**RXJS - Library**

**RXJS Marble Diagram**

[www.rxmarbles.com](http://www.rxmarbles.com)

-------

Reactive programming me kaam ati h

Observable ko use krke asynchronous task ko krte h

RXJS ko kahi bhi use kr skte h

---

Angular me automatic install hoke ati h

----

Observable - RXJX me introduce hua h - bht imp h

--

Observable - Data source

Isme items hote h - Observable ka data

Iska kaam hota h tym to tym data ko emmit krna

agr hme data chahiye to observable k data ko subscribe krna pdega.

-

agr data kaam ka hoga to le lega nhi to ignore krega

--

**Operators of RXJS**

Filter, Map, From, Of, toArray, Take, merge

----------------------------------------------------------------------------

**Observer and Subscribe**

jb data chahiye observable mese to subscribe krenga

or jo code likhenge subcribe k lye use bolte h observer

---

map bht imp hota h

isse apn data ko transform kr skte h

map(x => 10 \* x)

----------------------------------------------------------------------------

**Promise and Asynchronous:**

**D/f b/w Promise and Observable**

Promise | Observable

|

1. async | 1. async or sync (both)

2. Return Single Value | 2. Return Stream of value

- resolve (yes) | 0----0-----0

- reject (no) |

----------------------------------------------------------------------------

**Promise:**

definate - kuch chiz hm khud krne wale h

indefinate - Kisi se krwane wale h

-

Promise me 2 case honge

agr kisi ko samaan lene k lye bhja h

to yato samaan mil jayga ya nhi milega(reject)

-

is case me confirm nhi hota, hona h ya nhi hona h to wo async me ate h

in sb k lye promise use krte h

-

then - agr fulfill ho rha h promise to

catch - agr error arha h to

//Promise me data immediate nhi ata

ab hmne laptop lene bhja h, Dell ka ya HP ka agr nhi mile to esi ajao ye scenerio h

----------------------------------------------------------------------------

**Async Await :**

jb bhi hm fn se phle async likhte h tb wo return krta h promise

rukta h promise pura hone tk - rok k rakhta h

----------------------------------------------------------------------------

**Observable Streams**

in sb chizo se milti h

- User Input (Button Click Event)

- Http Request

- Array

- Object

--

Observable Handle(subscribe)

- Data

- Error

- Completion

--

Syntax

.subscribe(

(data) => log(data),

(error) => log(error),

() => log('Complete')

)

----------------------------------------------------------------------------

**Interval:**

#isse har 2(n) sec me stream crreate hogi

----------------------------------------------------------------------------

**Timer:**

same as interval bs e 2 argument leta h

(delay, interval)

----------------------------------------------------------------------------

**Of() operator**

Convert the arguments to an observable sequence

{ ki value -- &#123

} - &#125

----------------------------------------------------------------------------

**From() Operator**

Array ko convert kr skte h observable stream me

----------------------------------------------------------------------------

**ToArray Operator:**

collect all data then convert it into array

----------------------------------------------------------------------------

**Custom Observable**

Custom Interval Observable

ab jb kisi dusre component pr jayenge to ruk jayga

----------------------------------------------------------------------------

**MAP Operator:**

data ko tranform krne k lye use krte h.

API se data mila or same wesa use nhi krna h thoda change krna h tb use krte h ise

return krta h new observable

map(Aata=> Chapati)

----------------------------------------------------------------------------

**Pluck Operator in RXJS:**

data k object ki kisi property ka data received krne k lye use krte h

----------------------------------------------------------------------------

**Filter operator in RXJS:**

Kuch filter krna ho tb use krte h

----------------------------------------------------------------------------

**TAP/Do Operator in RXJS :**

transparently perform actions or side-effects, such as logging

agr map me direct condition lagate to map gadbad ho jata. Islye tap use kra h

or agr bht sare operators h or apn ko bich me data check krna h

to TAP use krte h.

----------------------------------------------------------------------------

**Take Operators in RXJS :**

observable streams me se jitna data chahiye utna data leke use kr skte h

----------------------------------------------------------------------------

**Retry, RetryWhen, Scan and Delay Operator in RXJS :**

Kisi chiz ko retry krne k lye use krte h

For example - server se response nhi aya to retry krega k response mil jaye.

**retry** - bar2 retry krega data get krne k lye ek limit tk

**retrywhen** - ek time limit k bad try krega

**delay** - ye btayga kitni der bad re try krna h

**scan** - ye btayga kb rukna h- kitne attempt k bad

----------------------------------------------------------------------------

**DebounceTime :**

jab hm kuch search krte h tb use hota h

har bar request server pr na jaye islye use krte h

0.5sec k bad request server pr jaygi

----------------------------------------------------------------------------

**DistinctUntilChanged :**

bar2 request nhi jana chahiye

jiski request ek bar ja chuki h, uski request dobara nhi jaye

Switch map me bhi use hota h ye

----

jitni bar bhi likh rhe hoge utni bar log ho rha hoga

mtlb utni bar request bhjega server pr (jo sahi nhi h)

islye debounceTime use krenge

----------------------------------------------------------------------------

**Concat Map (sneak peek)**

Use for mobile notifications

----------------------------------------------------------------------------

**Subject and BehaviorSubject:**

Special typ ka Observable h

mtlb observer bhi h and observable bhi h

Observable

- Pipe

- Subscribe

----

Observer

- Next

- Error

- Complete

----

cross compoent communication k lye use krte h

data communication me use krte h

jaha par cmponent me parent child relation nhi h

or apn ko data send karna h tb use krte h Subject

------

**Event Emitter :**

Isse hm child ka data send kar skte h parent me

Parent Data Child

(emit)

-

Subject bhi Event Emitter ki tarha kaam krta h

Componet2 Data Component1 Data Service

(next) (next)

-

Subject me multiCasting bhi kr skte h.

jaha se data mil rha h whi se dekh payenge

purana nhi dikhega

-----

ab story ye h k apn ko wo chahiye jab subject ka component open kre tb

header me exclusive likha hua aaye

-

then story ye h k comp1 me hm input de or har jagha wo show ho jaye

eg. Subject me, comp2 me, comp3 me

**BehaviorSubject:**

Same as subject bs isme ek initial value hoti h

----------------------------------------------------------------------------

**Replace Subject:**

multiple value ko store krke rakhta h

or value kitne tym tk store rakhni h

kitne tym phle emit hui h, ye sb kr skte h

example

kisi ne subscribe kra h youtube channel.

to apn daal skte h last 5 days k 5 video ki notification dikha do

agr last 5 days me 8 videos dali h to bs 5 hi dikhayga

----------------------------------------------------------------------------

**Async Subject:**

ye value ko rok k rakhega

fir last wali video(data) dikha dega

----------------------------------------------------------------------------

**Concat Operators:**

agr 2 stream h to phle first stream ka data lega jb wo complete ho jaygi then dusri stream ka lega.

for exampple - jb ek project ka kaam khtm krega then dusra start krega

----------------------------------------------------------------------------

**Merge Operator:**

Jese2 stream milti rhegi merge krta rhega

for example- ek project kr rha h, dusre project ka bug agya to phle bug resolve kra

----------------------------------------------------------------------------

**MergeMap Operator in RXJS:**

It is a flattening operator

subscribe k andar ek or ko subscribe krke value produce krna

isko thk krke k deta h

mergeMap = map + mergeAll

----------------------------------------------------------------------------

**ConcatMap:**

Flattening operator - ye order me kaam krta h

jese jese data milta jayga ye subscribe krta jayga

similar h mergeMap jese bs mergeMap ek bar me value de deta h

or concatMap await jesa hota h, ek kaam complete hoga tb dusra krega.

----------------------------------------------------------------------------

**SwitchMap Operator :**

Flatening Operator - Isme hota h cancelling effect - request cancel krta h

jab saerch kre kuch likhe then hata kar kuch new likh de, tab wo old request cancel kr dega, or new request bhejega .

SwitchMap operator (for searching) :

----------------------------------------------------------------------------

**ExhaustMap Operator:**

DND typa ka hota h

kisi request ko multiple tym send krenge to ek bar hi jaygi request bs

Jab tk ek request complete nhi hoga, dusri request nhi lega - ignore krega

----------------------------------------------------------------------------

**ShareReplay Operator :**

avaoid kr skte h duplicate http request – All products, Mobile, Laptop

----------------------------------------------------------------------------

**CombineLatest:**

2 stream ka data combine kr deta h

dono stream ki latest video ko combine krke de deta h

----------------------------------------------------------------------------

**WithLatestFrom:**

isme do hote h, master and slave

jb master change hoga tbi box create krega, nhi to kuch nhi krega

----------------------------------------------------------------------------

**ZIP:**

Jab dono source me change detect hoga tbi chalega

----------------------------------------------------------------------------

**ForkJoin:**

last value k sath kaam krta h

jb stream khtm hogi tb ayega last me (tb show krega)

----------------------------------------------------------------------------

**CatchError & ThrowError:**

agr error user ko dikhana ho - UI pr

Service me error likh k code throw krdo

Or bad me CatchError se lelo

==================================================================================