**RXJS - ANGULAR2**

[RXJS-Import](#_25hets5o0k3z)

[Observable -](#_flxiiwf5varv)

[Observables, observers, subscription](#_bjwedf52vbby)

[How to create observable](#_sxsteeidvi0b)

[MAP operator](#_lmed5ghfl7s6)

[Subject:](#_o2sww6fgu63a)

[trigger emission of new value manually, we want to use observable like event emitter,](#_fo1cqnlkuf2g)

[Filter](#_8xko7b8czt3l)

[REDUCE](#_n9sqpsbbllnu)

[SCAN](#_8z0dvutpu7lp)

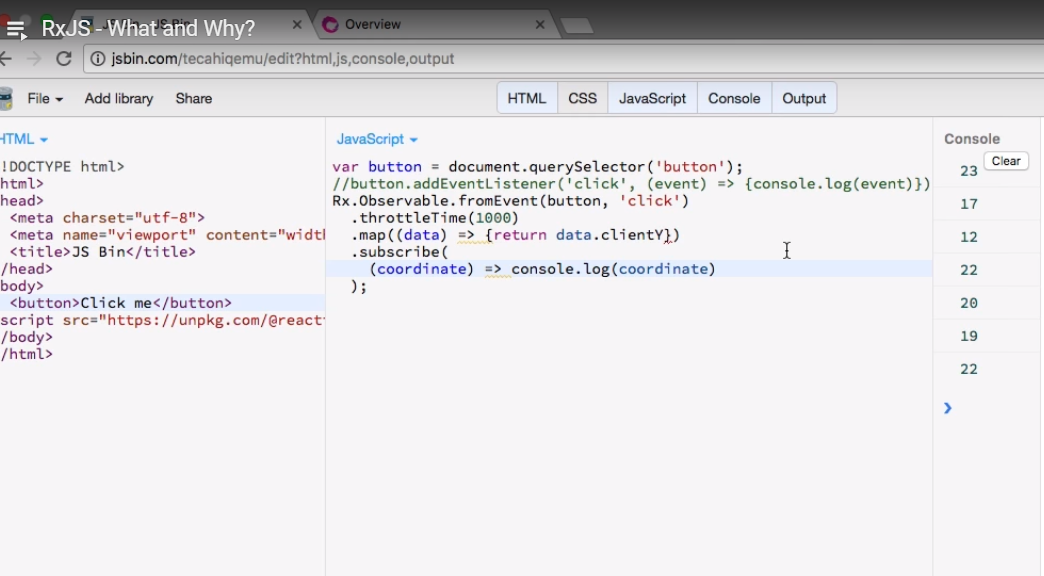
[Behaviour Subject](#_4ssmgzvuf0iq)

[Debounce time](#_jznw1cqeypci)

[Merge Map](#_r5nd5nylca16)

[Switch Map](#_4aeeurqdrz4n)

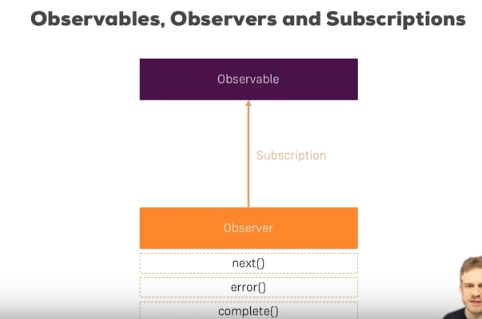
##### RXJS-Import

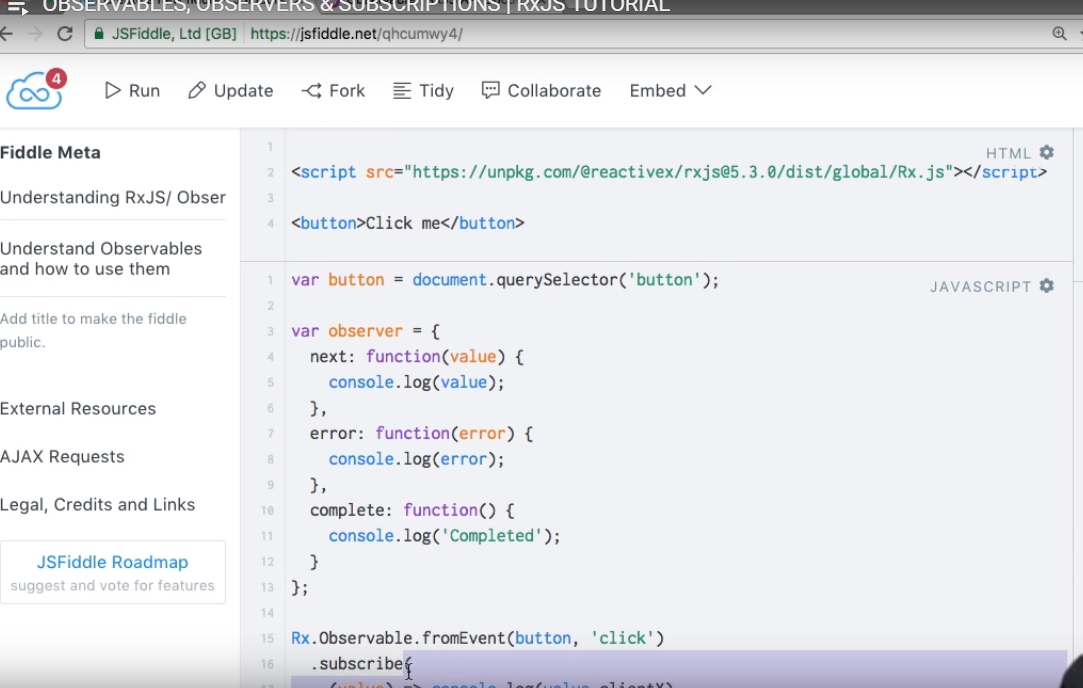


##### Observable -

Wrapper around stream of data source (stream of values), observer runs a code whenever the stream receives next value, error or completed value and we connect observer to observable thru subscription

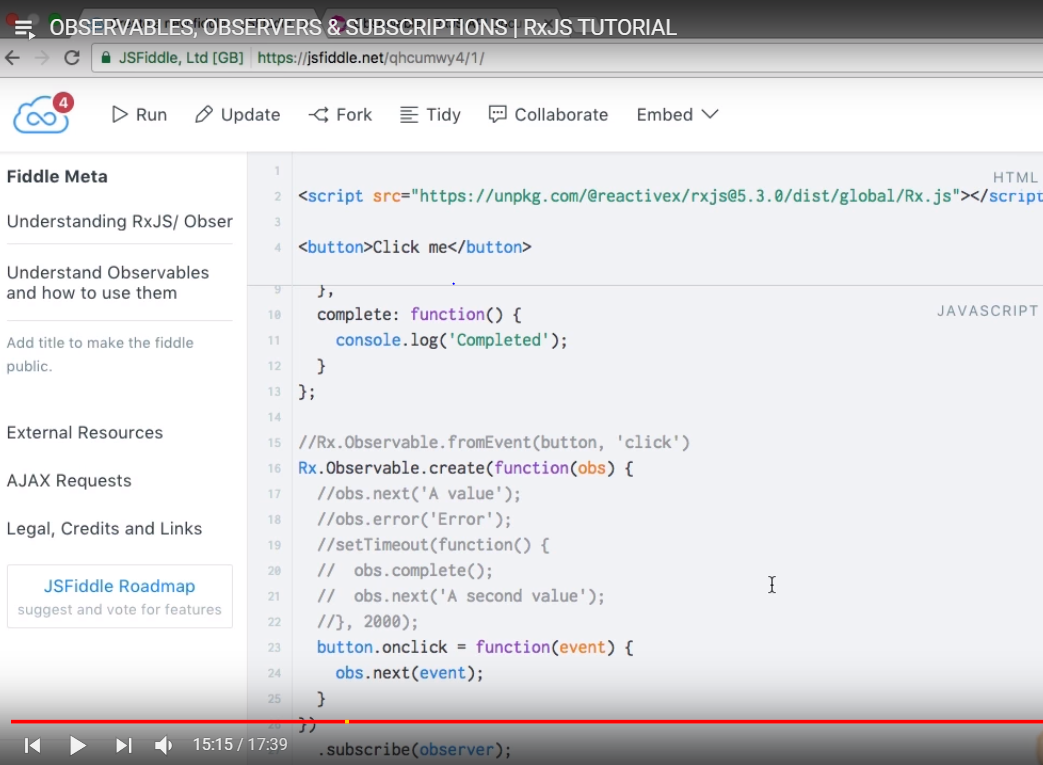
##### Observables, observers, subscription





##### How to create observable





##### MAP operator



##### **Subject:**

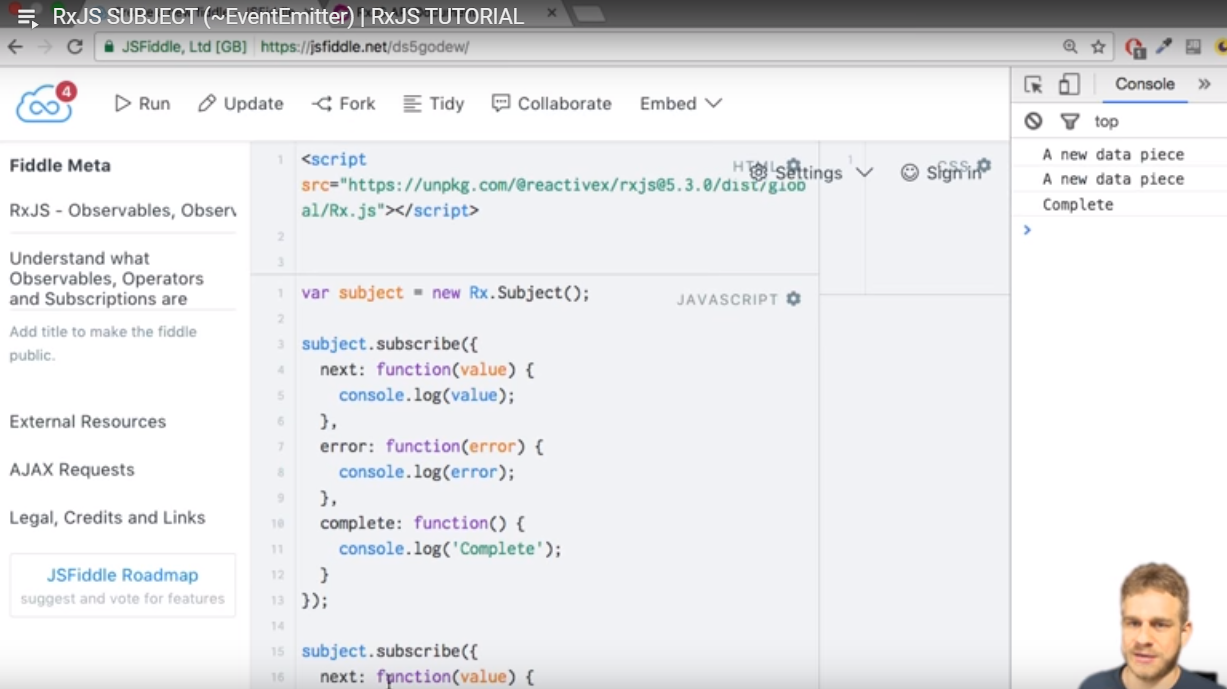
##### trigger emission of new value manually, we want to use observable like event emitter,

We control when new value is emitted.we use subject which inherit from observable but we can call

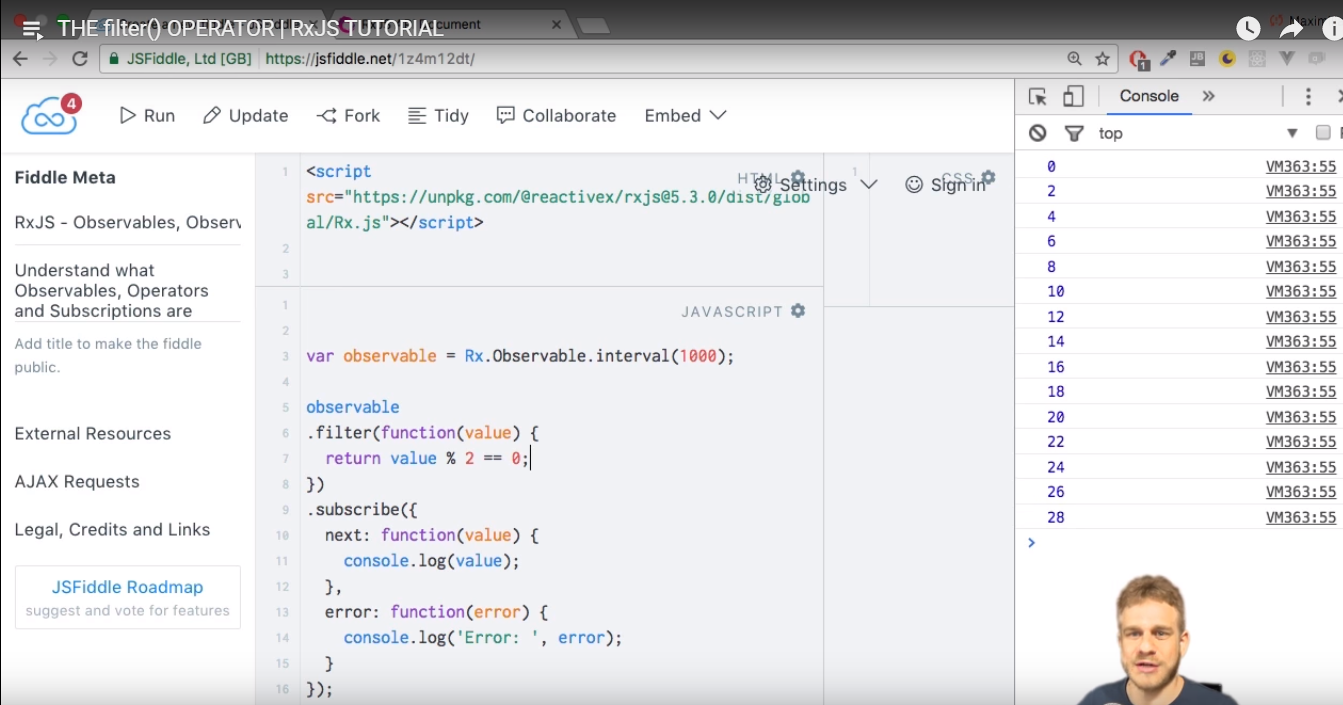
next () method manually to force it to emit new value, so therefore we can have more active approach

Of using observable with subject, used when we need to use like event emitter

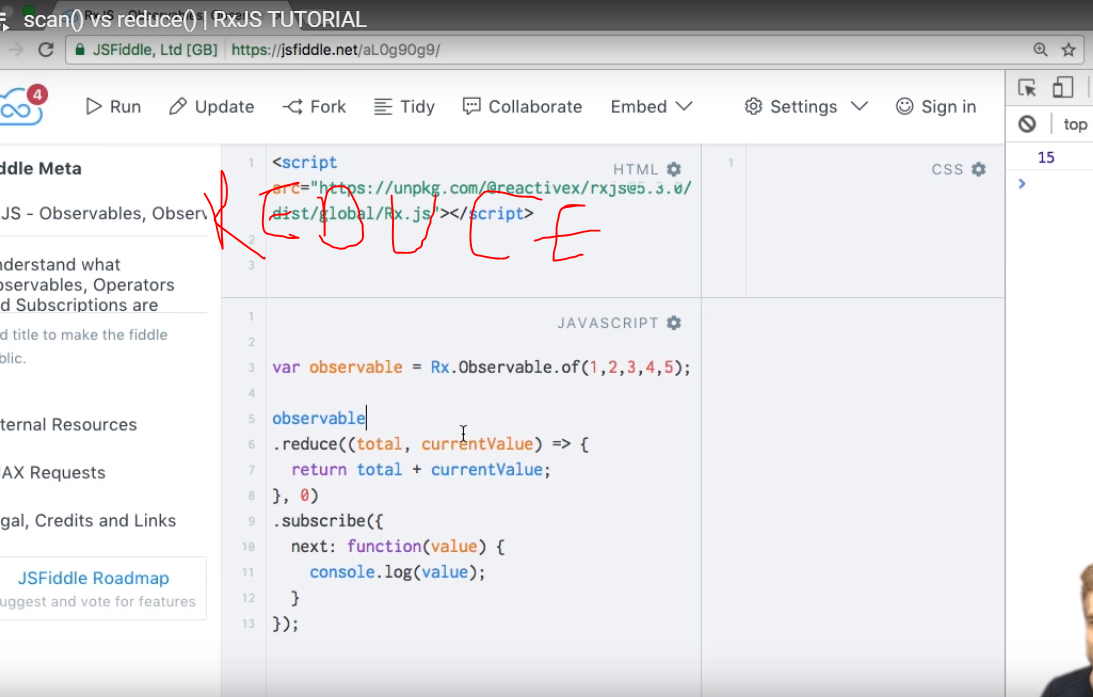
We decide when to emit value, error, completion with subject



##### Filter

****

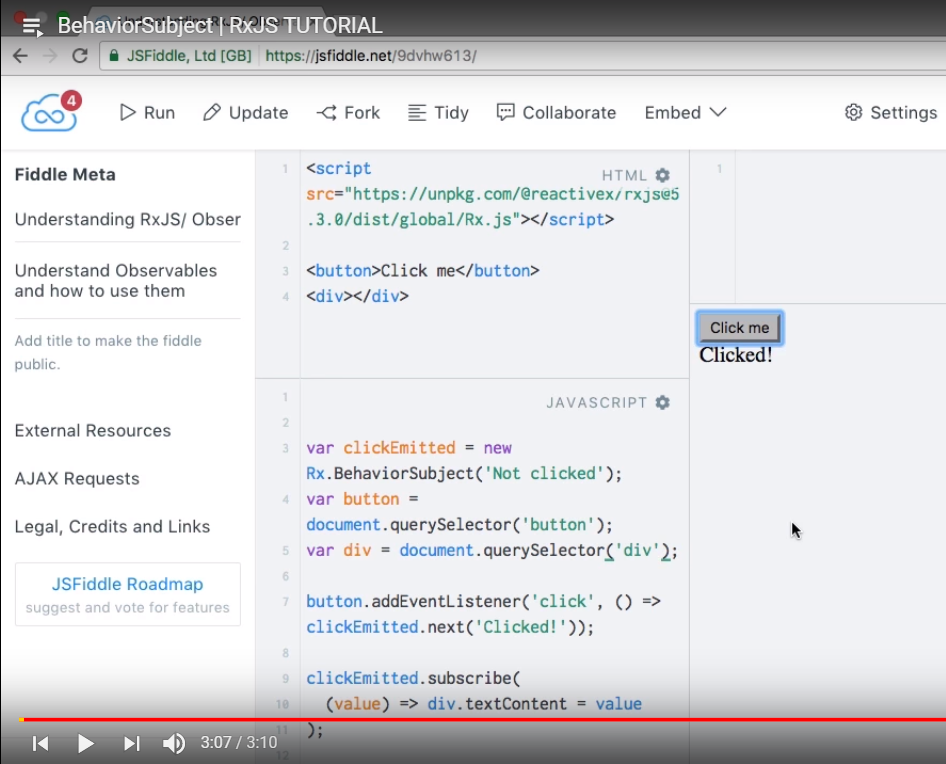
##### REDUCE

****

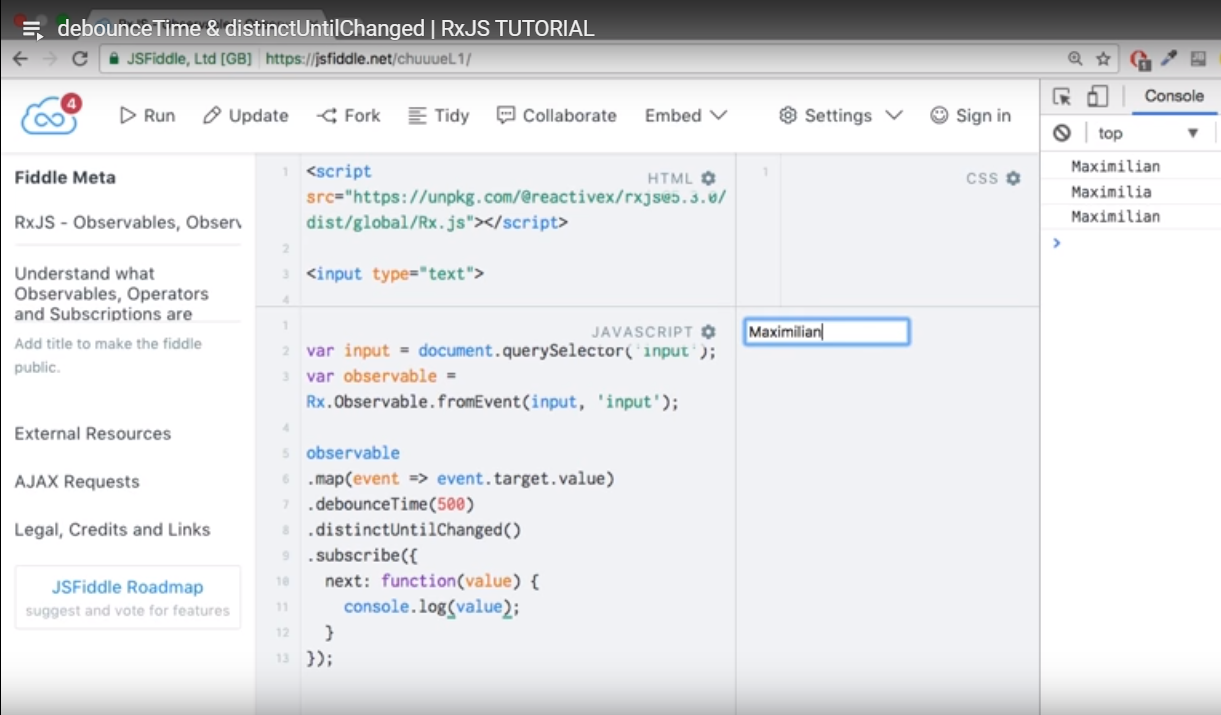
##### SCAN

****

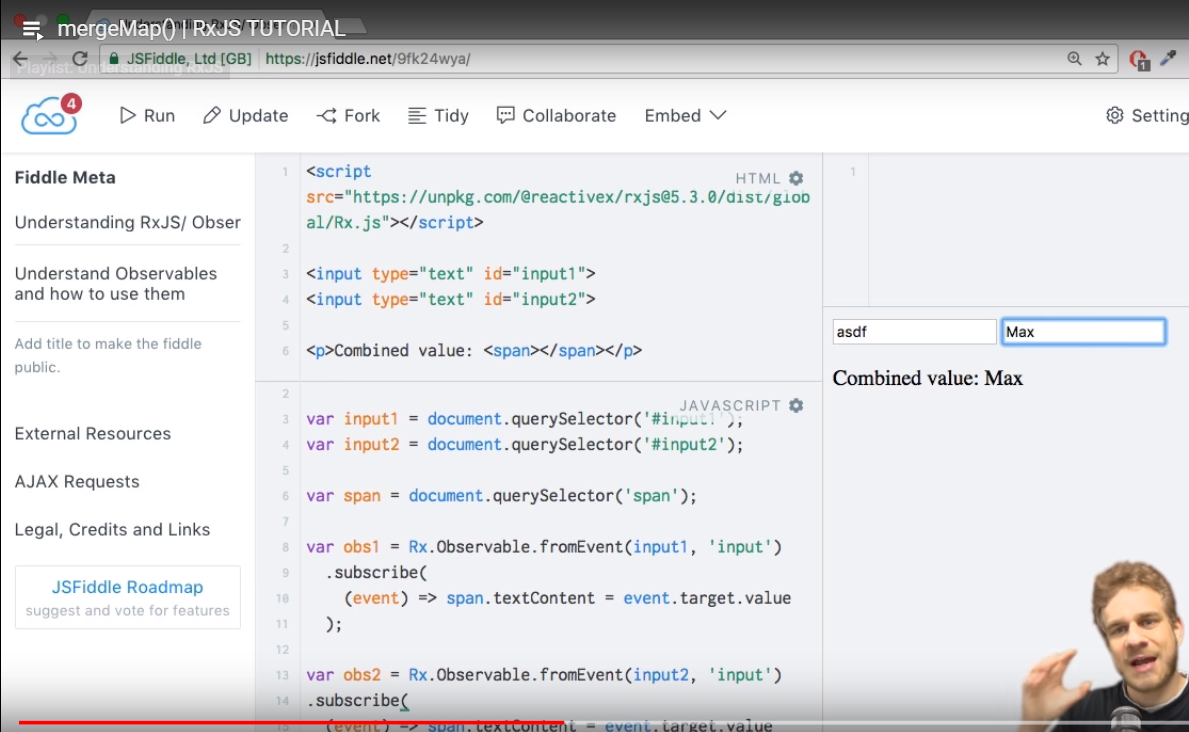
##### Behaviour Subject

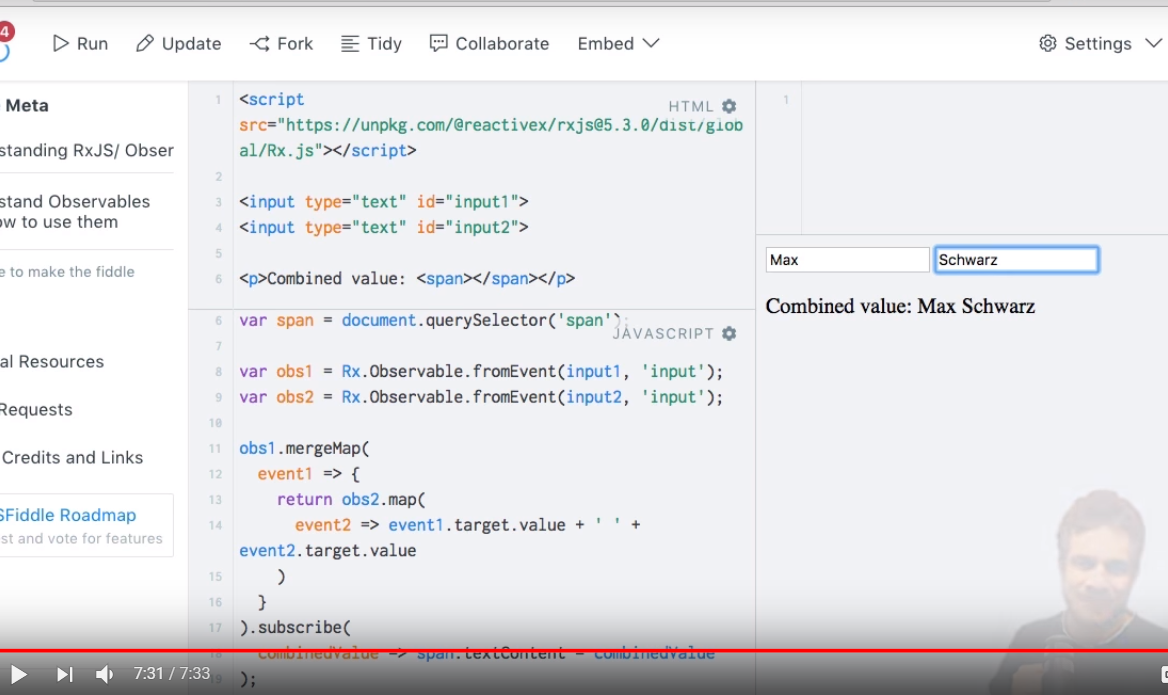
****

##### Debounce time

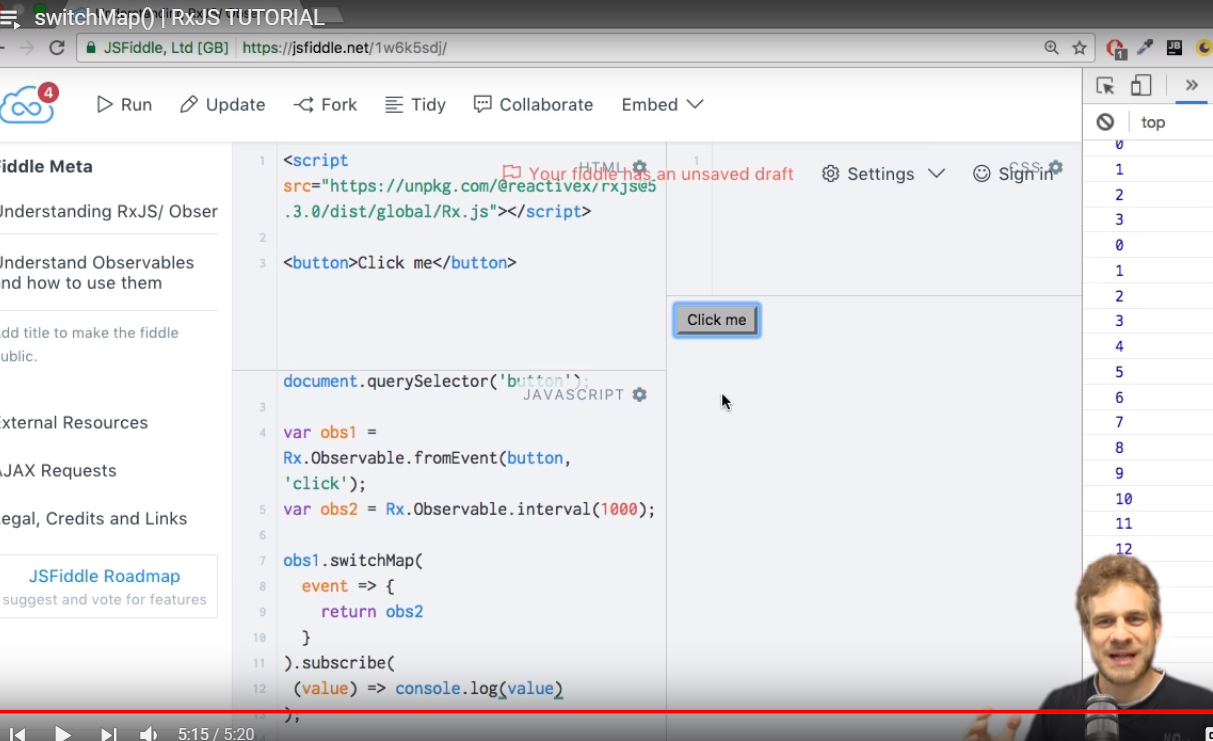
****

##### Merge Map

****

****

##### Switch Map

****