# Creating a Boring To Do List app with Angular 2

* Node.js - <https://www.youtube.com/watch?v=BcynBGSoQ1I>
* TypeScript – <https://www.typescriptlang.org/docs/tutorial.html>
* WebPack - <https://webpack.github.io/docs/tutorials/getting-started/>

## Choose your IDE. In the lecture we will use VS Code

With plugin for template strings - <https://marketplace.visualstudio.com/items?itemName=natewallace.angular2-inline>

## Install Node.js > 6.x and npm > 3.x

## Install `typescript`, `webpack`, `webpack-dev-server`, `tslint` globally

npm install --global typescript webpack webpack-dev-server tslint

## Download initial template from the SoftUni system

## Restore packages & fire the dev server

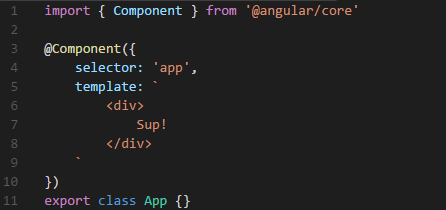
npm install

npm start

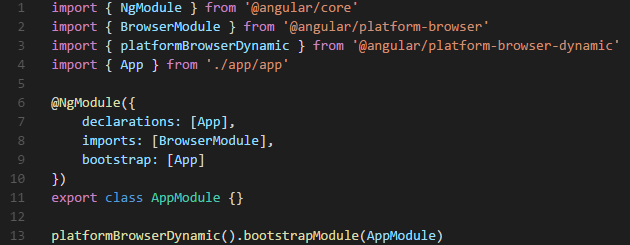
Go to <http://localhost:3000>

## Bootstrap the application

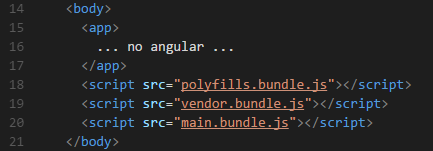
Create your first component in `app/app.ts`



Add the following bootstrap code in `main.ts`

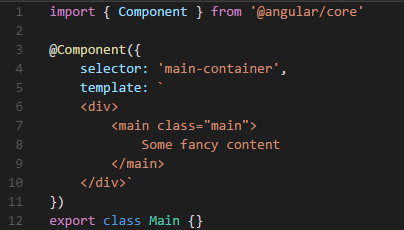


In the `index.html` add a simple `app` tag



## Add Component in Component

Add `main.ts` in `app/containers`:

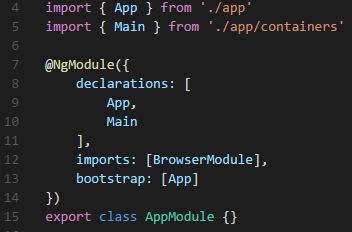


Add `index.ts` in `app/containers`:



You may do the same for the `app` folder.

Add the `Main` component in the declarations in the `main.ts` file:



Use the `Main` component in the `App` component.

## Add UI components

Create `navbar.ts` in `app/ui`



Create `index.ts` and add the component in the declarations and the `Main` container:



Use the following template:

|  |
| --- |
| <header class="app-bar row middle-xs">  <span class="logo col-xs-10">  Boring To Do List  </span>  <nav class="col-xs-2">  <div class="row middle-xs between-xs">  <span class="link">Settings</span>  <span class="link">signout</span>  </div>  </nav>  </header> |

And the following styles:

|  |
| --- |
| .app-bar {  height: 65px;  padding: 5px 30px;  background-color: #00BCD4;  margin-right: 0;  margin-left: 0;  }  .logo {  color: white;  font-size: 30px;  font-weight: 300;  cursor: pointer;  }  .link {  color: white;  font-size: 24px;  font-weight: 400;  cursor: pointer;  } |

## Add Note cards

Create `note-card.ts` in `app/ui`:



Export the component and add it in the declarations.

Add the following template:

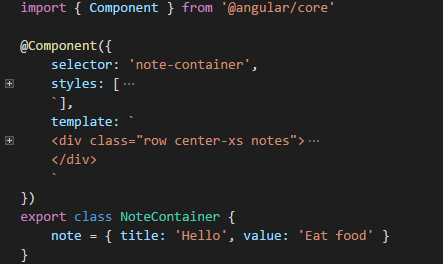
|  |
| --- |
| <div class="note-card row shadow-1">  <div class="icon" (click)="onChecked()">  <i class="material-icons">check</i>  </div>  <div class="col-xs-12 title">  **{{ notedata.title }}**  </div>  <div class="col-xs-12 value">  {{ notedata.value }}  </div>  </div> |

Add the following styles:

|  |
| --- |
| .note-card {  padding: 15px;  border-radius: 2px;  width: 100%;  position: relative;  }  .title {  font-size: 1.2rem;  font-weight: bold;  text-align: left;  color: rgba(0,0,0,0.8);  }  .value {  text-align: left;  font-size: 1.4rem;  font-weight: 200;  }  .icon {  position: absolute;  color: black;  border: 1px solid lightgrey;  background-color: white;  font-size: 30px;  top: -10px;  left: -10px;  width: 40px;  height: 40px;  border-radius: 100%;  cursor: pointer;  } |

## Add Note container

Add `notes.ts` in `app/containers`:



Export it and add it to the declarations.

Add the note container tag into the main element.

Add the following template:

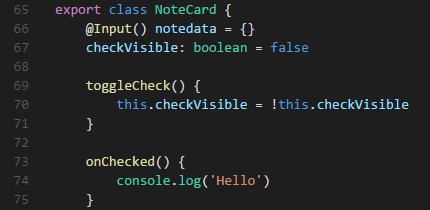
|  |
| --- |
| <div class="row center-xs notes">  <div class="col-xs-6 creator">  note creator here  </div>  <div class="notes col-xs-8">  <div class="row between-xs">  **<note-card**  class="col-xs-4"  **[notedata]="note"**  >  **</note-card>**  </div>  </div>  </div> |

Add the following styles:

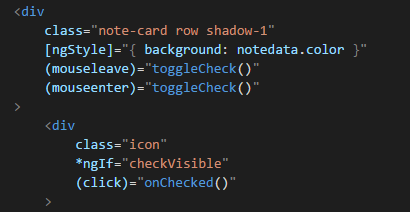
|  |
| --- |
| .notes {  padding-top: 50px;  }  .creator {  margin-bottom: 40px;  } |

## Make the Note dynamic and fancy

Add the following to the `NoteCard` class:

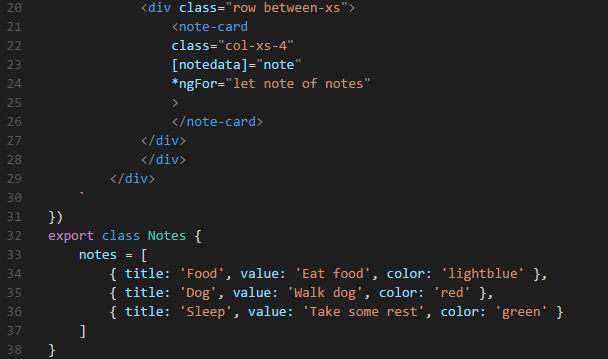


And then to the template:



## Display multiple notes

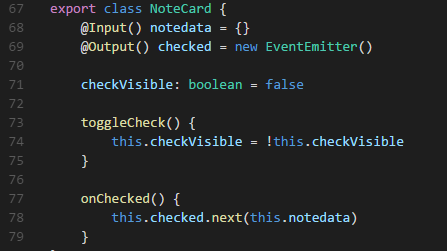
Change the notes container to have multiple notes:



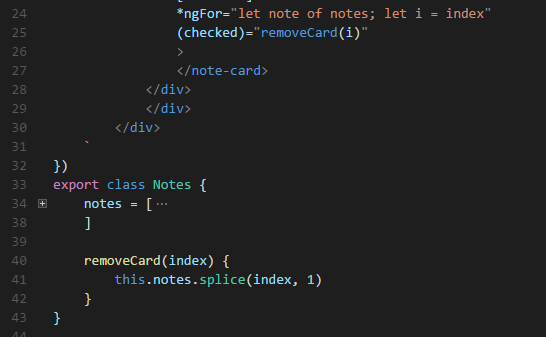
Don't forget the `\*ngFor` in the `note-card` tag.

## Make the check remove the note

Add output event in the `NoteCard` class:

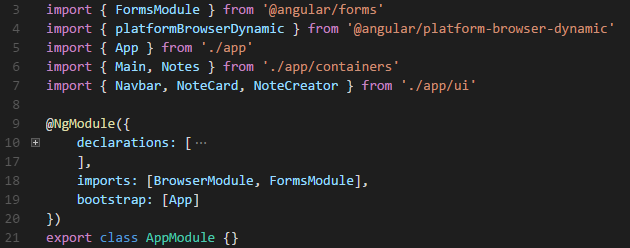


And then in the notes container:



## Add Note creator

Import `FormsModule` from `@angular/forms`:



Add `note-creator` in `app/ui` and export it:



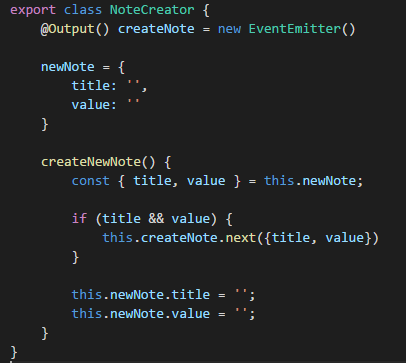
Add the following template:

|  |
| --- |
| <div class="note-creator shadow-2">  <form class="row">  <input  type="text"  **[(ngModel)]="newNote.title"**  name="newNoteTitle"  placeholder="Title"  class="col-xs-10 title"  >  <input  type="text"  **[(ngModel)]="newNote.value"**  name="newNoteValue"  placeholder="Take a note..."  class="col-xs-10"  >  <div class="actions col-xs-12 row between-xs">  <button  type="submit"  class="btn-light"  >  Done  </button>  </div>  </form>  </div> |

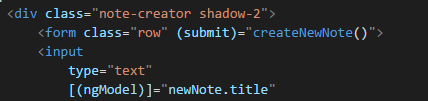
And the following styles:

|  |
| --- |
| .note-creator {  padding: 20px;  background-color: white;  border-radius: 3px;  }  .title {  font-weight: bold;  color: rgba(0,0,0,0.8);  }  .full {  height: 100px;  } |

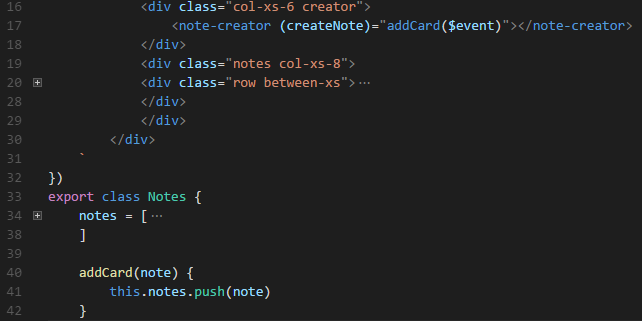
Add the following code to the `NoteCreator` class



Subscribe to the `submit` event on the form:



And in the `notes` container add the necessary functionality



Try to make the form expandable. Hide the title input and the submit button until the value input is focused. Use `\*ngIf` and `(focus)`

## Add option to choose colors

Exercise: add `color-picker` component allowing the use to change the color of the note card

Example template:

|  |
| --- |
| <div class="color-selector">  <i class="material-icons icon">color\_lens</i>  <div class="selector row center-xs">  <div class="color"></div>  </div>  </div> |

Example styles:

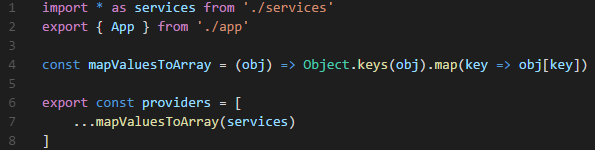
|  |
| --- |
| .color-selector {  position: relative;  }  .selector {  min-width: 120px;  border: 1px solid lightgrey;  padding: 10px;  background-color: #efefef;  position: absolute;  top: -50px;  left: 0;  }  .color {  height: 30px;  width: 30px;  border-radius: 100%;  cursor: pointer;  margin-right: 10px;  margin-bottom: 10px;  }  .color:hover {  border: 2px solid darkgrey;  }  .icon {  font-size: 1.4rem;  color: grey;  cursor: pointer;  } |

## Add HTTP

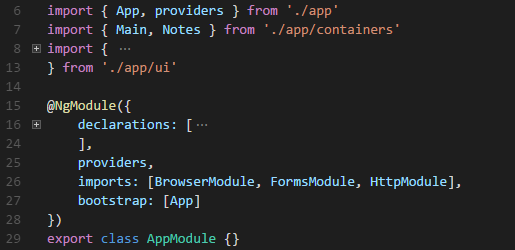
Import `HttpModule` in `main.ts`.

Add `app/services` folder and then add three empty files - `index.ts`, `api.ts` and `notes.ts`.

In `app/index.ts` add the following magic:



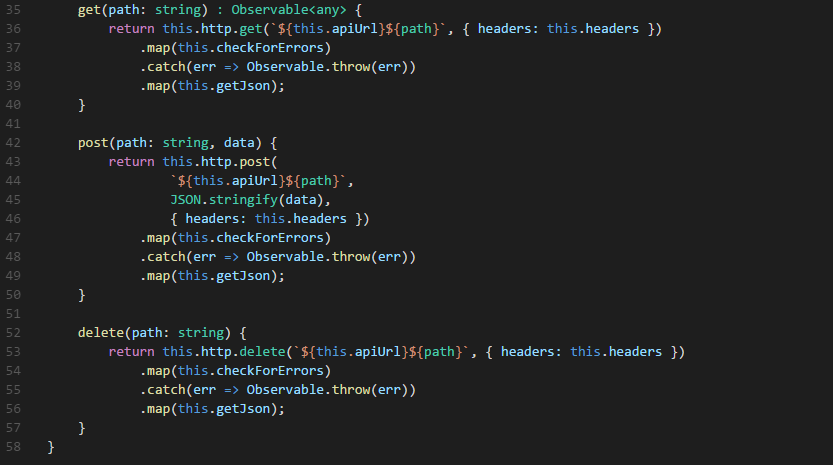
Add the providers in the `main.ts`



## Add the ApiService

In `app/services/api` add the following:





## Add Notes resource

In `app/services/notes` add the following:



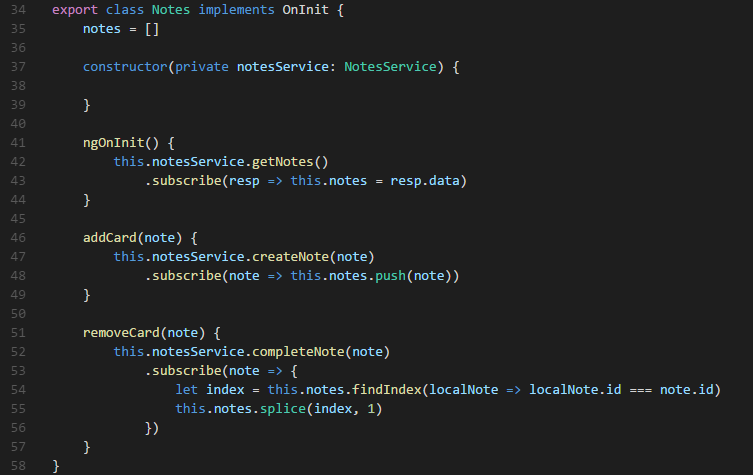
Add both `ApiService` and `NoteServices` in the `index.ts` file.

## Run the API server

Run `npm run api` and a Node.js server should start listening for request on port 3500.

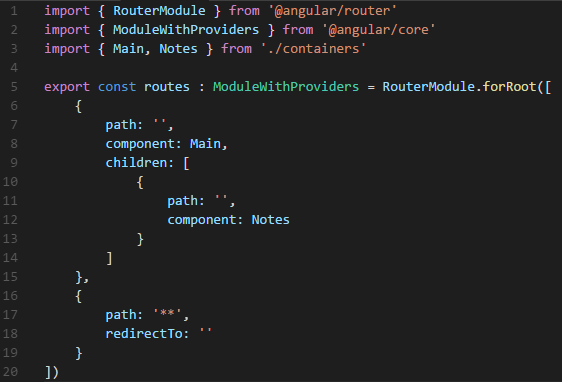
## Use the Notes service in the Notes container

Change the `containers/notes` to:



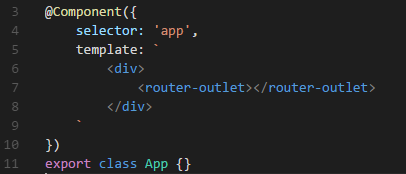
## Add routing

Create `routes.ts` in `app`:



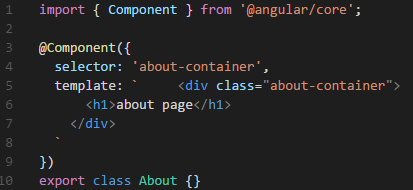
Add it to the `imports`.

Change the `app.ts` and the notes container to have the router outlet.



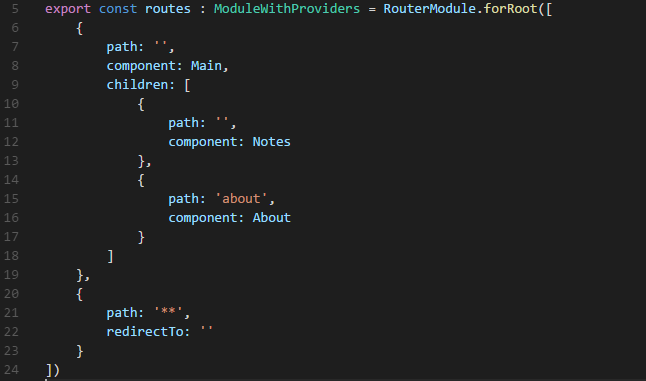
## Add another route and menu

Create `about.ts` in `app/containers`:



Add it to the declarations.

Register the route:



Change the `navbar.ts` in `app/ui`:

