Exercise (Instructions): Event & Binding with ngModel in Project blog

Objectives and Outcomes

In this exercise you will get familiar with Event & Binding using ngModel.

At the end of this exercise you will be able to:

* Create a submit form
* Create and bind data with an event
* Submit data to services

Clone the project MyBlogApplication from here:

<https://github.com/DuongPT4/MyBlogApplication.git>

from the downloaded location, open cmd. Install node server:

*npm init*

*npm install lite-server*

to run the application, in the cmd type:

*ng serve --open*

In the app.component.ts, add the <form>:

import { Component } from '@angular/core';

import {PostService} from './services/post.service';

import {Post} from './services/post';

@Component({

  selector: 'app-root',

  template: `<h1>Welcome </h1>

  <form>

      <input type="text" placeholder="Post Title" [(ngModel)]="title" name="title" />

      <br><br>

      <textarea placeholder="Post Body"  [(ngModel)]="body" name="body"> </textarea>

      <br><br>

      <input type="submit" value="Submit"/>

    </form>

  <div class = "container">

    <ul>

      <li \*ngFor="let post of posts">

        <h3> {{post.title}} </h3>

        <p>{{post.body}} </p>

      </li>

    </ul>

  </div>

  `,

  providers:[PostService],

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  posts : Post[];

  title: string;

  body:string;

  constructor(private \_postService : PostService){

    \_postService.getPost().then(

      ps => {this.posts=ps;

      });

  }

}

in the app.module.ts

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';

import{FormsModule} from '@angular/forms';

@NgModule({

  declarations: [

    AppComponent

  ],

  imports: [

    BrowserModule,

    FormsModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

Change the post.service.ts

@Injectable()

export class PostService{

    getPost():Promise<Post[]>{

        return Promise.resolve(POSTS);

    }

    addPost(post:Post){

        POSTS.push(post);

    }

}

Add the function addPost to app.component.ts

@Component({

  selector: 'app-root',

  template: `<h1>Welcome </h1>

  <form (submit)=addPost()>

      <input type="text" placeholder="Post Title" [(ngModel)]="title" name="title" />

      <br><br>

      <textarea placeholder="Post Body"  [(ngModel)]="body" name="body"> </textarea>

      <br><br>

      <input type="submit" value="Submit"/>

    </form>

  <div class = "container">

    <ul>

      <li \*ngFor="let post of posts">

        <h3> {{post.title}} </h3>

        <p>{{post.body}} </p>

      </li>

    </ul>

  </div>

  `,

  providers:[PostService],

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  posts : Post[];

  title: string;

  body:string;

  constructor(private \_postService : PostService){

    \_postService.getPost().then(

      ps => {this.posts=ps;

      });

  }

  addPost(){

   var newPost = {

     id:0,

     title:this.title,

     body:this.body

   }

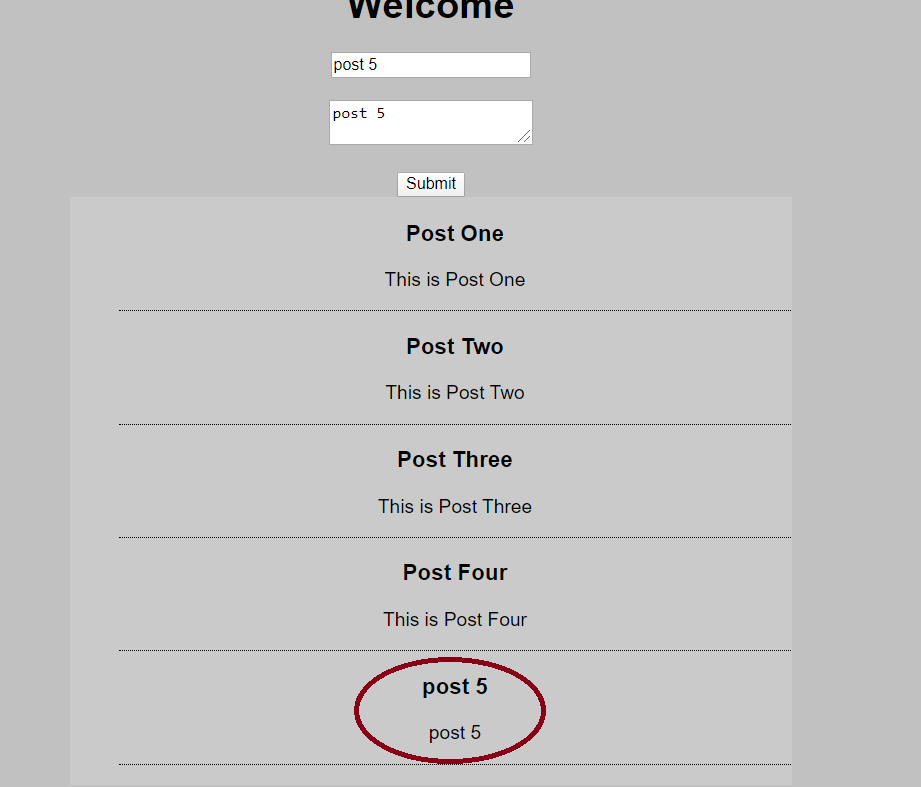
    this.\_postService.addPost(newPost);

    return false;

  }

}

Run the Application:



Conclusions