# Angular Template-driven Form Demo

## 1. Setup Project

### 1.1 Install Dependencies

1. Change directory to calab:

* cd calab

1. Install dependencies by running the following command:

* npm install

## 2. Create a Form Component

### 2.1 Create a New Component

1. Create a new component using CLI and name it OrderForm:

* npx -p @angular/cli ng generate component components/order-form

1. Import OrderForm into src/app/app.component.ts:

* import { OrderFormComponent } from './components/order-form/order-form.component';

1. Inside src/app/app.component.ts update imports to include OrderForm:

* imports: [RouterOutlet, OrderFormComponent],

1. Open src/app/app.component.html template and add the following element after the <div class="divider"...:

* <app-order-form></app-order-form>

### 2.2 Review Changes

1. Start Angular Development Server if not yet started:

* npx -p @angular/cli ng serve
* *Otherwise refresh the browser tab to see updated view.*

1. You should see placeholder for OrderForm component rendered on a screen.

## 3. Setup Template-driven Form in OrderFormComponent

### 3.1 Update Component

1. Open src/app/components/order-form/order-form.component.ts file and do the following:
   * Import FormsModule:
   * import { FormsModule } from '@angular/forms';
   * Update imports to include FormsModule:
   * @Component({  
      ...  
      imports: [FormsModule],  
      ...  
     })

### 3.2 Create The Form Template

1. Open src/app/components/order-form/order-form.component.html file and do the following:
   * Create a form template:
   * <form>  
      <label for="product">Product:</label>  
      <input type="text" id="product" name="product" required>  
      <label for="quantity">Quantity:</label>  
      <input type="text" id="quantity" name="quantity" required>  
      <button type="submit">Submit</button>  
     </form>
2. Open src/app/components/order-form/order-form.component.css file and do the following:
   * Add some style:
   * /\* Style inputs \*/  
     input, select {  
      width: 100%;  
      padding: 12px 20px;  
      margin: 8px 0;  
      display: inline-block;  
      border: 1px solid #ccc;  
      border-radius: 4px;  
      box-sizing: border-box;  
     }  
       
     /\* Style the submit button \*/  
     button[type=submit] {  
      width: 100%;  
      background-color: #04AA6D;  
      color: white;  
      padding: 14px 20px;  
      margin: 8px 0;  
      border: none;  
      border-radius: 4px;  
      cursor: pointer;  
     }  
       
     /\* Add a background color to the submit button on mouse-over \*/  
     button[type=submit]:hover {  
      background-color: #45a049;  
     }  
       
     .ng-valid[required], .ng-valid.required {  
      border-left: 5px solid #42A948; /\* green \*/  
     }  
     .ng-invalid:not(form) {  
      border-left: 5px solid #a94442; /\* red \*/  
     }

### 3.3 Data Bind With ngModel

1. Create a new class inside src/app/components/order-form/ representing an Order Model:

* // order.ts  
  export class Order {  
   constructor(  
   public product: string,  
   public quantity?: number,  
   ) {}  
  }

1. Open src/app/components/order-form/order-form.component.ts file and do the following:
   * Import Order model:
   * import { Order } from './Order';
   * Declare a model that you want to bind to the template:
   * order = new Order('');
   * *default product set to empty string.*
2. Open src/app/components/order-form/order-form.component.html file and do the following:
   * Update a form template with the following:
   * <form>  
      <label for="product">Product:</label>  
      <input type="text" id="product" name="product" required [(ngModel)]="order.product">  
      <label for="quantity">Quantity:</label>  
      <input type="text" id="quantity" name="quantity" required [(ngModel)]="order.quantity">  
      <button type="submit">Submit</button>  
     </form>

### 3.4 Submitting the Form

1. Open src/app/components/order-form/order-form.component.ts file and do the following:
   * Create an onSubmit() callback method, allowing you to process the captured form data as needed:
   * onSubmit(){  
      console.log(this.order);  
     }
2. Open src/app/components/order-form/order-form.component.html file and do the following:
   * Add an ngSubmit event listener to the form tag with the onSubmit() callback method:
   * <form (ngSubmit)="onSubmit()">  
     ...

### 3.5 Review Changes

1. Start Angular Development Server if not yet started:

* npx -p @angular/cli ng serve --host 0.0.0.0
* *Otherwise refresh the browser tab to see updated view.*

1. You should see Form with submit button displayed on your screen.

## 4. Template-driven Form Validation

### 4.1 Show And Hide Validation Error Messages

1. Open src/app/components/order-form/order-form.component.html file and do the following:
   * Add a local reference to the input #product="ngModel":
   * <input type="text" id="product" name="product" required [(ngModel)]="order.product" #product="ngModel">
   * Add a conditional error message to product:
   * <div [hidden]="product.valid || product.pristine">  
      Product name is required  
     </div>

### 4.2 Review Changes

1. Start Angular Development Server if not yet started:

* npx -p @angular/cli ng serve --host 0.0.0.0
* *Otherwise refresh the browser tab to see updated view.*

1. You should see error message displayed if no value is entered in Product field.