

# Intro to RxJS

Or

How I Learned To Stop Worrying and Love RxJS

Luka Skukan  
JS Developer @ Oradian

# Returning Data in JavaScript

|              | Single    | Multiple           |
|--------------|-----------|--------------------|
| Synchronous  | Primitive | Array, Object, ... |
| Asynchronous | Promise   | ???                |

# Rx = Query Operators + Observables

- **Reactive Extensions** - Libraries for async/event data flows using *Observable* and LINQ-like *operators* in various languages
- **Pushed** data (as compared to **pulling** iterators)
- Dataflow-based with array-like transformations
- Handles data, errors, and completion

# Returning Data in JavaScript II

|              | Single    | Multiple           |
|--------------|-----------|--------------------|
| Synchronous  | Primitive | Array, Object, ... |
| Asynchronous | Promise   | Observable         |

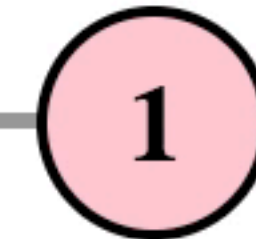
# Basics of RxJS

- Observables emit streams of data asynchronously, we transform them functionally
- Anything can be observed - HTTP calls, DOM event sources, basic promises, generators, even arrays or primitives
- Subscribers use the final result of transformations to do something
- Streams can be combined in various ways

# Basic Example

```
integersSource$  
  .filter((i) => i % 2 === 0) // even  
  .map((i) => i * 2) // double  
  .take(6) // first 6  
  .subscribe((i) => console.log(i));
```

# Marbles



# Hot and Cold

- Observables come in two flavours: **hot** and **cold**
- Hot run even if we do not subscribe - DOM events
- Cold run only if, and every time, we subscribe - HTTP
- Cold to hot, refCount



“A demo is worth a thousand words.”

*-Me*

<https://github.com/the-overengineer/rxjs-talks-demos>

# RxJS and the Ecosystem

- RxJS and Angular
- RxJS and React
- Cycle.js

# Cons of RxJS

- Adds complexity (119+ operators), special (F)RP patterns for solving certain problems
- Added Size - 428 kB, 141 kB minified (fixable, though)
- Hard to debug pre-v5
- Glue layer can be difficult

# Should I use RxJS?

- If you have few async interactions: **Probably Not**
- If you have simple async interactions: **Probably Not**
- Stack already has a nice way of handling async interactions: **Probably Not**
- Otherwise: **Knock Yourself Out**

# What Next?

- Very basic intro into RxJS - It's a large ecosystem
- RTFM
- Be careful what version of RxJS is being used (great outdated resources out there)
- Be smart - Don't use where not needed
- Other RP implementations - bacon.js, kefir

# Links & References

- <https://github.com/ichpuchtli/awesome-rxjs>
- <https://github.com/Reactive-Extensions/RxJS/blob/master/doc/gettingstarted/which-instance.md> (note: v4)
- <http://rxmarbles.com>
- <https://xgrommx.github.io/rx-book/index.html> (note: v4)
- [v4 - v5 migration guide](#)
- <https://gist.github.com/staltz/868e7e9bc2a7b8c1f754>

# Thank You!

Luka Skukan

JS Developer @ Oradian



*Still don't use twitter*



the-overengineer