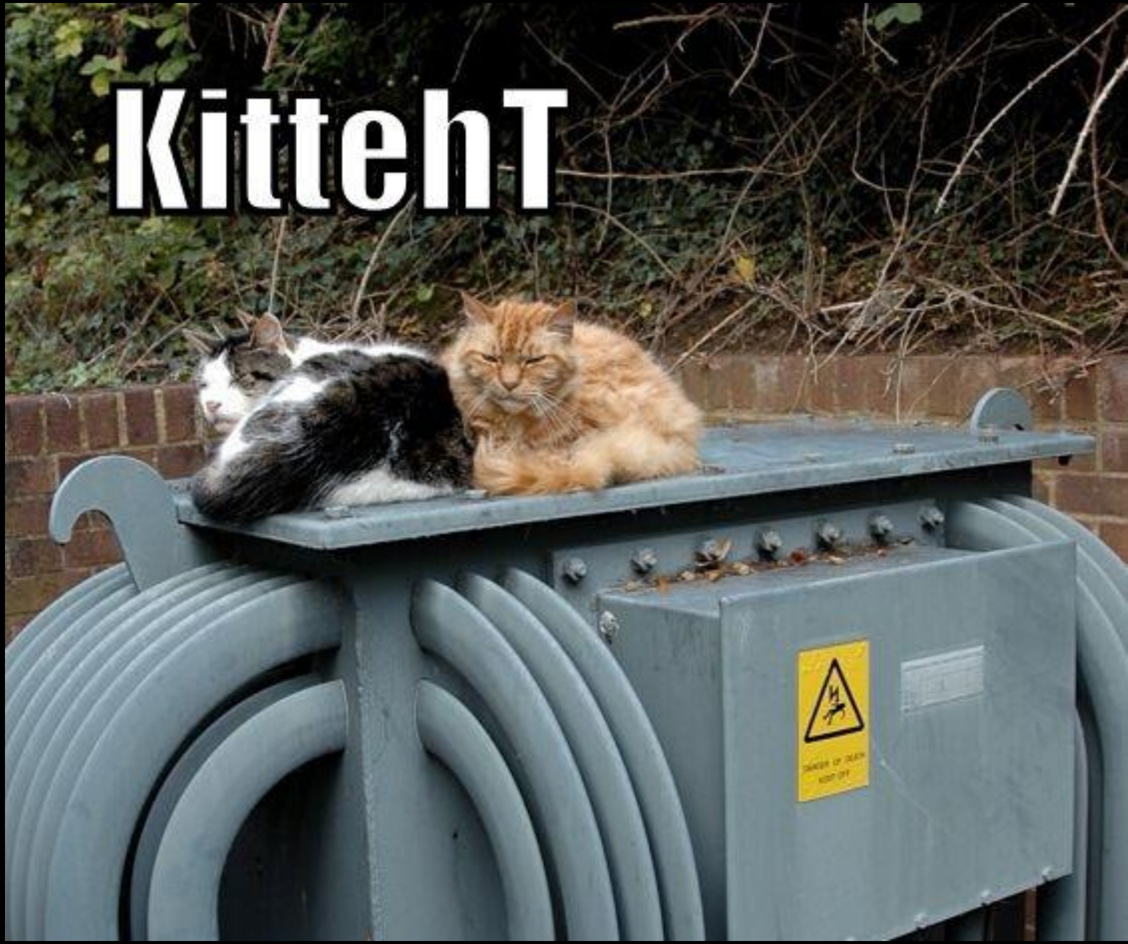
- Continuous Time
- Behaviors: Values over time
- Events: Discrete phenomena with a value and a time
- Compositional behavior for behavior and events

What it is not

- High order functions on events like map, filter, reduce
- Most so-called FRP libraries out there…

http://stackoverflow.com/questions/1028250/what-is-functional-reactive-programming

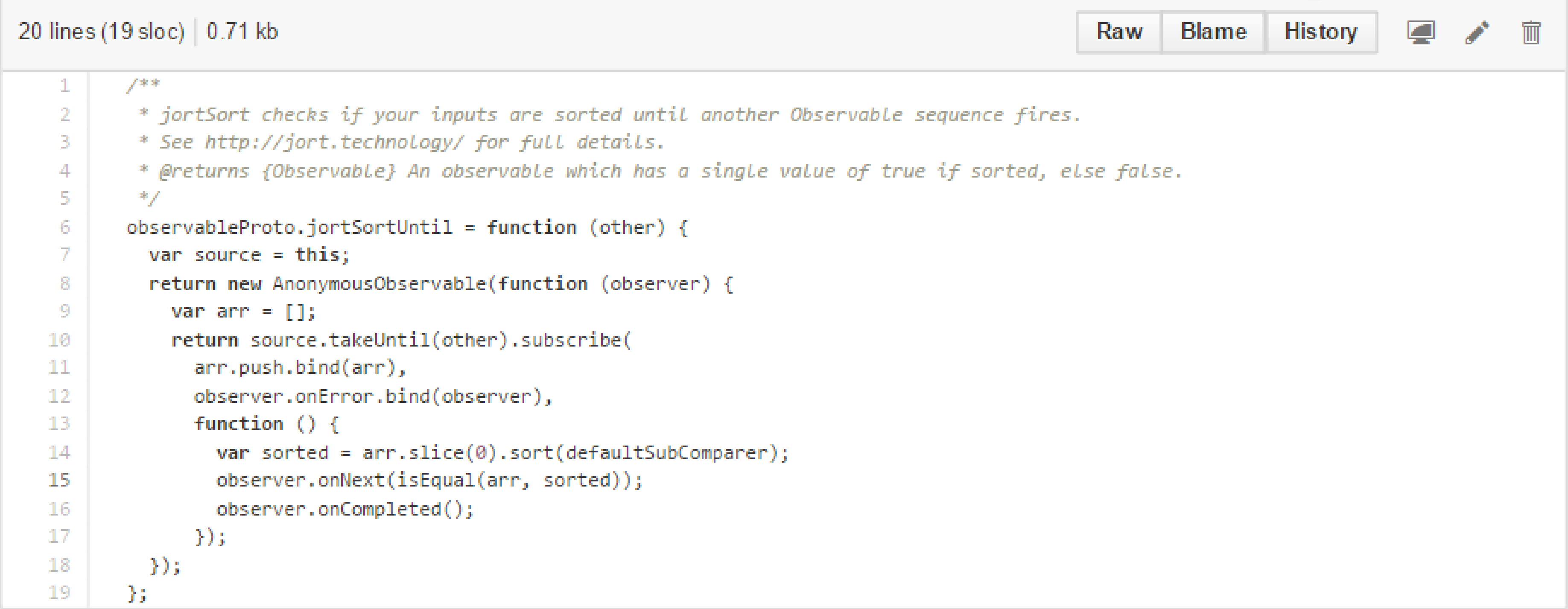
```
Rx.spawn(function* () {
  var result = yield get('brooklynjs.com')
    .retry(3)
    .catch(cachedVersion);

  console.log(result);
}());
```



Oh, and one more thing...

We have jortSort!

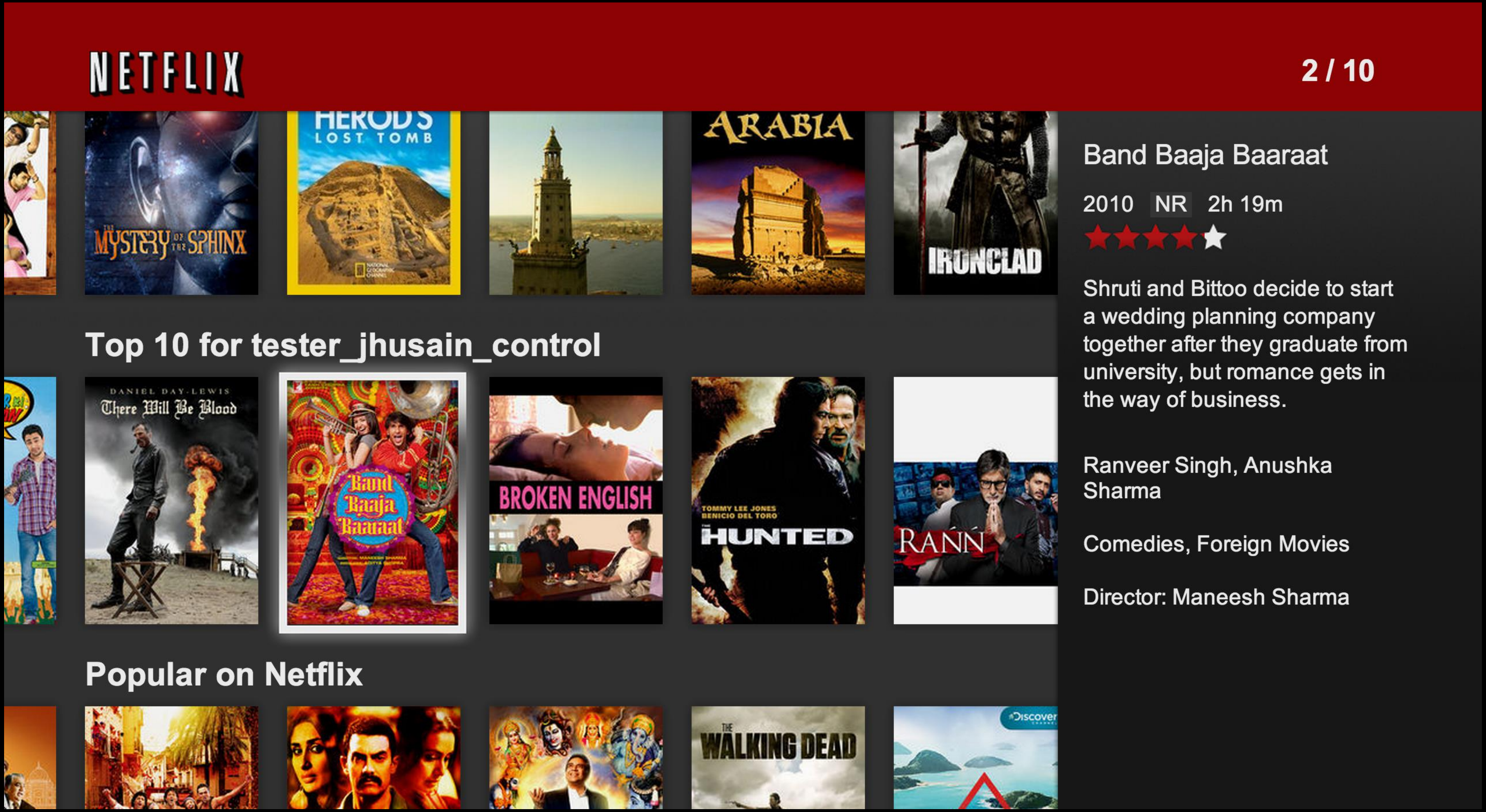
http://jort.technology/



```
/**
 * jortSort checks if your inputs are sorted until another Observable sequence fires.
 * See http://jort.technology/ for full details.
 * @returns {Observable} An observable which has a single value of true if sorted, else false.
 */
observableProto.jortSortUntil = function (other) {
  var source = this;
  return new AnonymousObservable(function (observer) {
    var arr = [];
    return source.takeUntil(other).subscribe(
      arr.push.bind(arr),
      observer.onError.bind(observer),
      function () {
        var sorted = arr.slice(0).sort(defaultSubComparer);
        observer.onNext(isEqual(arr, sorted));
        observer.onCompleted();
      });
  });
};
```

Your Netflix Video Lists

Netflix Row Update Polling



Band Baaja Baaraat

2010 NR 2h 19m

★★★★☆

Shruti and Bittoo decide to start a wedding planning company together after they graduate from university, but romance gets in the way of business.

Ranveer Singh, Anushka Sharma

Comedies, Foreign Movies

Director: Maneesh Sharma

Client: Polling for Row Updates

```
function getRowUpdates(row) {
    var scrolls = Rx.Observable.fromEvent(document, "scroll");
    var rowVisibilities =
        scrolls.throttle(50)
             .map(function (scrollEvent) { return row.isVisible(scrollEvent.offset); })
             .distinctUntilChanged()
             .publish().refCount();
    var rowShows = rowVisibilities.filter(function (v) { return v; });
    var rowHides = rowVisibilities.filter(function (v) { return !v) });

    return rowShows
       .flatMap(Rx.Observable.interval(10))
       .flatMap(function () { return row.getRowData().takeUntil(rowHides); })
       .toArray();
};
```