Angular Building Blocks

Goal

In this lab, you will:

- Create an Angular application consisting of multiple modules
- Create a Quotes component that uses a Quotes service to load quotes
- Use dependency injection to provide a default Quote service
- Overwrite dependency injection so multiple instances of the component can use different services
- Create a pipe to display only a limited set of quotes

Your mission

In this lab you will create a quote application that will display funny facts about Jon Skeet and Chuck Norris.

Type it out by hand?

> Typing it drills it into your brain much better than simply copying and pasting it. You're forming new neuron pathways. Those pathways are going to help you in the future. Help them out now.

Create a basic quote application using the Angular CLI

In this section, you will use the Angular CLI to create a simple to-do list application.

1. Open the **begin** folder of the lab. Notice there is nothing here yet.
2. Create a new Angular application named **quotes** using the Angular CLI. This can be done with the command: `ng new quotes`.
3. Make sure the application works by navigating to the new **quotes** folder, starting the web server using the CLI and opening this in the web browser at http://localhost:4200/.

```
cd quotes
ng serve
```

4. Create a new **quotes** module and make this the current folder in the console.

```
ng generate module quotes
cd src/app/quotes
```

5. Use the Angular CLI to generate a Quotes component and service

```
ng generate component quotes
ng generate service quotes
```

6. Open **quotes.service.ts** and add a **getQuotes()** function that returns a few basic quotes.

```
import { Injectable } from '@angular/core';

@Injectable()
export class QuotesService {

  constructor() { }

  getQuotes(): string[] {
    return [
      'This is a quote.',
      'This is another quote.',
      'Create your custom QuotesService to display other quote
s.'
    ]
  }
}
```

7. Open **quotes.module.ts** and add the **QuotesService** as a provider. Also export the **QuotesComponent** so it can be used in other parts of the application.

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { QuotesComponent } from './quotes/quotes.component';
import { QuotesService } from './quotes.service';

@NgModule({
  imports: [
```

```
    CommonModule
  ],
  providers: [QuotesService],
  declarations: [QuotesComponent],
  exports: [QuotesComponent]
})
export class QuotesModule { }
```

8. Open **app.module.ts** and import the **QuotesModule**.

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { HttpModule } from '@angular/http';

import { QuotesModule } from './quotes/quotes.module';

import { AppComponent } from './app.component';

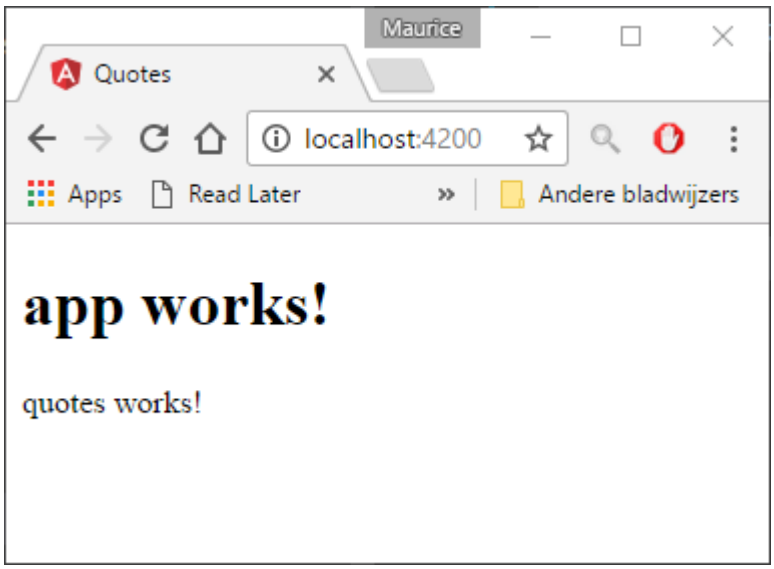
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    HttpModule,
    QuotesModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

9. Open **app.component.html** and add the **QuotesComponent**.

```
<h1>
  {{title}}
</h1>

<app-quotes></app-quotes>
```

10. Switch back to the browser. You should see it update automatically and show that quotes works!.



11. Open **quotes.component.ts** and inject the **QuotesService** into the constructor as a private property. Add a quotes property and populate this from the **ngOnInit()** life-cycle function by calling the **QuotesService.getQuotes()** function.

```
import { Component, OnInit } from '@angular/core';
import { QuotesService } from '../quotes.service';

@Component({
  selector: 'app-quotes',
  templateUrl: './quotes.component.html',
  styleUrls: ['./quotes.component.css']
})
export class QuotesComponent implements OnInit {

  quotes: string[];

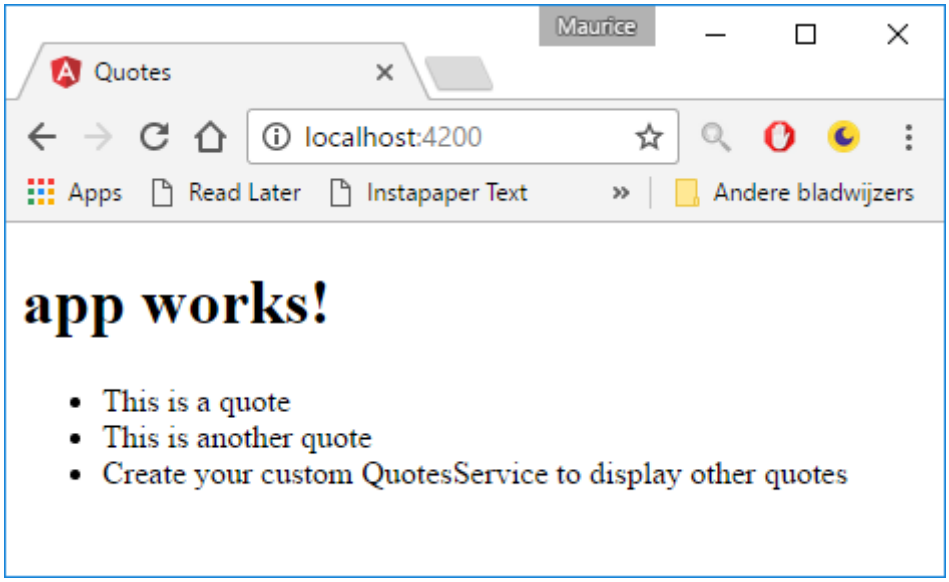
  constructor(private quotesService: QuotesService) { }

  ngOnInit() {
    this.quotes = this.quotesService.getQuotes();
  }
}
```

12. Open **quotes.component.html** and display the list of quotes using the **ngFor** directive.

```
<ul>
  <li *ngFor="let quote of quotes">
    {{quote}}
  </li>
</ul>
```

13. Switch back to the browser. You should see a list of quotes.



Display two lists of quotes

In this section you will use the same QuotesComponent and display a list of Jon Skeet quotes and a list of Chuck Norris quotes.

1. Change the active directory in the console window to the main **quotes** folder. Once there generate a **SkeetQuotesComponent** and a **SkeetQuotesService**.

```
cd ../../..
ng g c SkeetQuotes
ng g s SkeetQuotes
```

2. Open **app.component.html** and replace the **app-quotes** with the **app-skeet-quotes** component.

```
<h1>
  {{title}}
</h1>

<app-skeet-quotes></app-skeet-quotes>
```

3. Open **skeet-quotes.component.html** and replace the content with a header stating that these are Jon Skeet quotes and the original **QuotesComponent**.

```
<h2>Jon Skeet Quotes</h2>

<app-quotes></app-quotes>
```

4. Open **skeet-quotes.service.ts** and add a **getQuotes()** function returning some of the popular quotes about Jon Skeet. You can use a longer list of quotes from the file **Jon-Skeet.txt**

```
  getQuotes(): string[] {
    return [
      "Jon Skeet can divide by zero. Twice.",
      "When Jon Skeet's code fails to compile the compiler apo
logises.",
      'When you search for "guru" on Google it says "Did you m
ean Jon Skeet?"',
      "Jon Skeet can recite π. Backwards.",
      "When Jon Skeet points to null, null quakes in fear.",
      "Jon Skeet is the traveling salesman. Only he knows the
 shortest route.",
      "Google is Jon Skeet behind a proxy."
    ]
  }
```

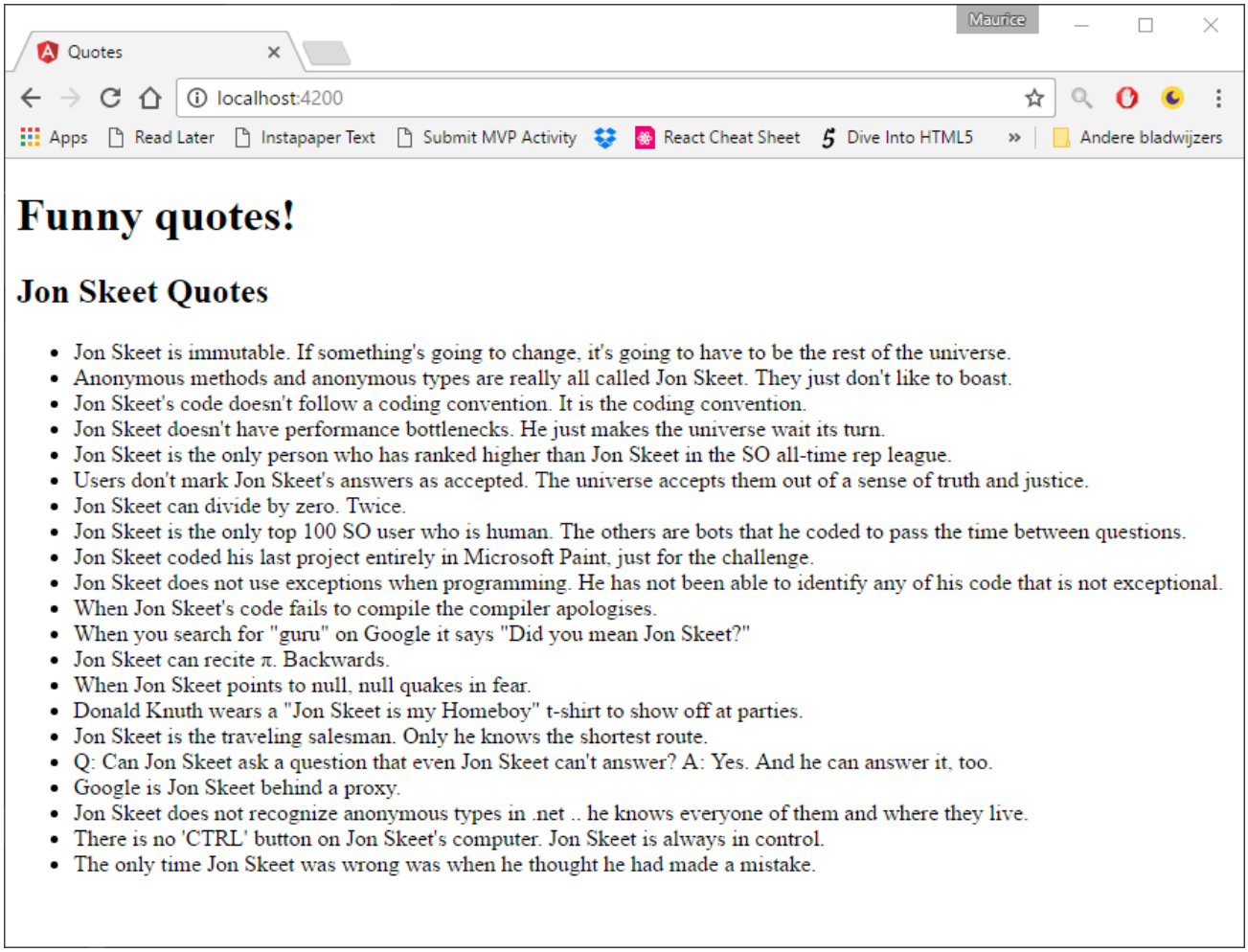
5. Open **skeet-quotes.component.ts** and provide the class **SkeetQuotesService** to use instead of the **QuotesService** in all nested components.

```
@Component({
  selector: 'app-skeet-quotes',
  templateUrl: './skeet-quotes.component.html',
  styleUrls: ['./skeet-quotes.component.css'],
  providers: [{
    provide: QuotesService, useClass: SkeetQuotesService
  }]
})
export class SkeetQuotesComponent implements OnInit {

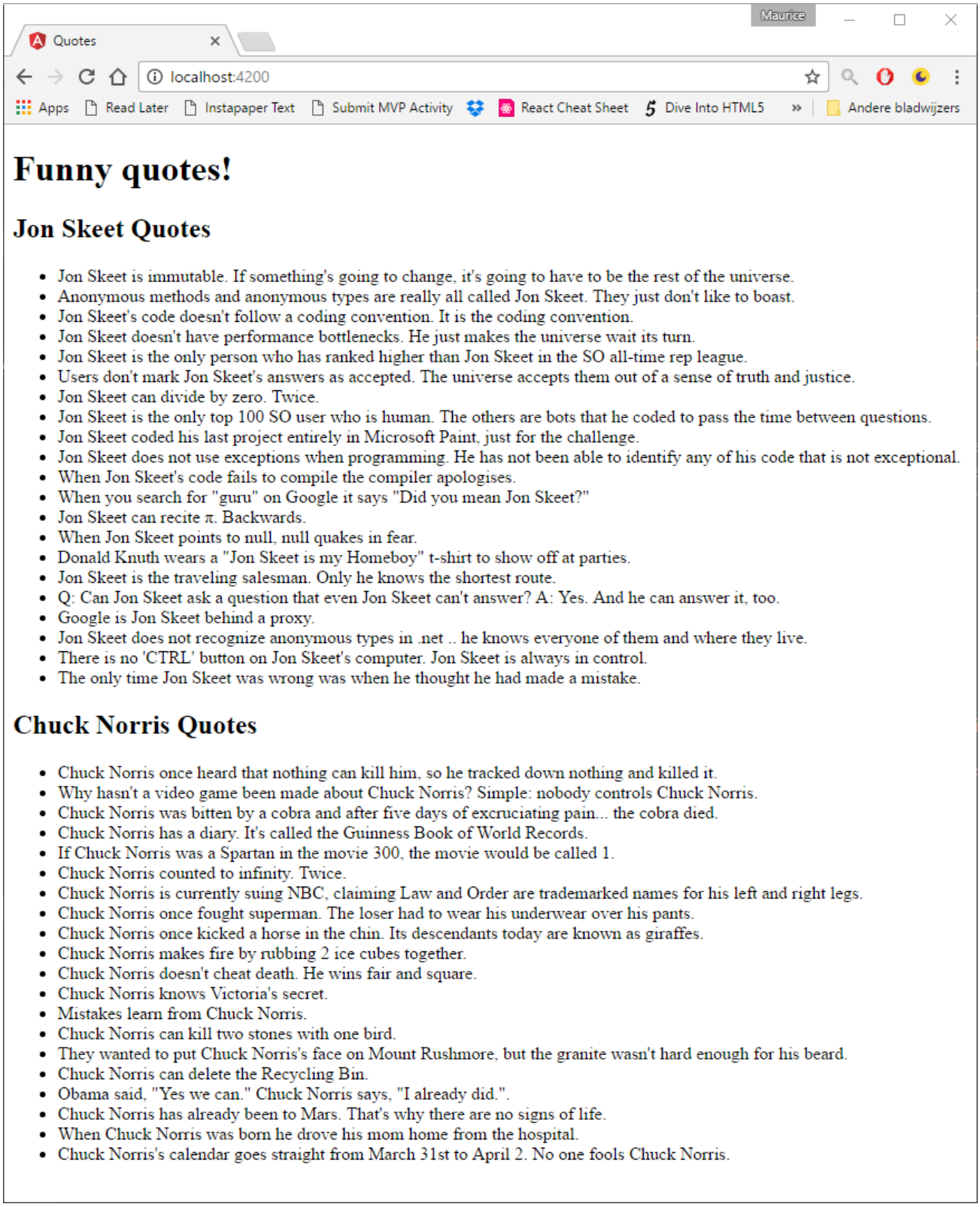
  constructor() { }
```

```
  ngOnInit() {
  }
}
```

6. Switch back to the browser. You should see the list of Jon Skeet quotes.



7. Repeat this process for an additional list of Chuck Norris facts. There is a list of facts in **Chuck-Norris.txt**.



Create a pipe to display random facts

1. Create a new pipe named **RandomizePipe**.

```
cd .\src\app\quotes
ng g p randomize
```

2. Open **randomize.pipe.ts** and implement the **transform()** function to sort the array in a random order and take the first 5 items.

```
import { Pipe, PipeTransform } from '@angular/core';

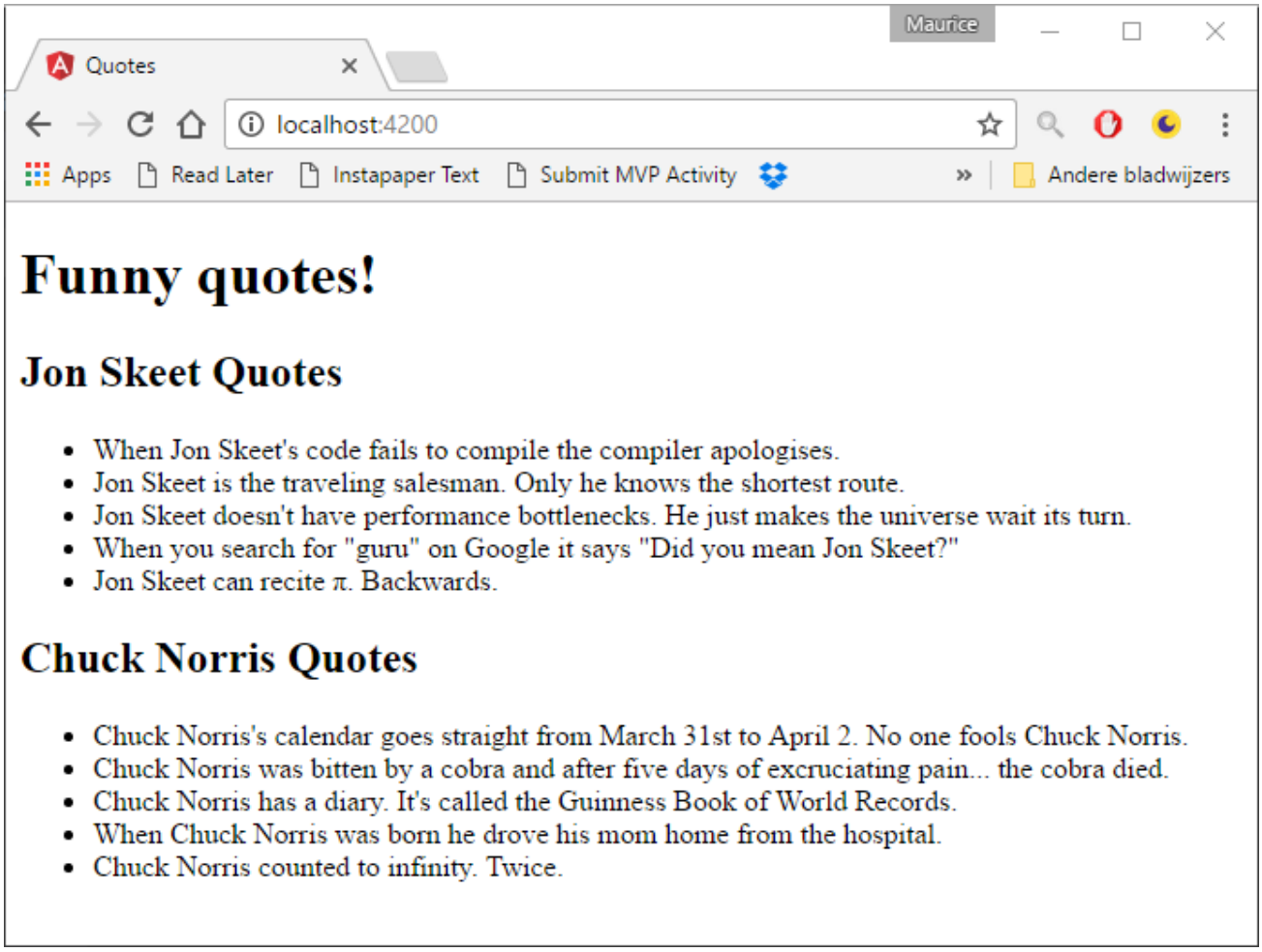
@Pipe({
  name: 'randomize'
})
export class RandomizePipe implements PipeTransform {

  transform(value: string[], args?: any): string[] {
    return value
      .concat()
      .sort(() => Math.random() - 0.5)
      .slice(0, 5)
  }
}
```

3. Open **quotes.component.html** and update the markup to use the **RandomizePipe**.

```
<ul>
  <li *ngFor="let quote of quotes | randomize">
    {{quote}}
  </li>
</ul>
```

4. Switch back to the browser. You should see a random list of five Jon Skeet and five Chuck Norris quotes.



Make the number of quotes displayed configurable.

1. Add a **quoteCount** number to the **QuotesComponent** and set the value to 5.

```
export class QuotesComponent implements OnInit {

  quotes: string[];
  quoteCount: number = 5;

  constructor(private quotesService: QuotesService) { }

  ngOnInit() {
    this.quotes = this.quotesService.getQuotes();
  }
}

2. Open **quotes.component.html** and add the **quoteCount** p
roperty to the **randomize** pipe. Add a data bound input cont
rol to update the **quoteCount** property. Make sure the **quo
```

```
teCount** is always between 0 and the available number of quot
es using the **min** and **max** properties.

```html
<div>
  Number of quotes:
  <input type="number" [(ngModel)]="quoteCount" min=0 [max]="q
uotes.length">
</div>
<ul>
  <li *ngFor="let quote of quotes | randomize:quoteCount">
    {{quote}}
  </li>
</ul>
```

3. Open **quotes.module.ts** and make use to import the **FormsModule** module for the **ngModel** directive to work.

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
import { QuotesComponent } from './quotes/quotes.component';
import { QuotesService } from './quotes.service';
import { RandomizePipe } from './randomize.pipe';

@NgModule({
  imports: [
    CommonModule,
    FormsModule
  ],
  providers: [QuotesService],
  declarations: [QuotesComponent, RandomizePipe],
  exports: [QuotesComponent]
})
export class QuotesModule { }
```

4. Open **randomize.pipe.ts** and update the **transform()** function to take the **quoteCount** as the argument. Use 5 as the default in case it is not sepecified and return the number of quotes.

```
export class RandomizePipe implements PipeTransform {
  transform(value: string[], quoteCount: number = 5): string[]
```
```

```
  {
    return value
      .concat()
      .sort(() => Math.random() - 0.5)
      .slice(0, quoteCount)
  }
}
```

 ```

Solution

The Solution can be found in **complete** folder