PROGRAMMING TECHIE

Build a Full Stack Reddit Clone with – Spring boot and Angular – Part 15

0 Comments

BY SAI UPADHYAYULA

April 16, 2020

Reddit clone with Spring boot and Angular
#15
Create Subreddits and Posts in Angular

Share Tweet Pin

Welcome to Full Stack Reddit Clone with Spring boot and Angular – Part 15. In Part 14, we refactored the Home Page Component in the Angular application.

In this article, we will add the functionality to create Subreddit and Posts from our Angular application, we will also make some miscellaneous improvements to our frontend application.

If you are a visual learner like me you can checkout the Video version of this tutorial below:

Download Source Code

> **Source code for Angular application** – https://github.com/SaiUpadhyayula/angular-reddit-clone
> **Source code for Spring boot backend** –https://github.com/SaiUpadhyayula/spring-reddit-clone

Generate Components to Create Subreddits and Posts

The first thing we are going to do is to generate components which hold the functionality to Create Posts and Subreddits. As always, we will be using Angular CLI to generate our components, so open your terminal and type the commands you see in the below image.

In the previous article, we have already created the buttons for **Create Subreddit** and **Create Post** inside the sidebar, so now we have to add a routing logic from those buttons to the components we created just now.

Let's update our `<strong>app-routing.module.ts</strong>` file

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { SignupComponent } from './auth/signup/signup.component';
import { LoginComponent } from './auth/login/login.component';
import { HomeComponent } from './home/home.component';
import { CreatePostComponent } from './post/create-post/create-post.component';
import { CreateSubredditComponent } from './subreddit/create-subreddit/create-subreddit.component';

const routes: Routes = [
  { path: '', component: HomeComponent },
  { path: 'create-post', component: CreatePostComponent },
  { path: 'create-subreddit', component: CreateSubredditComponent },
  { path: 'sign-up', component: SignupComponent },
  { path: 'login', component: LoginComponent }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

In the previous article, we have already added the value for **click** directive the for **Create Post** and **Create Subreddit Buttons** inside the `side-bar.component.html` file as **goToCreatePost()** and **goToCreateSubreddit()**, let’s create the method definitions for them inside the `<strong>side-bar.component.ts</strong>`

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';

@Component({
  selector: 'app-side-bar',
  templateUrl: './side-bar.component.html',
  styleUrls: ['./side-bar.component.css']
})
export class SideBarComponent implements OnInit {

  constructor(private router: Router) { }

  ngOnInit() {
  }

  goToCreatePost() {
    this.router.navigateByUrl('/create-post');
  }

  goToCreateSubreddit() {
    this.router.navigateByUrl('/create-subreddit');
  }

}
```

Create Subreddit

Let’s implement the logic to create subreddits from our Angular application. Copy the below HTML and CSS code and paste it into the corresponding files in your project.

`create-subreddit.component.html`

```
<div class="container">
  <div class="row">
    <div class="create-subreddit-container">
      <form class="post-form" [formGroup]="createSubredditForm" (ngSubmit)="createSubreddit()">
        <div class="form-group">
          <div class="create-subreddit-heading">Create Subreddit</div>
          <hr />
          <input type="text" [formControlName]="'title'" class="form-control" sty
```

```
            placeholder="Title">
          <textarea type="text" [formControlName]="'description'" style="width: 100%; margin-top: 5px"
            placeholder="Description"></textarea>
          <div>
            <div style="margin-top: 5px" class="float-right">
              <button (click)="discard()" class="btnDiscard">Discard</button>
              <button (click)="createSubreddit()" class="btnCreateSubreddit">Create</button>
            </div>
          </div>
        </div>
      </form>
    </div>
    <div class="col-md-3">
      <div class="sidebar">
        <h5 class="guidelines">Posting to Spring Reddit</h5>
        <hr />
        <ul>
          <li>Remember the human</li>
          <hr />
          <li>Behave like you would in real life</li>
          <hr />
          <li>Don't spam</li>
          <hr />
        </ul>
      </div>
    </div>
  </div>
</div>
```

create-subreddit.component.css

```
.create-subreddit-container {
  margin-top: 10px;
  --post-line-color: #ccc;
  border: 1px solid #ccc;
  margin-top: 10px;
  margin-bottom: 10px;
  overflow: hidden;
  background-color: rgba(255, 255, 255, 0.8);
  color: #878A8C;
  position: relative;
  border-radius: 4px;
  padding: 5px;
}

.create-subreddit-heading {
  font-size: 18px;
  font-weight: 500;
  line-height: 22px;
  color: #1c1c1c;
  -ms-flex: 1;
  flex: 1;
}

.btnCreateSubreddit {
  background-color: #0079D3;
  border-color: #0079D3;
  color: aliceblue;
  fill: #0079D3;
  border: 1px solid;
  border-radius: 4px;
  text-align: center;
  letter-spacing: 1px;
  text-decoration: none;
  font-size: 12px;
  font-weight: 700;
  letter-spacing: .5px;
  line-height: 24px;
  text-transform: uppercase;
  padding: 3px 16px;
  opacity: 1;
  width: 288px;
  height: 34px;
}

.btnDiscard {
  fill: #0079D3;
  border: 1px solid;
  border-radius: 4px;
  text-align: center;
  letter-spacing: 1px;
  text-decoration: none;
  font-size: 12px;
  font-weight: 700;
  letter-spacing: .5px;
  line-height: 24px;
  text-transform: uppercase;
  padding: 3px 16px;
  opacity: 1;
  width: 288px;
  height: 34px;
  margin-top: 5px;
  color: #0079D3;
  background-color: transparent;
}

.guidelines {
  text-align: center;
  font-size: 16px;
  font-weight: 500;
  line-height: 20px;
  color: #1c1c1c;
}

.sidebar {
  width: 312px;
  height: 242px;
  --post-line-color: #ccc;
  border: 1px solid #ccc;
  margin-bottom: 10px;
  overflow: hidden;
  background-color: rgba(255, 255, 255, 0.8);
  color: #878A8C;
  position: relative;
  border-radius: 4px;
  padding: 5px;
  margin-top: 10px;
  margin-bottom: 10px;
}
```

This is how the Create Subreddit should look like once you copy-pasted the above code.

Now let’s add some dynamic functionality to this static page, copy the below code into the `create-subreddit.component.ts` file

`<strong>create-subreddit.component.ts</strong>`

```
import { Component, OnInit } from '@angular/core';
import { FormGroup, FormControl, Validators } from '@angular/forms';
import { SubredditModel } from '../subreddit-response';
import { Router } from '@angular/router';
import { SubredditService } from '../subreddit.service';

@Component({
  selector: 'app-create-subreddit',
  templateUrl: './create-subreddit.component.html',
  styleUrls: ['./create-subreddit.component.css']
})
export class CreateSubredditComponent implements OnInit {
  createSubredditForm: FormGroup;
  subredditModel: SubredditModel;
  title = new FormControl('');
  description = new FormControl('');

  constructor(private router: Router, private subredditService: SubredditService) {
    this.createSubredditForm = new FormGroup({
      title: new FormControl('', Validators.required),
      description: new FormControl('', Validators.required)
    });
    this.subredditModel = {
      name: '',
      description: ''
    }
  }

  ngOnInit() {
  }

  discard() {
    this.router.navigateByUrl('/');
  }

  createSubreddit() {
    this.subredditModel.name = this.createSubredditForm.get('title').value;
    this.subredditModel.description = this.createSubredditForm.get('description')
    this.subredditService.createSubreddit(this.subredditModel).subscribe(data => {
      this.router.navigateByUrl('/list-subreddits');
    }, error => {
      console.log('Error occurred');
    })
  }
}
```

Let's see what's going on inside this component

- First things first, we declared a `FormGroup` with variable name `createSubredditForm`, along with the fields `title` and `description` and we are initializing this `FormGroup` inside the constructor.
- We also declared and initialized the `SubredditModel` we created in the last article.
- We have the `createSubreddit()` method which reads the `FormControl` values for fields `title` and `description` and assigning it to the `SubredditModel` object.
- Next, we are calling the `createSubreddit()` method inside the SubredditService and subscribing to the response, once we receive a success response we are navigating to the route /list-subreddits

Now let's implement the `createSubreddit()` method inside the `SubredditService`, for this let's refer to our REST API documentation we created in Full Stack Reddit Clone with Spring Boot and Angular – Part 10.

So the createSubreddit REST API call takes the `SubredditModel` object, let's go ahead and implement it inside the `SubreditService`.

`<strong>subreddit-service.ts</strong>`

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { SubredditModel } from './subreddit-response';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class SubredditService {
  constructor(private http: HttpClient) { }
  getAllSubreddits(): Observable<Array<SubredditModel>> {
    return this.http.get<Array<SubredditModel>>('http://localhost:8080/api/subreddit');
  }

  createSubreddit(subredditModel: SubredditModel): Observable<SubredditModel> {
    return this.http.post<SubredditModel>('http://localhost:8080/api/subreddit',
      subredditModel);
  }
}
```

Show all Subreddits

Let's create another component to display all the subreddits we have in our application. As said above, we will be navigating to this page when the user created a new subreddit, or through the **View All Subreddits** option inside the Subreddit Sidebar section.

Create the components by typing the below commands you see in the image.

Now copy the below HTML, CSS and TS code into the respective files in the project.

`list-subreddits.component.html`

```
<div class="container">
  <div class="row">
    <hr />
    <div class="col-md-9">
      <h2>List of Subreddits</h2>
      <ul>
        <li *ngFor="let subreddit of subreddits"><a routerLink="/view-subreddit/{{subreddit.id}}">
{{subreddit.name}}</a>
        </li>
      </ul>
    </div>
    <div class="col-md-3">
      <app-sidebar></app-sidebar>
    </div>
  </div>
</div>
```

`list-subreddits.component.ts`

```
import { Component, OnInit } from '@angular/core';
import { SubredditModel } from '../subreddit-response';
import { SubredditService } from '../subreddit.service';
import { throwError } from 'rxjs';

@Component({
  selector: 'app-list-subreddits',
  templateUrl: './list-subreddits.component.html',
  styleUrls: ['./list-subreddits.component.css']
})
export class ListSubredditsComponent implements OnInit {

  subreddits: Array<SubredditModel>;
  constructor(private subredditService: SubredditService) { }

  ngOnInit() {
    this.subredditService.getAllSubreddits().subscribe(data => {
      this.subreddits = data;
    }, error => {
      throwError(error);
    })
  }
}
```

- This should be easy to understand, we are now calling the `getAllSubreddits()` method inside the `SubredditService` inside the `ngOnInit()` of the component, subscribing to the response and assigning it to the subreddits object which is of type `Array<SubredditModel>`
- If we receive an error response, we throw an error using the `throwError(error)` method of `rxjs`

Lastly, let's register the route for **/list-subreddits** to the `ListSubredditsComponent` inside the `app-routing.module.ts`

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { SignupComponent } from './auth/signup/signup.component';
import { LoginComponent } from './auth/login/login.component';
import { HomeComponent } from './home/home.component';
import { CreatePostComponent } from './post/create-post/create-post.component';
import { CreateSubredditComponent } from './subreddit/create-subreddit/create-subreddit.component';
import { ListSubredditsComponent } from './subreddit/list-subreddits/list-subreddits.component';

const routes: Routes = [
  { path: '', component: HomeComponent },
  { path: 'list-subreddits', component: ListSubredditsComponent },
  { path: 'create-post', component: CreatePostComponent },
  { path: 'create-subreddit', component: CreateSubredditComponent },
  { path: 'sign-up', component: SignupComponent },
  { path: 'login', component: LoginComponent }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Now let's change the routerLink value for the View All button inside the Subreddit View Side Bar Section.

`subreddit-side-bar.component.html`

```
<div class="sidebar-view-subreddit">
    <div style="color: black; font-weight: bold">Browse Subreddits</div>
    <hr />
    <span *ngFor="let subreddit of subreddits">
        <span class="subreddit-text"><a href="/view-subreddit/{{subreddit.id}}">{{subreddit.name}}</a>
</span>
        <hr />
    </span>
    <div style="text-align: center" *ngIf="displayViewAll">
```

```
        <a style="font-weight: bold" routerLink="/list-subreddits">View All</a>
    </div>
</div>
```

This is how the List Subreddits Page should look like once you try to create a subredit.

Create Posts

Alright, we are halfway through, our next task is to implement the functionality to create posts and view single post. We have already created the components and the supporting source files in the first section of this post, so now let's dive ahead into coding.

Wait before that we need to install some dependencies to our application, to create a Post we need some kind of editor a cool WYSIWYG Editor like TinyMCE.

We can integrate TinyMCE into our Angular project by typing the below command.

```
$ npm install --save @tinymce/tinymce-angular
```

This should install and add the `tinymce-angular` dependency in the `package.json` file

Next enable TinyMCE in our project by adding the EditorModule to the app-module.ts file

`app-module.ts`

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

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { HeaderComponent } from './header/header.component';
import { SignupComponent } from './auth/signup/signup.component';
import { ReactiveFormsModule } from '@angular/forms';
import { HttpClientModule, HTTP_INTERCEPTORS } from '@angular/common/http';
import { LoginComponent } from './auth/login/login.component';
import { NgxWebstorageModule } from 'ngx-webstorage';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { ToastrModule } from 'ngx-toastr';
import { TokenInterceptor } from './token-interceptor';
import { HomeComponent } from './home/home.component';
import { FontAwesomeModule } from '@fortawesome/angular-fontawesome';
import { PostTileComponent } from './shared/post-tile/post-tile.component';
import { VoteButtonComponent } from './shared/vote-button/vote-button.component';
import { SideBarComponent } from './shared/side-bar/side-bar.component';
import { SubredditSideBarComponent } from './shared/subreddit-side-bar/subreddit-side-bar.component';
import { CreateSubredditComponent } from './subreddit/create-subreddit/create-subreddit.component';
import { CreatePostComponent } from './post/create-post/create-post.component';
import { ListSubredditsComponent } from './subreddit/list-subreddits/list-subreddits.component';
import { EditorModule } from '@tinymce/tinymce-angular';


@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    SignupComponent,
    LoginComponent,
    HomeComponent,
    PostTileComponent,
    VoteButtonComponent,
    SideBarComponent,
    SubredditSideBarComponent,
    CreateSubredditComponent,
    CreatePostComponent,
    ListSubredditsComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    ReactiveFormsModule,
    HttpClientModule,
    NgxWebstorageModule.forRoot(),
    BrowserAnimationsModule,
    ToastrModule.forRoot(),
    FontAwesomeModule,
    EditorModule
  ],
  providers: [
    {
      provide: HTTP_INTERCEPTORS,
      useClass: TokenInterceptor,
      multi: true
    }
  ],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Alright, so before diving into the code let's refer our Swagger REST API documentation one more time to see what is the request payload we need to Create a Post.

Let’s create a class that holds all the required fields for our Request Payload to Create Post. Let’s call this class as `CreatePostPayload` and create it inside the **create-post** package

`create-post-payload.ts`

```
export class CreatePostPayload {
    postName: string;
    subredditName?: string;
    url?: string;
    description: string;
}
```

Now let’s copy the below HTML and CSS and TS files into your project.

`create-post.component.html`

```
<div class="container">
  <div class="row">
    <hr />
    <div class="create-post-container col-md-9">
      <form class="post-form" [formGroup]="createPostForm" (ngSubmit)="createPost()">
        <div class="form-group">
          <div class="create-post-heading">Create Post</div>
          <hr />
          <input type="text" [formControlName]="'postName'" class="form-control" style="margin-top: 5px"
            placeholder="Title">

          <input type="text" class="form-control" [formControlName]="'url'" style="margin-top: 5px"
placeholder="URL">

          <select class="form-control" style="margin-top: 10px" [formControlName]="'subredditName'">
            <option value="" selected disabled>Select Subreddit</option>
            <option *ngFor="let subreddit of subreddits">{{subreddit.name}}</option>
          </select>

          <editor [formControlName]="'description'" [init]="{
                      height: 500,
                      menubar: false,
                      plugins: [
                        'advlist autolink lists link image charmap print preview anchor',
                        'searchreplace visualblocks code fullscreen',
                        'insertdatetime media table paste code help wordcount'
                      ],
                      toolbar:
                        'undo redo | formatselect | bold italic backcolor | \
                        alignleft aligncenter alignright alignjustify | \
                        bullist numlist outdent indent | removeformat | help'
                    }"></editor>
          <span>
            <div style="margin-top: 5px" class="float-right">
              <button (click)="discardPost()" class="btnDiscard">Discard</button>
              <button (click)="createPost()"
class="btnCreatePost">Post</button>
            </div>
          </span>
        </div>
      </form>
    </div>
    <div class="col-md-3">
      <guidelines></guidelines>
      <about></about>
    </div>
  </div>
</div>
```

`create-post.component.css`

```
.btnCreatePost,
.btnCreateSubreddit {
  margin-top: 5px;
}

.post-form {
  margin: 2px;
}

.comment-notification {
  margin-top: 5px;
}

.create-post-container {
  margin-top: 10px;
  --post-line-color: #ccc;
  border: 1px solid #ccc;
  margin-top: 10px;
  margin-bottom: 10px;
  overflow: hidden;
  background-color: rgba(255, 255, 255, 0.8);
  color: #878A8C;
  position: relative;
  border-radius: 4px;
  padding: 5px;
}

.btnDiscard {
    fill: #0079D3;
    border: 1px solid;
    border-radius: 4px;
    text-align: center;
    letter-spacing: 1px;
    text-decoration: none;
    font-size: 12px;
    font-weight: 700;
    letter-spacing: .5px;
    line-height: 24px;
    text-transform: uppercase;
    padding: 3px 16px;
    opacity: 1;
    width: 288px;
    height: 34px;
    margin-top: 5px;
    color: #0079D3;
    background-color: transparent;
  }

.create-post-heading {
  font-size: 18px;
  font-weight: 500;
  line-height: 22px;
  color: #1c1c1c;
  -ms-flex: 1;
  flex: 1;
}

.btnCreatePost {
    background-color: #0079D3;
    border-color: #0079D3;
    color: aliceblue;
    fill: #0079D3;
    border: 1px solid;
    border-radius: 4px;
    text-align: center;
    letter-spacing: 1px;
    text-decoration: none;
    font-size: 12px;
    font-weight: 700;
    letter-spacing: .5px;
    line-height: 24px;
    text-transform: uppercase;
    padding: 3px 16px;
    opacity: 1;
    width: 288px;
    height: 34px;
  }
```

create-post.component.ts

```
import { Component, OnInit } from '@angular/core';
import { FormGroup, Validators, FormControl } from '@angular/forms';
import { SubredditModel } from 'src/app/subreddit/subreddit-response';
import { Router } from '@angular/router';
import { PostService } from 'src/app/shared/post.service';
import { SubredditService } from 'src/app/subreddit/subreddit.service';
import { CreatePostPayload } from './create-post-payload';
import { throwError } from 'rxjs';

@Component({
  selector: 'app-create-post',
  templateUrl: './create-post.component.html',
  styleUrls: ['./create-post.component.css']
})
export class CreatePostComponent implements OnInit {

  createPostForm: FormGroup;
  postPayload: CreatePostPayload;
  subreddits: Array<SubredditModel>;

  constructor(private router: Router, private postService: PostService,
    private subredditService: SubredditService) {
    this.postPayload = {
      postName: '',
      url: '',
      description: '',
      subredditName: ''
    }
  }

  ngOnInit() {
    this.createPostForm = new FormGroup({
      postName: new FormControl('', Validators.required),
      subredditName: new FormControl('', Validators.required),
      url: new FormControl('', Validators.required),
      description: new FormControl('', Validators.required),
    });
    this.subredditService.getAllSubreddits().subscribe((data) => {
      this.subreddits = data;
    }, error => {
      throwError(error);
    });
  }

  createPost() {
    this.postPayload.postName = this.createPostForm.get('postName').value;
    this.postPayload.subredditName = this.createPostForm.get('subredditName').value;
    this.postPayload.url = this.createPostForm.get('url').value;
    this.postPayload.description = this.createPostForm.get('description').value;

    this.postService.createPost(this.postPayload).subscribe((data) => {
      this.router.navigateByUrl('/');
    }, error => {
      throwError(error);
    })
  }

  discardPost() {
    this.router.navigateByUrl('/');
  }

}
```

Okey that's a lot of code thrown at your face, let's step back and see what we have inside the `create-post.component.ts` file

- We declared and initialized the `createPostForm` variable of type `FormGroup` inside the `ngOnInit()`
- Inside the `FormGroup` declaration we defined all the fields which our Form has and we also added `Validators` to this `FormControl`, which just a basic validation whether the provided text is not empty.
- Next, we are reading all the Subreddit information as we have to display them in the dropdown when creating the post, after reading them from the `SubredditService`, we are assigning the response to a subreddits variable.
- Next, we have the `createPost()` method which first reads the `FormControl` values and creates the `CreatePostPayload` object.
- Once we have the necessary data, we call the `createPost()` method inside the `SubredditService`, we subscribe to the response, and once we receive a success response we navigate to the home page, or else we throw an error.
- Lastly, we have a `discardPost()` method which re-directs us also to the home page.

And inside the `create-post.component.html` file we have added just one tag called editor which displays a beautiful Editor inside the page, let's see that in action in the next section.

Testing Time

Now let's test our implementations, open up the application at http://localhost:4200 and c be navigated to the List Subreddits Page where you can see all the subreddits created in our application.

Next Click on the Create Post button on the sidebar section of the Home Page and provide the necessary details. The cool TinyMCE Editor we installed looks something like below.

Once you click on Post button, we should be navigated to the home page, and we should be able to see the post we created. But if you tried to experiment with the cool Editor features, you may observe that it just displays the HTML code inside the Post Body like below:

For that we need to make a small change inside the app-tile component, instead of adding the interpolation to display the post body, we have to use the innerHtml directive to display the HTML code as normal text.

`app-tile.component.html`

```
            <!-- Display Posts-->
            <div class="row post" *ngFor="let post of posts$">
              <app-vote-button [post]="post"></app-vote-button>
              <!-- Section to Display Post Information-->
              <div class="col-md-11">
                <span class="subreddit-info">
                  <span class="subreddit-text"><a class="posturl" routerLink="">{{post.subredditName}}</a>
    </span>
                  <span> . Posted by <a class="username" routerLink="/user/{{post.use
    </a></span>
                  <span> . {{post.duration}}</span>
                </span>
                <hr />
                <div class="post-title">
                  <a class="postname" href="{{post.url}}">{{post.postName}}</a>
                </div>
                <div>
                  <p class="post-text" [innerHtml]="post.description"></p>
                </div>
                <hr />
                <span>
                  <a class="btnCommments" role="button">
                    <fa-icon [icon]="faComments"></fa-icon>
                    Comments({{post.commentCount}})
                  </a>
                  <button class="login">
                    Read Post
                  </button>
                </span>
              </div>
            </div>
```

Now you should see the text like below:

Conclusion

So we came to the end of this article, **Full Stack Reddit Clone with Spring boot and Angular – Part 15,** where we saw how to create Posts and Subreddits. Also, we are slowly reaching the end of the Full Stack Reddit Clone with Spring boot and Angular series, there is just one article left where we will see how to post comments, submit votes and Logout from our application. I will see you in the next article, until then Happy Coding Techies 🙂

Icons made by Freepik from www.flaticon.com

About the author

Sai Upadhyayula