Angular

From a .Net developer's perspective

Isaac Martinez
Software Architect

Twitter: @IkeMtz
Repo: github.com/PcAlchi/ngTodo

Goals of the walk through

- Connect to ToDo micro service
- Get List Of ToDo
- Add bootstrap 4 (styling only)
- Create ToDo form

The Todo MicroService

github.com/pcalchi/Todo.core

- Build on .net Core
- Will run on docker, kestrel, IIS and Azure
- Connects to SQL Database (sql db project schema included)
- Restful API that gets, upserts and deletes todos

Why Angular?

- It’s a COMPLETE framework, with an impressive toolset.
- DI and IoC from the get go!
- The tooling makes it easy to adopt standards across a team.
- It’s backed by Google and it’s used internally on a variety of applications.
- It’s constantly updated with new features and bug fixes.
- Community.

Recommended Tools

- VS Code
- Augury
  augury.angular.io/
- Angular-Cli
  cli.angular.io
- TSLint (VS Code Addon)

Requirements for demo

- Node.Js
  https://nodejs.org/en/

- Angular CLI
  Available via NPM (Node Package Manager)
  `“npm i -g @angular/cli”`

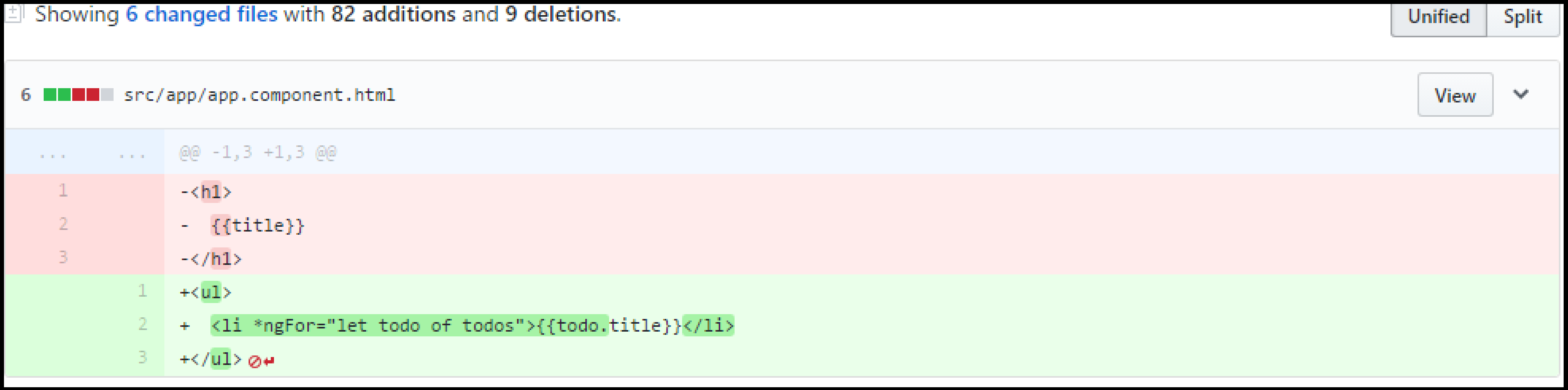
All items in courier and gray are terminal (console) commands that must be executed within the project folder structure .

- An instance of the Todo.core micro service.
  Note: Keep in mind that you’ll need to update the baseUrl field in the todo.service.ts file so angular can communicate with it.

Pairing the slide deck with the GitHub repo

The repo is broken out into steps, each step is to a branch.

To illustrate the changes implemented per step, look at the commits in each individual branch/step. You can use GitHub or a GitHub client of your choice.



Showing 6 changed files with 82 additions and 9 deletions.

```
6 src/app/app.component.html

@@ -1,3 +1,3 @@
-<h1>
-  {{title}}
-</h1>
+<ul>
+  <li *ngFor="let todo of todos">{{todo.title}}</li>
+</ul>
```

Step 1: Hello World

- Create new ToDo project, this will generate a brand new Angular CLI project.
  `“ng new ngTodo”`
- Run the application, this will start a development web server to host your application. This server is typically setup to listen at http://localhost:4200.
  `“ng serve”` or `“ng serve -o”`
- Modify the src/app/app.component.html to have the page render “Hello World”.
- Notice that one-way data binding is achieved via {{…}}.

Step 2: Getting Data from Todo API

Services: These classes provide a layer of abstraction from outside dependencies, facilitating mocking and unit testing.

- Add a service
  `“ng g Service todo-service”`
- Generation of services are not injected to any of your modules by the CLI, you’ll have to do this manually. App.module.ts
- Add Http provider dependency to the todo-service.

Step 3: Making it pretty with Bootstrap 4

To download Bootstrap 4, use the following command:
`“npm i --save bootstrap@4.0.0-alpha6”`

.angular-cli.json is the equivalent of .csproj file for a C# project. We’ll be adding bootstrap4 as a part of the project by adding it to the styles collection.

It’s important to note that the Angular CLI development web server will watch all files under the app folder.

As the .angular-cli.json file is outside of this folder, changes to this file require restarting the web development server.
`“ng serve”` or `“ng serve –o”`

Step 4: ng-If directive

- ng-if will not add nested items to the DOM if the condition is false.

  As this is not part of the DOM by default, you will not see the nested items in the DOM inspector until you click “Add Todo”.
- Notice how we’re doing event binding **(click)=“mymethod()”**; Angular supports all events available in HTML5.

Step 5: Simple Angular Form

- There are a couple of ways to do forms in Angular. The more complex way is required when you have complex forms where portions of the form are dynamically created. The simple way will likely work for you.
- Notice how we do two way data binding **[(ngModel)]="BLAHBLAH"** on the text fields.

Step 6: Posting data to Todo API service

- Extending Todo Service to handle posting of data to the RESTful API.
- Notice that we’re abstracting the HTTP post logic from our component as this is all contained in our Todo Service.

Step 7: Deleting data

- Once again extending Todo service to invoke HTTP delete call on the Todo API.

Additional Resources

- Angular Tour of heroes
  https://angular.io/docs/ts/latest/tutorial/
- Bootstrap V4
  https://v4-alpha.getbootstrap.com/getting-started/introduction/
- Angular CLI
  https://github.com/angular/angular-cli
- My Angular CLI Demo Repo
  https://github.com/pcalchi/ng4CliDemo