Zer0Sense

2º Projeto

Tecnologias e Programação Web

Helder Troca Zagalo
2021/2022

Alexandre Serras, 97505
Gonçalo Leal, 98008
Ricardo Rodriguez, 98388

Índice

Introdução

Para o segundo projeto da disciplina de Tecnologias e Programação Web (TPW), foi-nos proposto o desenvolvimento de uma aplicação web, em Angular, juntamente com uma API RESTful, em Django.

Ao longo deste relatório, explicaremos no que consiste o projeto, a sua implementação e como é que as diferentes estruturas do projeto estão interligadas.

Definição da aplicação

A aplicação **Zer0Sense** permite que utilizadores leiam publicações feitas por autores. Desta forma, existem vários tipos de entidades que fazem parte do modelo de negócio:

Existem três estados de **publicações**:

- **Por aprovar:** A publicação está sujeita a uma aceitação por parte de um Gestor ou Admin.
- **Aprovado:** A publicação foi aceite por um Gestor ou Admin, estando disponível para leitura por qualquer utilizador.
- **Arquivado:** A publicação foi arquivada por um Gestor ou Admin, guardando toda a sua informação mas deixando de estar disponível na página de publicações.

Publicações:

- Possuem um autor, uma data de criação, o tópico da publicação e o seu conteúdo. Pode ter comentários associados ao mesmo.

Tópico de publicação:

- Tem como objetivo descrever a categoria da publicação. Podem ser adicionados ou alterados.

Favoritos:

- Um favorito a uma publicação pode ser adicionado ou removido por um utilizador registado.

Tipos de utilizadores (grupos):

- **(Anónimo):** Visitante da aplicação. Não está registado mas pode ver publicações.
- **Leitor:** Utilizador registado. Só pode ver publicações e adicionar/remover uma publicação aos favoritos.

- **Autor:** Utilizador registado com permissões para criar publicações e ver todo o tipo de publicações que submeteu.
- **Gestor:** Utilizador registado. Tem todo o tipo de permissões, excluindo a possibilidade de manipular um Admin.
- **Admin:** Utilizador registado. Tem todo o tipo de permissões. Serve como supervisor de todos os gestores.

Comentários:

- Um utilizador registado pode adicionar e remover comentários numa determinada publicação.

Funcionalidades

É importante mencionar as páginas que fazem parte da aplicação e, da mesma forma, dar uma visão geral sobre as suas funcionalidades e quem pode, ou não, ter acesso às mesmas.

- **Publications:** Lista todo o tipo de publicações do tipo ‘Aprovado’. Acedido por todos.

- **My Publications:** Lista todas as publicações (de todos os tipos) realizadas pelo utilizador. Permite também navegar para a página ‘Add Publication’, que permite criar uma publicação com formatação. Pode ser acedido por utilizadores com permissões iguais ou superiores a Autor.

- **Favourites:** Lista todas as publicações preferidas pelo utilizador, ou seja, aquelas em que ele marcou como favorito. Acedido por todos os utilizadores registados.

- **Pendent Publications:** Lista todas as publicações do tipo ‘Por Aprovar’. Só pode ser acedido por um Gestor ou Admin.

- **Closed Publications:** Lista todas as publicações do tipo ‘Arquivado’. Só pode ser acedido por um Gestor ou Admin.

- **Manage Users:** Lista todos os utilizadores da aplicação com permissões inferiores, ou iguais, ao do utilizador atual. Permite manipular o grupo dos utilizadores. Só pode ser acedido por um Gestor ou Admin.

- **Manage Topics:** Lista todos os tópicos de publicação existentes e permite uma nova adição de tópico. Podemos alterar a descrição de um tópico, desativar (deixa de ser permitido adicionar uma publicação com esse tópico) ou voltar a ativar o tópico. Só pode ser acedido por um Gestor ou Admin.

- **Publication:** Página específica de uma publicação. Apresenta o autor, data de criação, tópico e o conteúdo da publicação, formatado em HTML. Permite a adição/remoção de favoritos por parte de um utilizador, aceitação/arquivação por parte de um Gestor/Admin e apresenta, também, uma seção de comentários. Só pode ser acedido por um Gestor ou Admin.

- **Search:** É possível filtrar publicações por título, autor, tópico e data de criação nas páginas que listam publicações. Da mesma forma, os gestores e administradores podem filtrar utilizadores pelo seu username, nome completo e por grupo na página *Manage Users*.

- **Autenticação**: Quando um utilizador regista-se na plataforma, um novo *token*, próprio do *Django REST framework*, é criado e associado a esse utilizador para futuras autenticações. Deste modo, quando um utilizador está logado e realiza operações que necessitem de autenticação, o *token*, guardado no *frontend*, é disponibilizado para garantir a autenticidade da pessoa nos pedidos *HTTP*, devendo o token ser enviado nas respetivas operações que esse utilizador faça para do lado do backend ser possivel verificar que o utilizador que esta a aceder aos dados é quem diz que é.

- **Autorização:** Um utilizador logado está limitado ao nível de operações permitidas consoante o seu grupo (leitor, autor, gestor, administrador). Assim, várias funcionalidades são só permitidas a um grupo restrito de utilizadores consoante o seu grau hierárquico, o que foi apresentado anteriormente na definição de cada página.

A aplicação é single page e algumas destas funcionalidades são incorporados nesta página através do app-root, cada uma destas funcionalidades é constituída por diversos componentes reutilizados entre elas.

A componente publication é utilizado em várias funcionalidades, como em ,**Publications , My Publication , Pendent Publications ,Closed Publications.**
Outra componente bastante utilizado foi o search , onde foi preciso utilizar noções mais aprofundadas de angular , como o @Output, e com isto permitia-nos mostrar apenas as publicações que correspondiam aos resultados do search,

API REST

Ao aceder a página do backend, presente no final deste relatório, podemos ter uma listagem de todos os endpoints da API necessários para o projeto.

Entre estes, temos endpoints que permitem a criação (POST), verificação (GET), atualização (PUT) e a remoção (DELETE) das diferentes entidades, fundamentais para o funcionamento do sistema.

A descrição de cada endpoint da API está presente neste no ficheiro DocumentaçãoApi.pdf, presente na pasta do projeto

Execução do sistema

Para executar o sistema, é necessário executar tanto a aplicação web como o *Django REST framework*.

Localmente

Para correr localmente, é necessário correr o Django na porta 7007 e, para o Angular, é necessário instalar as bibliotecas adicionais:

- npm i @kolkov/angular-editor (Angular Editor)

No angular fazer ng serve e fica a correr na porta 4200

Deployment

Para o *deployment* do projeto, foi usado o Netlify para suportar o frontend e o PythonAnywhere para hospedar a API RESTful.

Link do *frontend*: https://pedantic-tereshkova-7a6fb9.netlify.app/

Link do *backend*: https://zer0sense2.pythonanywhere.com/

Testagem com utilizadores

Para testar os diferentes tipos de utilizadores, criámos quatro utilizadores diferentes:

Admin:
- Username: serroso
- Password: talisca123

Gestor:
- Username: reis
- Password: talisca123
- Username:pedrofilo
- Password:angular

Autor:
- Username: leal
- Password: talisca123
- Username: joazinho
- Password:vitinha

Leitor:
- Username: bernas
- Password: talisca123
- Username:bulastro
- Password:marega123