RxJS: The Big Picture

-Rx stands for Reactive Extensions API aka Reactive X aka Rx

-API for asynchronous programming with observable streams (the data flows into your app)

-RxJS is the Javascript implementation of the Reactive Extension API

What problems does RxJS solve?

-provides a single API that facilitates and simplifies the processing of data from multiple/ sources

-Many different types of streams of data is processed a little differently in plain JS.

-RxJS helps you deal with these disparate data sources and the multiple ways they might be processed by giving you a single clean API you can use with all of them

-Named observable because the reactive extensions get their conceptual foundation from the observer software development pattern

A red and orange arrow pointing to a red object

Description automatically generated

-Observable pushes value to a single observer?

-Register an observer to receive the values from an observable by calling a method on the observable named subscribe

-The observer that should receive the value is passed as a parameter to the subscribe method

A diagram of a diagram

Description automatically generated with medium confidence

-The observable sends values to the observer by calling methods on the observer object itself

-Once the observer is subscribed to the observable, the first value can be pushed to the observer by calling next

-Operators are functions that manipulate the data produced by an observable and return a new observable

Library

-supplemental pieces of functionality

-solve specific problems but doesn’t dictate the overall architecture of your app

Frameworks

-much larger and prescriptive



-$ is a convention when using RxJS to store observables

-from will create an observable

Subjects: let you produce values for multiple observers

Schedulers: gives fined grained control over when your observables are executed

A diagram of an object

Description automatically generated

Subjects

-similar to observables

-can push to multiple observers (multicast)

-Observables can only produces values to a single observer (unicast)

A close up of a sign

Description automatically generatedA close up of a text

Description automatically generated

-Observables can be configured with schedulers to control the execution context for the observable

-Only have to worry about schedulers if you want to do something very precise

A screen shot of a computer program

Description automatically generatedA computer screen with text and numbers

Description automatically generated

-Observables will never call error and complete

RxJs in Angular: Reactive Development

-Declaring data streams, combining streams, reacting to actions

A black text on a white background

Description automatically generated

A close up of a text

Description automatically generatedA screenshot of a phone

Description automatically generatedA screenshot of a screen

Description automatically generated

A close up of a text

Description automatically generated

-Cache means retain it in a service to reuse the date without another HTTP request

A blue background with white text

Description automatically generated

-Angular service requests an HTTP GET request to a back-end server and client code continues and doesn’t have to wait

-Backend server receives the GET request and locates the data (DB) returns an response

-Callback is a function that can be called back after an async operation is complete, but difficult to manage when working with nested async operations

Can only handle a single emission and is not cancelable (promise and async/await)

-Promise is an object that may produce a single value sometime in the future.

-async/await special syntax that allows asynchronous code that looks synchronous.

A screenshot of a computer program

Description automatically generated

A group of symbols with text

Description automatically generated with medium confidence

-We work with different data sources using the same technique and operators

-Lazy, evaluation does not start until subscription

Reactive Development

A diagram of mathematical equations

Description automatically generated

-Like in a shopping cart where we change our quantity to 3

Orange and grey text on a white background

Description automatically generatedWhat is Reactive Development

A screenshot of a cell phone

Description automatically generatedA white background with orange and grey text

Description automatically generated

RxJS Terms and Syntax

A screenshot of a computer

Description automatically generated

-Apples on the conveyor are observable and I am the observer

-An observer is an object that observes and responds to notifications specified as methods

-So the observable emits notifications and associated values, and it’s the observer that responds to those notifications and consumes those values

Subscriber

-implements the observer interface

-Inside RxJS, each observer is converted to a subscriber

-is rarely used except within RxJS, normally use observer

A screenshot of a computer code

Description automatically generated

-Handles three types of notification specified as methods

A diagram of a funnel

Description automatically generated

Observable

A white background with black text

Description automatically generated

-Think of it as a conveyor collecting and emitting items to an observer

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

-Observables can by synchronous, meaning the items are emitted immediately and in sequence

Hot: Observables starts emitting items even with no subscriber

Cold: Observables waits for a subscriber

A screenshot of a computer code

Description automatically generated

-Observer think of set of callbacks

-Observable think of collections of items emitted over time

Shortcut

A computer code with text

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

-Unsubscribing does not call the observer’s complete()

-Properly unsubscribing from each Observable helps avoid memory leaks

Creating an Observable



-From is useful for creating an observable from any data

-… is called Spread operator and it spreads the element of the array

A graph with an arrow pointing to the right

Description automatically generated with medium confidence-of/from automatically emit a complete notification and unsubscribe

A screenshot of a computer code

Description automatically generated

-fromEvent creates an observable from any document, object, model or DOM

-interval creates an observable that emits a sequential number at a defined interval (1000ms)

A close-up of a computer code

Description automatically generated

A computer code with text

Description automatically generated with medium confidenceA white background with green text

Description automatically generatedA white background with green text

Description automatically generated

Operators

-is a function

-used to transform and manipulate emitted items

-apply it in sequence using Observab’s pipe method

A close up of a text

Description automatically generated

-A function performs a side effect if it changes any state that does not affect its return value

Ex. using tap to increment a counter or set a boolean

A screenshot of a white text

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

-Main point is operator takes in an input observable, subscribe to it, creates an output observable and emits items and notification to that output observable

-many of us learned procedural patterns for retrieving data, now we will learn Reactive

A white background with black text

Description automatically generatedA blue and black line drawing of a pipe

Description automatically generated

RxJS features: catchError, EMPTY, throwError

Async PipeA screenshot of a video

Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer screen shot of a program

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer error

Description automatically generated

-Can use JS throw statement instead of throwError()

A screenshot of a computer

Description automatically generated

-EMPTY is a constant we can use to replace an errored observable with a valid but empty observable

-emits no items and immediately completes

A screen shot of a computer code

Description automatically generatedError Handling Strategies

A white background with black text

Description automatically generated



A computer screen shot of a program

Description automatically generated

-Bottom part is how we do an Reactive approach since we don’t have subscribe method anymore

-We don’t want to rethrow from our component because the error will propagate to our template, so return an EMPTY observable

A diagram of a triangle and a green arrow

Description automatically generated with medium confidence

A screen shot of a computer

Description automatically generated

-Angular uses change detection to track changes to app data so that it knows when to update the UI

A screen shot of a computer program

Description automatically generated

-We have a procedure that calls from our component

A screen shot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

- we declare an observable property and assign it directly to HTTP GET

-Called declarative because each observable are a declared property instead of part of a method

-We work with observables directly instead of calling methods that return observables

A screenshot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

-HHTP will return an observable that has the shape of < >

-We define properties for the desired new product shape and set the value of each product using values from the original retrieved product as needed

-To ensure we keep things strongly typed, we add as Product to type the object literal

A screen shot of a computer

Description automatically generated

-Use the spread operator as a shortcut

Note: we want an object literal so we add ( ) around our { }

-Must wrap map((products: Product[]) =>

Combining Data Streams

A diagram with multiple colored circles and numbers

Description automatically generatedA diagram of a combination of operators

Description automatically generated

A close-up of a text

Description automatically generated

-Does not emit anything until all of the observable have emitted at least one value

A screenshot of a computer

Description automatically generated

A white background with black text

Description automatically generatedA screenshot of a computer

Description automatically generated

A diagram of a stream

Description automatically generated

withLatestFrom: if the source observable completes before the other input observables, the result observable will complete without emitting anything

A computer screen shot of a code

Description automatically generatedA screenshot of a computer code

Description automatically generated

-combineLatest was deprecated

-.pipe( map( (val, index) =>

Reaction to Actions

A screenshot of a computer code

Description automatically generated

ngOnInit(): runs each time the page is initialized

when we subscribe to the observable, then the service issues HTTP request

http get returns an observable?

Note: we need the return keyword since we now have a multiline arrow function