

Observables

Created by :
Sangeeta Joshi

Topic 1

- Async calls with ES6 promises:
- Limitations of promises
 - Can return only one value
 - Guaranteed response, i.e. can't be canceled

- Observable HTTP Events

A common operation in any web application is *getting or posting data to a server.*

Angular applications do this with the Http library, which previously *used Promises* to operate in an asynchronous manner.

The updated Http library now incorporates *Observables for triggering events and getting new data.*

Terminology

- Observable - data stream that pushes data over time
- Observer - consumer of observable
- Subscriber - connects observer with observable
- Operator - en-route data transformation

Observables

- The observer pattern is a software design pattern
- an object, called the subject, maintains a list of its dependents, called observers
- notifies them automatically of any state changes, usually by calling one of their methods.
- It is mainly used to implement distributed event handling systems.

Callbacks

- The use of callbacks certainly offer asynchronicity,
- but they are less scalable as they often become overly nested

Callbacks

- `getStuff(function(result){`
- `getMoreStuff(function(results){`
- `getSomeStuffForEachResult(results,`
`function(moreStuff){`
- `doTheThingWeWantedToDo(moreStuff);`
- `});`
- `});`
- `});`

Callbacks

- So what's the problem?
- this turns into spaghetti code really, quickly,
- and doesn't handle errors...

Promises

- Promises are often cited as the best way to handle asynchronicity,
- They have their own built-in object methods.
- The problem with Promises is that they are neither cancellable
- nor are they a good way to manage collections of data

why RxJS?

- What about Promises?
- Promises are good for solving asynchronous operations such as querying a service with an XMLHttpRequest, where the expected behavior is one value and then completion.
- The Reactive Extensions for JavaScript unifies both the world of Promises, callbacks as well as evented data such as DOM Input, Web Workers, Web Sockets.
- Once we have unified these concepts, this enables rich composition.

RxJS

- RxJS is an incredibly sophisticated library
 - initially built by Microsoft,
 - then revised by the good folks at Netflix
 - maintained by many open source contributors.
- It includes the use of observables,
 - to handle data that is streamed into collections,
 - RxJS offers a clean way to leverage the power of observables to work with complex data,
 - to be cancellable and to clean up after themselves.

- To properly manage data streams in app, we're going to want to use observables.
- And to properly use Observables Angular-powered app, we need to know RxJS API.

- Observable HTTP Events

A common operation in any web application is *getting or posting data to a server.*

Angular applications do this with the Http library, which previously *used Promises* to operate in an asynchronous manner.

The updated Http library now incorporates *Observables for triggering events and getting new data.*