# Getting Started with Reactive Programming Using RxJS

## OBSERVERS AND OBSERVABLES

**Scott Allen**

OdeToCode.com - @OdeToCode

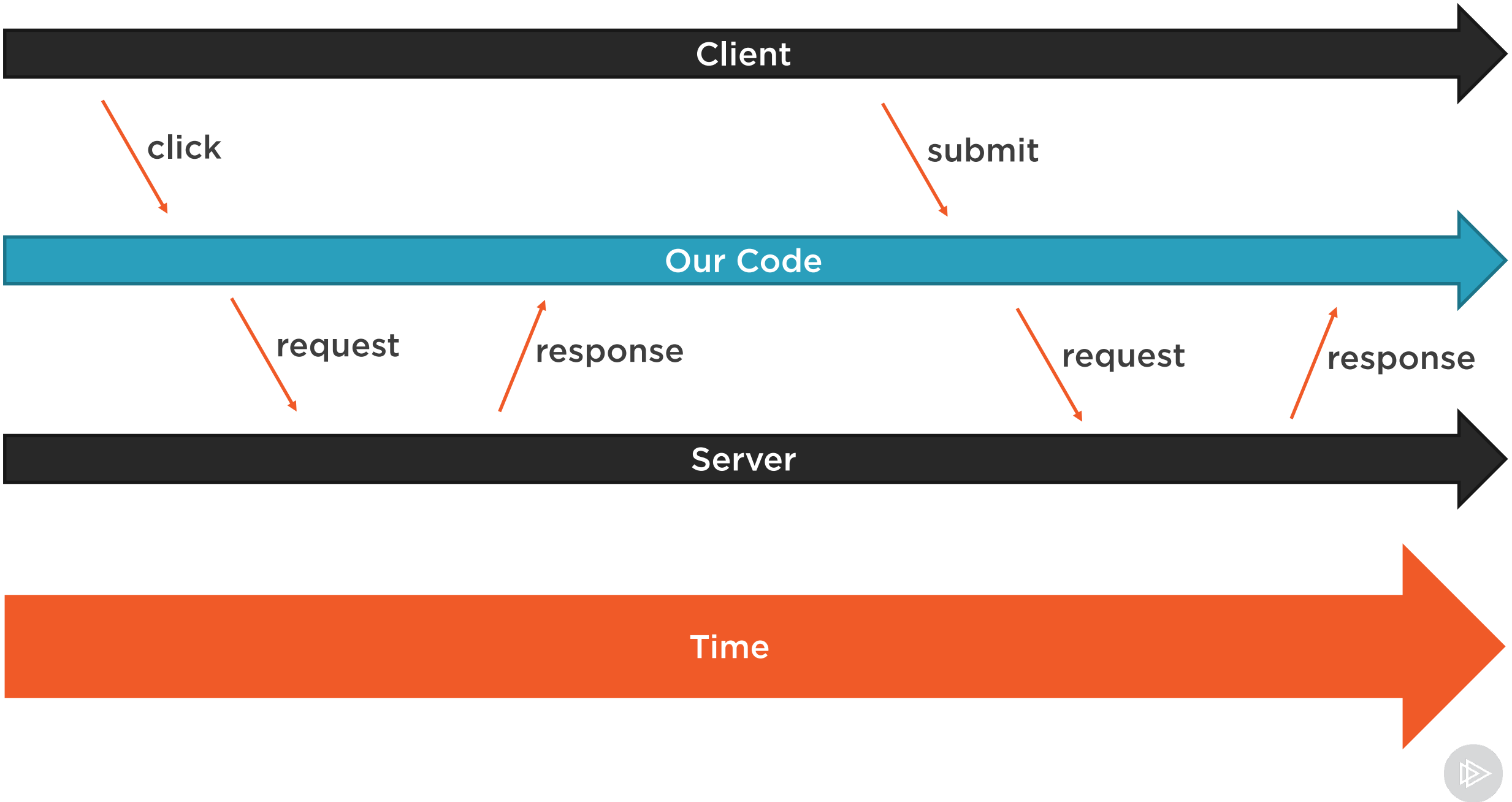

K. Scott
ReactiveX
reactivex.io
ReactiveX
Introduction
Docs
Languages
Resources
Community
ReactiveX
An API for asynchronous programming
with observable streams
Choose your platform

# Languages

- Java: RxJava
- JavaScript: RxJS
- C#: Rx.NET
- C#(Unity): UniRx
- Scala: RxScala
- Clojure: RxClojure
- C++: RxCpp
- Ruby: Rx.rb
- Python: RxPY
- Groovy: RxGroovy
- JRuby: RxJRuby
- Kotlin: RxKotlin
- Swift: RxSwift
- PHP: RxPHP

# ReactiveX for platforms and frameworks

- RxNetty


Client
click
submit
Our Code
request
response
request
response
Server
Time


```
var result = movies.filter(function(movie) {
    return movie.title[0] === "S";
});
```


```
var result = movies.filter(function(movie) {
    return movie.title[0] === "S";
});
```

{ title: “Star Wars” } { title: “Star Trek” } { title: “Shrek” }

{ title: “E.T.” } { title: “Casablanca” } { title: “Vertigo” }

```
var result = movies.filter(function(movie) {
    return movie.title[0] === "S";
});
```

{ title: “Jaws” } { title: “2001” } { title: “Terminator” }

# Summary

```
let source = Observable.create(observer => {

    let index = 0;
    let produceValue = () => {
        observer.next(numbers[index++]);

        if(index < numbers.length) {
            setTimeout(produceValue, 250);
        }
        else {
            observer.complete();
        }
    }

    produceValue();

}).map(n => n * 2)
  .filter(n => n > 4);
```