Reactive Angular 2

Barcelona, 15th December 2016

Who we are?

- web developers
- backend/frontend
- js, jquery, backbone, angular.js, **@angular**

tmpo.io

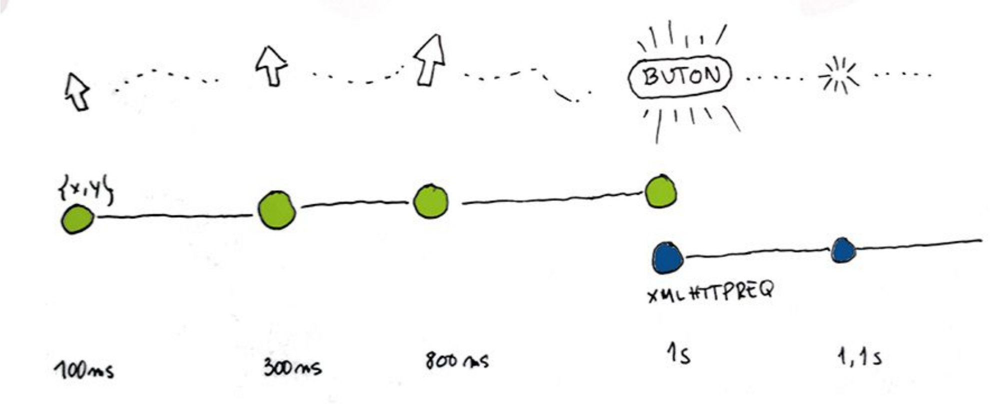
Oriol Collell
@oriolcp

Jordi Collell
@galigan

Agenda

- Observables
- Observables in Angular
  - Templates
  - Http
  - Router
  - Forms
- State Management (@ngrx)

What's an Observable?



A sequence of data over **TIME**

TIME is one of the main differences between back and front.

What's RxJS?

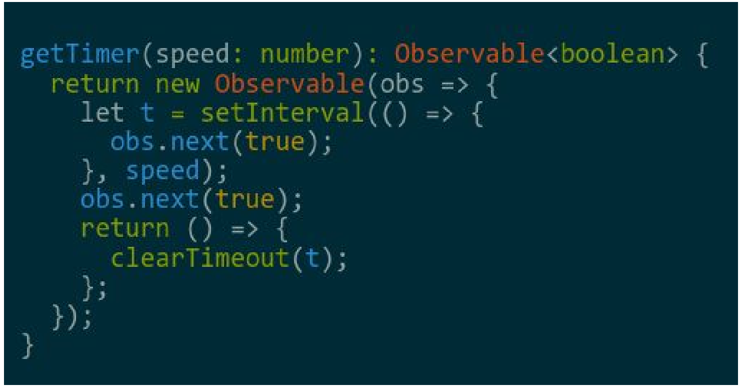
An API for asynchronous programming with observable streams

@benlesh
"Lodash for async"

A toolbox for working with async operations

Observables

A declarative way of defining async operations:



```
getTimer(speed: number): Observable<boolean> {
  return new Observable(obs => {
    let t = setInterval(() => {
      obs.next(true);
    }, speed);
    obs.next(true);
    return () => {
      clearTimeout(t);
    };
  });
}
```

Lazy:
Never executed till a subscription

Reusable:
Define them use when needed

Disposable:
Throw them away

Composable:
Compose streams to get new ones

vs ***Promises***

An Stream:
Promises = 1 value / Observables many

Cancelled:
*Can be disposed **after** subscription*

Composable:
Merge, join, concat, flat ...
any number of streams

BUT ALSO:

observable
.toPromise()

Templates | *async*

Send an observable to your template

Use the async pipe:
Will manage your subscription

```
// on Component
counter$ = Observable.interval(100)

// template
{{ counter$ | async }}

// result, every 100ms
1, 2, 3, 4, 5
```

Also works with Promises
https://github.com/angular/angular/blob/2.3.0/modules/%40angular/common/src/pipes/async_pipe.ts#L41-L140

Http

Query your endpoint

Transform the response

Catch Errors

… or retry

```
// on a service
res$ = this.http.get('/api')
   .map(r => r.json())
   .map(a => a.filter( b => b == 1))
 .catch((err, obs) => {
        logger(err); return obs.delay(500);
  })

// On template
<li *ngFor="let item of (res$ | async)">
   {{ item }}
</li>
```

Router

Take the observable from router

ActivatedRoute

transform the stream to a new one that handles an http request

Let the async pipe manage the subscription for us

```
// on a component, ActivatedRoute
res$ = this.route.params
   .map((p:any) => p.num)
   .switchMap(
         num => this.http.get('url', {num})
   );

// On template
<li *ngFor="let item of (res$ | async)">
    {{ item }}
</li>
```

Observables in Angular

Forms

Observable from a form element valueChanges

only one request each 500ms

switchMap cancels the pending request, when new one arrives

```
num = new FormControl()
res$ = num.valueChanges
   .distinctUntilChanged()
   .debounce(500)
   .switchMap(
        num => this.http.get('url', {num})
   );

// On template
<li *ngFor="let item of (res$ | async)">
    {{ item }}
</li>
```

Observables in Angular

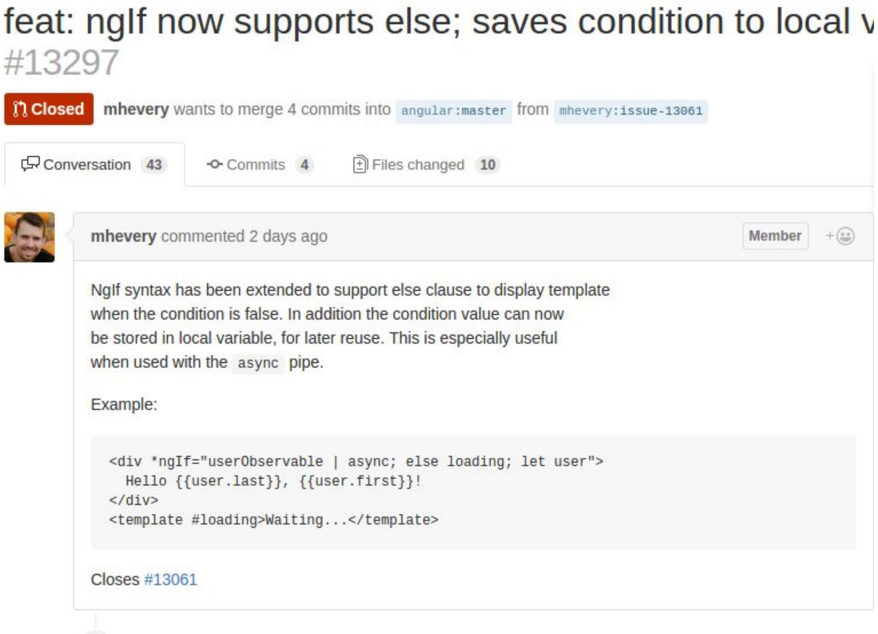
Incoming...

@angular will be more reactive

> local assignment on template via NgIf expression.
> Support cold DOM event streams #1348

...

Angular 4



feat: ngIf now supports else; saves condition to local v
#13297

Closed **mhevery** wants to merge 4 commits into `angular:master` from `mhevery:issue-13061`

mhevery commented 2 days ago

NgIf syntax has been extended to support else clause to display template when the condition is false. In addition the condition value can now be stored in local variable, for later reuse. This is especially useful when used with the `async` pipe.

Example:

```
<div *ngIf="userObservable | async; else loading; let user">
  Hello {{user.last}}, {{user.first}}!
</div>
<template #loading>Waiting...</template>
```

Closes #13061

@ngrx

RxJS powered state management for Angular applications, inspired by Redux

Knowing more...

https://reactivex.io/rxjs
https://reactivex.io/learnrx/

Ben Lesh (Angular Connect)
KOOT7BArVHQ, 3LKMwkuK0ZE

Rob Wormald
mhA7zZ23Odw, xAEFTSMEgIQ

André Staltz
https://egghead.io/technologies/rx

Also there is an active gitter room:
https://gitter.im/Reactive-Extensions/RxJS

MarbelTree

Merry XMAS!