Create a project

First, create a new project named angular-forms by using the following command:

1. ng **new** angular-forms

Now, go to the project folder by using the following command.

1. cd angular-forms

Now, use the Angular CLI command **ng generate class User** to generate a new class named User:

1. ng generate **class** User

Go to your project folder **angular-forms** and open the User.ts file under the app module. Write the following code:

1. export **class** User {
2. constructor(
3. **public** id: number,
4. **public** name: string,
5. **public** power: string,
6. **public** alterEgo?: string
7. ) {  }
8. }

The TypeScript compiler generates a public field for each public constructor parameter and when you create new Useres, it automatically assign the parameter's value to that field.

Here, alterEgo is optional and the constructor can omit it.

Create a Form component

**An Angular form consists of two parts:**

* HTML based template
* A component class to handle data and users

Now, generate a new component named **UserForm** by using the following command:

1. ng generate component UserForm

Write down the following code in **User-form.component.ts**

1. **import** { Component } from '@angular/core';
2. **import** { User }    from '../User';
3. @Component({
4. selector: 'app-User-form',
5. templateUrl: './User-form.component.html',
6. styleUrls: ['./User-form.component.css']
7. })
8. export **class** UserFormComponent {
9. powers = ['Really Smart', 'Super Flexible',
10. 'Super Hot', 'Weather Changer'];
11. model = **new** User(18, 'Dr IQ', **this**.powers[0], 'Chuck Overstreet');
12. submitted = **false**;
13. onSubmit() { **this**.submitted = **true**; }
14. // TODO: Remove this when we're done
15. get diagnostic() { **return** JSON.stringify(**this**.model); }
16. }

Revise app.module.ts file

The app.module.ts file is used to define the application's root module. The template-driven forms reside in their own module. You need to add the FormsModule to the array of imports for the application module before using forms.

Use the following code inside **app.module.ts** file:

1. **import** { NgModule }      from '@angular/core';
2. **import** { BrowserModule } from '@angular/platform-browser';
3. **import** { FormsModule }   from '@angular/forms';
4. **import** { AppComponent }  from './app.component';
5. **import** { UserFormComponent } from './User-form/User-form.component';
6. @NgModule({
7. imports: [
8. BrowserModule,
9. FormsModule
10. ],
11. declarations: [
12. AppComponent,
13. UserFormComponent
14. ],
15. providers: [],
16. bootstrap: [ AppComponent ]
17. })
18. export **class** AppModule { }

Here, we have imported FormsModule and added the FormsModule to the list of imports defined in the @NgModule decorator. This is used to access the application to all of the template-driven forms features, including ngModel.

AD

Revise app.component.html file

The app.component.html is used to host the new UserFormComponent. It is the application's root component. Write the following code in app.component.html

1. <app-User-form></app-User-form>

Create an initial HTML form template

Use the following code in User-form.component.html

1. <div **class**="container">
2. <h1>User Form</h1>
3. <form>
4. <div **class**="form-group">
5. <label **for**="name">Name</label>
6. <input type="text" **class**="form-control" id="name" required>
7. </div>
8. <div **class**="form-group">
9. <label **for**="alterEgo">Alter Ego</label>
10. <input type="text" **class**="form-control" id="alterEgo">
11. </div>
12. <button type="submit" **class**="btn btn-success">Submit</button>
13. </form>
14. </div>

Style the form

Open style.css and use the following code to import the bootstrap file.

1. @import url('https://unpkg.com/bootstrap@3.3.7/dist/css/bootstrap.min.css');

Add power list by using \*ngFor

The User form can choose power lists from a fixed list of agency-approved powers. Use the following HTML code immediately below the Alter Ego group in User-form.component.html

1. <div **class**="form-group">
2. <label **for**="power">User Power</label>
3. <select **class**="form-control" id="power" required>
4. <option \*ngFor="let pow of powers" [value]="pow">{{pow}}</option>
5. </select>
6. </div>

The basic template-driven form is completed now. You can use the ng serve command to run the project.

**Output:**