Wednesday:Form Component

Form Component

If we want to add goals to our application at the moment, we would have to do it by hard coding the goals in the goals array. What if we wanted our users to add goals of their liking from the frontend interface that they are interacting with? How would we go about it?

Well, thanks to web forms, we can give the user this power. A web form, also called a HTML form, helps us collect data from users by providing inputs in which users can fill in data. We will create a web form in which a user can fill in the details of a goal, that is, name, description and completion date to enable users to add goals from the user interface.

We will create our form the same way we have been creating other components. Let's create this form component using the angular CLI on our terminal:

```
$ ng generate component goal-form
```

This command has created the folder goal-form and inside it created the files that comprise a component. It has also added our goal-form component to our root module declarations array in the *app.module.ts* file.

Let's now define the structure of our form.

src/app/goal-form/goal-form.component.html

```
<div class="container-fluid">
  <h2 class="text-center">Create a new Goal</h2>
  <hr>
  <form>
      <div class="form-group">

          <label for="name">Name</label>
          <input type="text" required class="form-control" id="name">

      </div>
      <div class="form-group">

          <label for="description">Description</label>
          <textarea class="form-control" id="description" rows="4" required></textarea>

      </div>
      <div class="form-group">
          <label for="complete">Completion</label>
          <input type='date' id="complete" required>

      </div>
  </form>
</div>
```

We have created a form with three inputs, name, description and date which are required to create a goal object. We will now use the selector of the *goal-form* component to nest it inside the goal component.

src/app/goal/goal.component.html

```
<div class="container">
  <h1>MY GOALS</h1>
  <hr>

  <div class="row">
    <div class="col-md-6">
      <div *ngFor='let goal of goals;let i = index'>
        <div>
          <h6 id={{i}} appStrikethrough>{{goal.name}} due on {{goal.completeDate|date|uppercase}}</h6>
          <button (click)='toggleDetails(i)' class="btn btn-primary">Toggle Details</button>
        </div>

        <div *ngIf='goal.showDescription'>
          <app-goal-detail  [goal]='goal' (isComplete) = 'deleteGoal($event,i)'></app-goal-detail>
        </div>
      </div>
    </div>
    <div class="col-md-6">
      <app-goal-form></app-goal-form>
    </div>
  </div>
  <p *ngIf='goals.length > 5'>You have too many goals</p>
</div>
```

We have added the form to the right side of our goals array. This form only has input fields and angular does not know how to interact with it at this point. In angular, whenever we want to make HTML forms interact with the angular app, we need to import the forms module from angular forms. We do this by making this import in our root modules file, *app.modules.ts*.

src/app/app.module.ts

```
....
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
....
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule
  ],
....
```

At the top, we have imported the `FormsModule` and then added it to our `imports` array. We have done this import because the forms module is not available in our angular app by default. Our angular application can now work with any number of forms that we create thanks to the FormsModule.

Did you enjoy this content?

Yes No

Send Feedback