Reactive Programming Concepts

sambath.narayanan@dataeverconsulting.com

What is Async Programming ?

- Async programming is a technique that enables your program to start a potentially long running task and still be able to be responsive to other events while that task runs, rather than having to wait until that has finished
    - Example : making HTTP requests using fetch()
- Async programming is non-blocking

Asynchronous Programming

- To display useful data to your user on their screen, you first need to have that data in your application.

- If you need data that your application doesn’t have hard-coded into it, then you’ll have to get that data from somewhere, usually a server on the internet somewhere.

- To get that data you’re going to have to make an HttpRequest to that server asking for it.

Asynchronous Programming

- It’s impossible to know exactly how long it will take before that server will deliver your data to your application.

- Therefore it took an indefinite amount of time to accomplish that piece of logic giving you the data you asked for.

- The response didn’t come synchronously — or line after line of code — instead it came whenever it came.

- Usually, you’ve written some code prepared to do something with that data upon receipt. That’s why it’s called asynchronous.

Callback

- Callback is function that is passed into another function, with the expectation that callback will be called at the appropriate time.
- Callback is the primary method asynchronous functions were implemented in JS
- Callbacks allow you to specify what should happen after an asynchronous operation finishes without blocking the main program execution.

```
function doSomethingAsync(callback) {
  setTimeout(function () {
    console.log("Task done!");
    callback(); // Call the callback function
when the task is done
  }, 1000);
}

function afterTask() {
  console.log("Callback function called.");
}

doSomethingAsync(afterTask);
```

How to use promises?

- Promises are foundations of Async programming in JS.
- A promise is an object returned by an asynchronous function which represents the current state of the operation
- Promise object provides methods to handle eventual su cess or failure of the operation

Asynchronous JavaScript and XML(AJAX)

1. AJAX, is a set of web development techniques that allows you to send and receive data from a web server asynchronously without needing to refresh the entire web page. It enables you to update specific parts of a web page dynamically, making web applications more responsive

2. You simply provide the URL you want to fetch data from.
3. It returns a Promise, allowing you to chain **.then()** to handle the response asynchronously.
4. The first **.then()** checks if the response was successful (status code 200) and parses the response data (assuming it's in JSON format).
5. The second **.then()** handles the processed data, and the **.catch()** method handles any errors that may occur during the fetch operation.

```
fetch("https://api.example.com/data")
  .then(function (response) {
    if (!response.ok) {
      throw new Error("Network response was
not ok");
    }
    return response.json(); // Assuming the
response is in JSON format
  })
  .then(function (data) {
    // Process the JSON data here
  })
  .catch(function (error) {
    console.error("Fetch error:", error);
  });
```

What is Sync Programming ?

- Declares a string called `name`
- Declares another string called `greeting`, which uses `name`
- Outputs the greeting to the JS console

Browser effectively steps thru the program one line at a time. This makes the sync program

```
const name = “Mira”;

Const greeting = ‘Hello, my name
is $(name)’;

Console.log(greeting);

//”Hello my name is Mira!”
```

What is a Reactive Programming and RxJS ?

- RP is an asynchronous programming paradigm concerned with Data streams and the propagation of change
- RxJS is a library for RP using observables that makes it easier to compose asynchronous or call-back based method

What is a Reactive Programming and RxJS ?

- Simplifying further …
  - RP is altogether different way of developing application software
  - App software is created to react to changes that happen events that happen(click, data loading, http call) instead of traditional way of developing code where we explicitly write the code to handle those changes

What is RxJS ?

What is RxJS?

- RxJS is a library for composing asynchronous and event based programs by using observable sequences
- It provides one core type observable satellite types operators to allow handling asynchronous events as collections.
    - Satellite data type – observer, subject (specifying data type)
    - Operator – map, max, filter, reduce
    - All this help in data streams

RxJS concepts

Observable

An observable is a function that creates an observer and attaches it to the source where values are expected, for example, clicks, mouse events from a dom* element or an Http request, etc.

In simple terms - Source of data or events that can be observed by one or more observers

Observer

It is an object with next(), error() and complete() methods, that will get called when there is interaction to the with the observable i.e. the source interacts for an example button click, Http request, etc.

Subscription

When the observable is created, to execute the observable we need to subscribe to it. It can also be used to cancel the execution.

RxJS more concepts

Operators

An operator is a pure function that takes in observable as input and the output is also an observable.

Subject

A subject is an observable that can multicast i.e. talk to many observers. Consider a button with an event listener, the function attached to the event using addlistener is called every time the user clicks on the button similar functionality goes for subject too.

Schedulers

A scheduler controls the execution of when the subscription has to start and notified.

What is Observable ?

What is an observable ?

- Observables is a new push-system for JS
- An observer is a producer of multiple values and pushing them to observers

What is a Data stream?

- Events happening one after another over a time
- example : group of people chatting, shopping cart application of an e-commerce site

Observable is a Data stream?

I. Consider an e-commerce shopping-cart website

II. Customer visits the site and places an order

III. Order and shipping status will change over time /continuously changing (one variable takes different values over period of time)

1) Order status = Null (before order is placed)
2) Order status = Order placed
3) Order status = Order received
4) Order status = Order getting processed
5) Order submitted

IV. This needs to be informed/updated to multiple applications

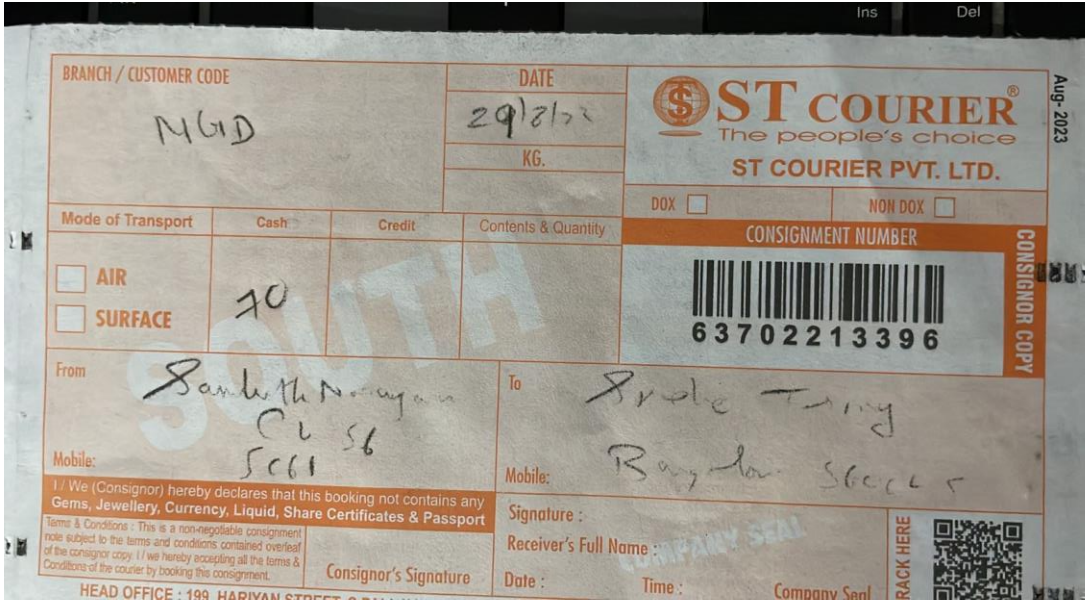
V. Customers,others can check updates which changes over time

Subscriber?

I. Subscriber listens to the observable for data changes or updates

II. App is the dashboard where status will be updated

III. Order status always latest since we have subscribed to the observable



BRANCH / CUSTOMER CODE MGD	DATE 29/8/23 KG.	ST COURIER® The people's choice ST COURIER PVT. LTD.
		DOX □ NON DOX □

Aug - 2023

Mode of Transport	Cash	Credit	Contents & Quantity
□ AIR □ SURFACE	70		

CONSIGNMENT NUMBER

6 3 7 0 2 2 1 3 3 9 6

CONSIGNOR COPY

From Sankith Narayan
CL 56
5661
Mobile:

To Srede Tarng
Bangalore 56cc25
Mobile:

I / We (Consignor) hereby declares that this booking not contains any **Gems, Jewellery, Currency, Liquid, Share Certificates & Passport**

Terms & Conditions : This is a non-negotiable consignment note subject to the terms and conditions contained overleaf of the consignor copy. I / we hereby accepting all the terms & Conditions of the courier by booking this consignment.

Consignor's Signature

Signature :

Receiver's Full Name :

Date : Time : Company Seal

COMPANY SEAL

TRACK HERE

HEAD OFFICE : 199, HARIYAN STREET

Observable is a Data stream? Example 2

I. Consider a courier company

II. Customer sends a packet from destination A to destination B

III. Courier shipping status will change over time /continuously changing (one variable takes different values over period of time)

   1) Order status = Null (before order is placed)

   2) Order status = Packet picked

   3) Order status = packet has reached source hub

   4) Order status = Packet is in transit

   5) Order status = packer reach destination hub

   6) Order status = packet out for delivery

IV. Customer and others can check updates which changes over time

Observable is a Data stream? Example 3

- Can you think of any other example ?

Subscriber

- You subscribe to a Youtube channel
- You get and update whenever there is a new video is posted

RxJs Operator

- Filter – look only at videos but not community messages
- Map – map to all users

Everything is a stream

Reactive programming

Functional programming

Reacting to user actions

  - Mouse click, sensor out put, keyboard input

Angular has RxJS as library(good combination)

Event driven programming

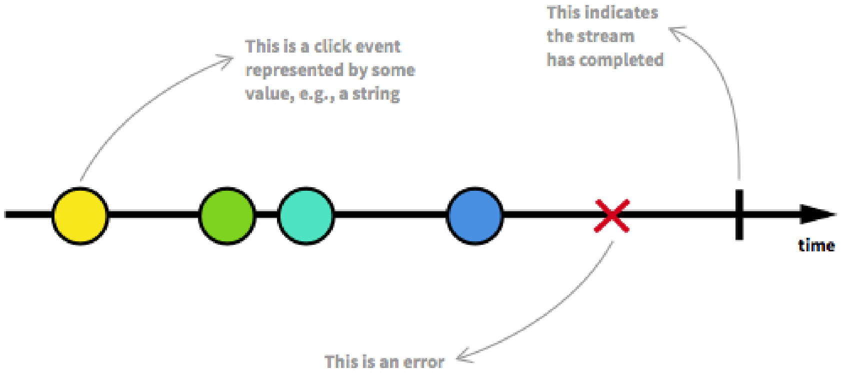
Event Driven Programming

You build your program so that it reacts to user actions — such as mouse clicks, key presses, and voice-activation — sensor outputs, or messages containing data from other programs — including programs running on other servers.

These are called **events**

Data Stream

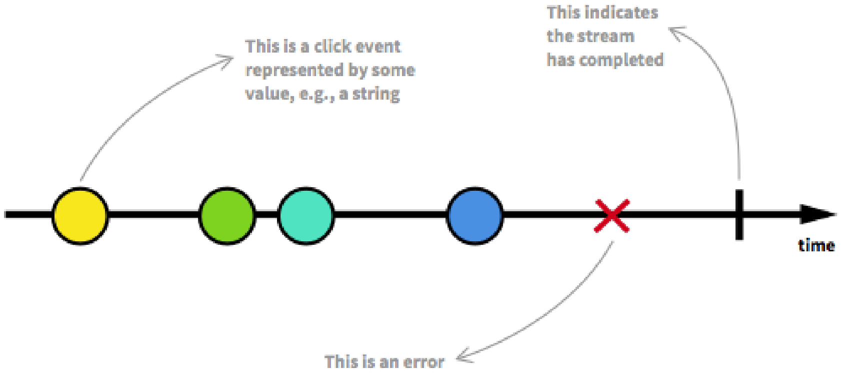
- A data stream is a sequence of ongoing events ordered in time.
- It can emit three different things:
- a value (of some type),
- an error, or
- a "completed" signal.
- We capture these emitted events only **asynchronously, by defining a function that will execute when a value is emitted.**


This is a click event represented by some value, e.g., a string
This indicates the stream has completed
time
This is an error

Subscribing and Observers

- The "listening" to the stream is called **subscribing.**

- The functions we are defining are **observers.**



This is a click event represented by some value, e.g., a string
This indicates the stream has completed
time
This is an error

The other illustration

```
--a---b-c---d---X---|->

a, b, c, d are emitted values
X is an error
| is the 'completed' signal
 ---> is the timeline
```

Understand and build streams

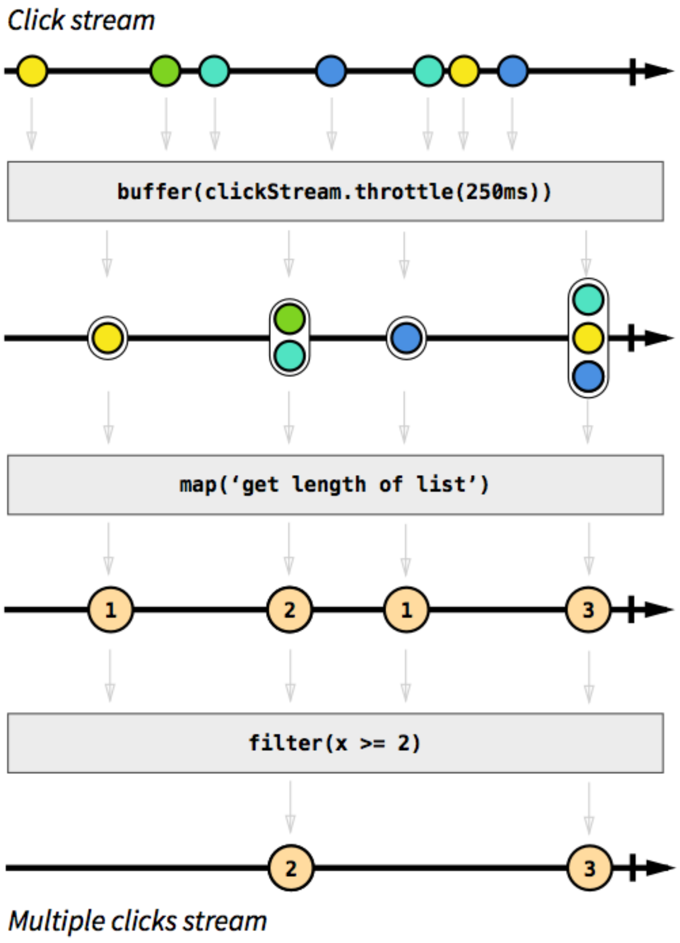
Grey boxes are functions transforming one stream into another.

First we accumulate clicks in lists, whenever 250 milliseconds of "event silence" has happened (that's what buffer(stream.throttle(250ms)) does, in a nutshell.

The result is a stream of lists, from which we apply map() to map each list to an integer matching the length of that list.

Finally, we ignore 1 integers using the filter(x >= 2) function. That's it: 3 operations to produce our intended stream.

We can then subscribe ("listen") to it to react


Click stream
buffer(clickStream.throttle(250ms))
map('get length of list')
1
2
1
3
filter(x >= 2)
2
3
Multiple clicks stream

RxJS

(Reactive eXtensions for Java Script)

Concepts

sambath.narayanan@dataeverconsulting.com

nodejs and npm installation

- You need to have nodejs and npm installed on your system.

- To verify if NodeJS and npm is installed on your system, try to execute the following command in your command prompt.

```
>node -v && npm -v
v10.15.1
6.4.1
```

Node.js

Node.js asynchronous event-driven JavaScript runtime.

Thread-based networking is relatively inefficient and very difficult to use.

Users of Node.js are free from worries of dead-locking the process, since there are no locks.

Almost no function in Node.js directly performs I/O, so the process never blocks.

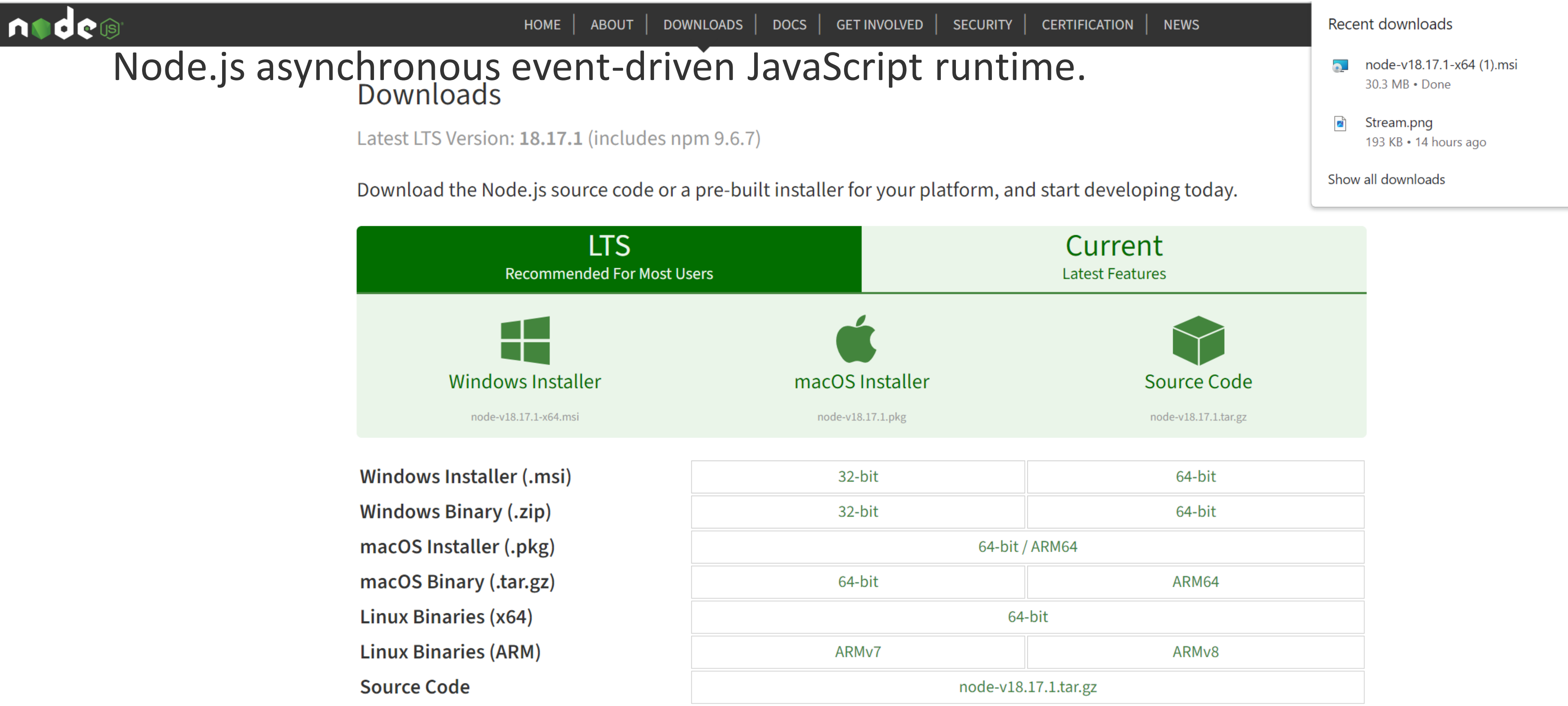
Because nothing blocks, scalable systems are very easy to develop in Node.js.

RxJS environment

- Node.JS
- NPM
- RxJS
- webpack

Node.js

Node.js asynchronous event-driven JavaScript runtime.



HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | CERTIFICATION | NEWS

Recent downloads

node-v18.17.1-x64 (1).msi
30.3 MB • Done

Stream.png
193 KB • 14 hours ago

Show all downloads

Downloads

Latest LTS Version: **18.17.1** (includes npm 9.6.7)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit / ARM64	
macOS Binary (.tar.gz)	64-bit	ARM64
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v18.17.1.tar.gz	

NPM

NPM is software registry

Consists of

- website
- CLI
- registry

What is babel and how to install it?

Babel is a JS compiler that is mainly used to convert code into a backwards compatible version of JavaScript in current and older browsers or environments

```
npm i -D @babel/preset-react @babel/preset-env
@babel/core babel-  loader @babel/plugin-proposal-
class-properties
```

Using Webpack, Webpack Dev Server and Babel from scratch

On dhruv, login as sambath

```
cd /shared/TCE…CLOUD/REACTIVE-PROGRAMMING/react-from-scratch/projectname
```

```
    npm run dev
```

In a browser, type

```
    http://localhost:8080/
```

RxJS is used in combination with **backend** technologies

node.js

RxJS can be used in backend development with Node.js. RxJS can be used to manage asynchronous operations, handle streams of data from sources like databases or APIs, and build reactive server-side

Webpack

- Webpack takes modules with dependencies and generates static assets representing those modules.[10]
- Webpack takes the dependencies and generates a dependency graph allowing web developers to use a modular approach for their web application development purposes.
- Webpack is a module bundler for JavaScript.

Webpack concepts

- module Entry:
  - the entry point is the module that webpack uses to start building its internal dependency graph. From there, it determines which other modules and libraries that entry point depends on (directly and indirectly) and includes them in the graph until no dependency is left. By default, the entry property is set to ./src/index.js
- Output
  - the output property instructs webpack where to emit the bundle and what name to use for the files. The default value for this property is `./dist/main.js`
- Loaders:
  - To process other types of files and convert them into valid modules, webpack uses loaders. Loaders transform the source code of non-JavaScript modules.

Webpack concepts

- Mode:
  - we work with two types of source code — one for the development build and one for the production build. Webpack allows us to set which one we want to be produced by changing the mode parameter to **development**, **production** or **none**.

Observable

What is observable?Continuous data stream is coming. This is observable. Observable emit data over a period of time

Observable is a stream

Observables emit

RxJS is a core piece of Data stream

Ecomm shopping cart application, user places order

Before placing order - status is nothing

Order will change over time

This Order status needs to informed to people

The data is continuously changing

Promises vs Observable

Promise vs observables

No ways to cancel promises

Promises

Subscribable

What is Subscriber?

Observables don’t do anything unless they are subscribed to

Usecase

You want to take out of the observable

You want to react to the event

Subscribable

What is Subscriber?

Observables don’t do anything unless they are subscribed to

Usecase

You want to take out of the observable

You want to react to the event

Observer

Observer has three methods

- next
- complete
- Error

Angular uses observables extensiv for

- Routing
- Forms

Additional material

https://www.sitepoint.com/webpack-beginner-guide/

https://www.javatpoint.com/rxjs-first-example

https://www.youtube.com/watch?v=ztspvPYybIY

https://www.sitepoint.com/webpack-beginner-guide/

https://nodejs.org/en/download/

https://nodejs.org/en/docs/guides/blocking-vs-non-blocking

https://www.tutorialspoint.com/rxjs/rxjs_quick_guide.htm

https://www.youtube.com/watch?v=uQGl9x9VHgA

Reactive Programming Concepts

sambath.narayanan@dataeverconsulting.com

Observer

Angular uses observables extensiv for

When creating a new Angular application, you have the option to choose between different stylesheets formats. The two most common stylesheet formats used in Angular projects are:

Use the CSS template

Reactive programming class for CS students

Hours = 2 , Saturday,02.09.2023, (10.00 am to 12.00 pm)

https://www.youtube.com/watch?v=PhggNGsSQyg&t=342s

The following is a good conceptual video

Summary : https://www.youtube.com/watch?v=h7CozVrhFoQ&list=PLp50dWW_m40WID0zCrftuRCVa3mCaxYG6&index=1&t=337s

RxJS is used in combination with **frontend** technologies

RxJS is widely used in combination with other frontend technologies to manage asynchronous operations, handle data streams, and create responsive user interfaces.

Angular

Angular is a frontend framework for building dynamic web applications.

It has built-in support for RxJS.

It is easy to integrate observables and leverage reactive programming in various parts of application.

Angular uses RxJS for handling data flows, event handling, and managing asynchronous operations.