

Writing Angular for Evergreen

Galen Charlton

26 April 2019

Evergreen International Conference



Goals

- Learning how to count cats properly
- ... by writing an Angular component
- ... that uses other Angular components written for Evergreen
- ... and letting the counted cats play in the sandbox

What this presentation is not

- A deep dive into TypeScript
- A workshop as such (sorry!)
- Me standing in front of you with all the answers
- But you will have some example code to play with

Counting cats



No really, counting cats

For our purposes, a cat does not have pointy ears.

Instead, let's count **cat**aloging actions, i.e., edits to bibliographic or authority records.

And let's build a component to do it.

Requirements

1. Let user specify type of record (bibliographic or authority) whose edits we want to count
2. Let user specify the date they're interested in
3. Display the count in an innovative manner

Existing components and services we'll use

- eg-combobox
- eg-date-select
- PcrudService

```
10 ITERATE  
20 GOTO 10
```


Structure of a component

- TypeScript source
- HTML template
- CSS

E.g.,

`cat-counter.component.ts`

`cat-counter.component.html`

`cat-counter.component.css`

Let's start with the HTML

```
<div class="eg-cat-counter">  
  <span i18n>Cat counter: {{count}}</span>  
</div>
```

Let's start with the HTML

```
<span i18n>Cat counter:</span>
```

```
<span *ngIf="count == 0" i18n>
```

```
  No cats have been counted yet
```

```
</span>
```

```
<span *ngFor="let x of [].constructor(count)"> </span>
```

```
<span *ngIf="count > 0" i18n>({{count}})</span>
```

Continuing with the HTML

```
<eg-combobox #recordTypeSelector  
  id="type-select"  
  (onChange)="selectRecordType($event)"  
  [startId]="initialRecordType"  
  [startIdFiresOnChange]="true"  
  placeholder="Record Type..." i18n-placeholder>
```

...

Open-ILS/src/eg2/src/app/share/cat-counter/cat-counter.component.html

Continuing with the HTML

```
<eg-combobox-entry entryId="biblio"  
  entryLabel="Bibliographic"  
  i18n-entryLabel></eg-combobox-entry>  
<eg-combobox-entry entryId="authority"  
  entryLabel="Authority"  
  i18n-entryLabel></eg-combobox-entry>  
</eg-combobox>
```

Continuing with the HTML

```
<div class="eg-cat-counter">
  <span i18n>Cat counter:</span>
  <eg-combobox #recordTypeSelector
    id="type-select"
    (onChange)="selectRecordType($event)"
    [startId]="initialRecordType" [startIdFiresOnChange]="true"
    placeholder="Record Type..." i18n-placeholder>
    <eg-combobox-entry entryId="biblio" entryLabel="Bibliographic"
      i18n-entryLabel></eg-combobox-entry>
    <eg-combobox-entry entryId="authority" entryLabel="Authority"
      i18n-entryLabel></eg-combobox-entry>
  </eg-combobox>
  <eg-date-select (onChangeAsDate)="updateCount($event)"></eg-date-select>
  <span *ngIf="count == 0" i18n>No cats have been counted yet</span>
  <span *ngFor="let x of [].constructor(count)">🐱</span>
  <span *ngIf="count > 0" i18n>({{count}})</span>
</div>
```

Open-ILS/src/eg2/src/app/share/cat-counter/cat-counter.component.html

Very brief CSS digression

```
.eg-cat-counter {  
    border-style: dotted;  
    padding: 4px;  
    margin: 4px;  
}
```

Building a component

```
import {Component, Input, Output} from '@angular/core';
import {ComboboxEntry} from
  '@eg/share/combobox/combobox.component';
import {CatCounterService} from './cat-counter.service';

@Component({
  selector: 'eg-cat-counter',
  templateUrl: 'cat-counter.component.html',
  styleUrls: ['cat-counter.component.css']
})
```


Building a component

```
export class CatCounterComponent {
  @Input() initialRecordType: string;
  @Output() count: number = 0;

  private recordType: string;

  constructor(
    private counter: CatCounterService
  ) {
    this.recordType = this.initialRecordType;
  }
}
```

Building a component

```
selectRecordType($event: ComboboxEntry) {  
    this.recordType = $event.id;  
}  
updateCount($event) {  
    this.counter.getCatCount(  
        this.recordType, $event  
    ).then(ct => {  
        this.count = ct;  
    });  
}
```

HTML callback

```
<eg-combobox #recordTypeSelector
  id="type-select"
  (onChange)="selectRecordType($event)"
  [startId]="initialRecordType" [startIdFiresOnChange]="true"
  placeholder="Record Type..." i18n-placeholder>
  <eg-combobox-entry entryId="biblio" entryLabel="Bibliographic"
    i18n-entryLabel></eg-combobox-entry>
  <eg-combobox-entry entryId="authority" entryLabel="Authority"
    i18n-entryLabel></eg-combobox-entry>
</eg-combobox>

<eg-date-select (onChangeAsDate)="updateCount($event)"></eg-date-select>
```



Wait, we're
missing
something

Ahhhhh....

}

Having a service

```
import {Injectable} from '@angular/core';  
import {PcrudService} from '@eg/core/pcrud.service';
```

```
@Injectable()  
export class CatCounterService {  
  
    constructor(  
        private pcrud: PcrudService  
    ) { }  
  
}
```

Stupid date math tricks

```
addDay(date: Date): Date {  
    let newDate = new Date(date);  
    newDate.setDate(date.getDate() + 1);  
    return newDate;  
}
```

Counting records, inefficiently

```
getCatCount(recordType: string, date: Date): Promise<number>
{
  const cls = (recordType === 'authority') ? 'are' : 'bre';

  let search: any = [
    { id: { "!=" : null } },
    { edit_date: { ">" : date.toISOString() } },
    { edit_date: { "<=" : this.addDay(date).toISOString() } },
  ];
```


Counting records, inefficiently

```
return this.pcrud.search(  
  cls, search, {}, {atomic:true}  
).toPromise().then(list => {  
  return list.length;  
});
```

Ahhhhh....

}

Open-ILS/src/eg2/src/app/share/cat-counter/cat-counter.service.ts

Playing in the sandbox

```
<div class="row flex pt-2">
  <div id="Counting cats">
    <div class="col-lg-3">
      <eg-cat-counter
initialRecordType="biblio"></eg-cat-counter>
    </div>
    <div class="col-lg-3">
      <eg-cat-counter
initialRecordType="authority"></eg-cat-counter>
    </div>
  </div>
```

Playing in the sandbox

```
import {NgModule} from '@angular/core';
import {StaffCommonModule} from '@eg/staff/common.module';
import {SandboxRoutingModule} from './routing.module';
import {SandboxComponent} from './sandbox.component';
import {CatCounterComponent} from
  '@eg/share/cat-counter/cat-counter.component';
import {CatCounterService} from
  '@eg/share/cat-counter/cat-counter.service';
```

Playing in the sandbox

```
@NgModule({
  declarations: [
    SandboxComponent,
    CatCounterComponent
  ],
  imports: [
    StaffCommonModule,
    SandboxRoutingModule,
  ],
  providers: [
    CatCounterService
  ]
})
```

Open-ILS/src/eg2/src/app/staff/sandbox/sandbox.component.ts

And the end result...

Counting cats:

Cat counter:

Bibliographic



2019-04-23



(23)

Cat counter:

Authority



2019-04-22



(14)

Making changes on the fly

- `ng-build --watch`
- `/openils/var/web/eg2 symlink`

Shifting from AngularJS to Angular

- Getting used to TypeScript and type constraints
- No magic two-way binding
- ngModel is still available, but you're going to be using promises and observables a lot more often
- but it's not *that* different

The code

user/gmcharlt/eg-cat-counter-evgils19 in the Evergreen repository:

<https://git.evergreen-ils.org/?p=working/Evergreen.git;a=shortlog;h=refs/heads/user/gmcharlt/eg-cat-counter-evgils19>

Thanks!

Galen Charlton

@gmcharlt

gmc@equinoxinitiative.org



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).