

Relatório Projeto

Angular Framework Django Rest Framework

TECNOLOGIAS PROGRAMAÇÃO WEB

Autores:

Diogo Filipe Amaral Carvalho
Rafael Ferreira Baptista

Nº Mec: 92969 Participação: 1/2
Nº Mec: 93367 Participação: 1/2

Curso: Licenciatura em Engenharia Informática



universidade
de aveiro

Departamento de Eletrónica,
Telecomunicações e Informática

1.Introdução

Foi nos proposto o desenvolvimento de uma aplicação web de tema livre através da utilização das tecnologias Django Rest Framework e Angular Framework.

2. Conceito & Objetivos

BeautyCare é uma aplicação web focada no mercado de negócios do mundo da beleza. O principal objetivo desta aplicação é oferecer aos donos de institutos de beleza uma montra de publicidade para os seus institutos, bem como para os produtos e serviços que estão a vender aos seus clientes.

3. Estrutura e Listagem Funcionalidades

A aplicação pode ser dividida em 3 partes distintas:

- Parte Pública (Sem Login). Funcionalidades:
 - Visualização de instituto/servico/produto em destaque na homepage
 - Procura com possibilidade de filtragem simples ou por combinação dos campos "Nome" e "Localização" de institutos.
 - Visualização da página de um instituto, onde contem informação do mesmo, contactos, conjunto serviços oferecidos, conjunto produtos em venda e membros do staff.
 - Procura com possibilidade de filtragem simples ou por combinação dos campos "Nome", "Instituto", "CategoriaServiço", "Preço Mínimo" e "Preço Máximo" de serviços.
 - Visualização da página de um serviço, onde tem informação detalhada sobre o mesmo
 - Procura com possibilidade de filtragem simples ou por combinação dos campos "Nome", "Instituto", "CategoriaProduto", "Preço Mínimo" e "Preço Máximo" de produtos.
 - Visualização da página de um produto, onde tem informação detalhada sobre o mesmo.
 - Efetuar Login numa conta
 - Registar uma nova conta

- Parte Semi-Privada (Login Dono de Instituto). Funcionalidades:
 - Todas as funcionalidades descritas no ponto 2.1, à exceção das 2 últimas, estão aqui também disponíveis
 - Acesso à área de gestão pessoal dos seus institutos, produtos, serviços e staff.
 - Inserção de um novo instituto, produto, serviço ou membro de staff
 - Edição de um instituto, produto, serviço ou membro do staff da sua posse
 - Remoção de um instituto, produto, serviço ou membro do staff da sua posse
 - Alteração de informação pessoal do utilizador na sua área pessoal. (Email e Password)
 - Efetuar logout
- Parte Privada (Login Administrador). Funcionalidades:
 - Todas as funcionalidades descritas no ponto 2.1, à exceção das 2 últimas, e no ponto 2.2 estão aqui também disponíveis
 - Acesso à área reservada para administrador, onde pode efetuar a gestão das categorias de serviços e produtos, permitindo inserir uma nova categoria ou remover categorias existentes.
 - Acesso total a todos os institutos, produtos, serviços e membros de staff na área de Gestão, podendo editar e remover qualquer um destes.

4. Funcionalidades Extra

- Serviços e Produtos expostos encontram-se ordenados por preço visto ser natural essa ordenação tendo em conta o intuito da webpage.
- Mecanismos de autorização implementados. Para além dos botões que permitem acesso às áreas semi-privadas e privadas estarem apenas disponíveis na situação específica (i.e, quando o login esta efetuado), também existe controlo caso exista tentativa de entrada direta através do URL nestas áreas. O mesmo se aplica aos URL's atribuídos para edição e remoção de entidades (institutos,serviços,produtos,staff) em específico.
- Paginação nas páginas de pesquisa de serviços e produtos na parte pública da aplicação

5. Autenticação

O processo de autenticação foi implementado com JWT (Json Web Tokens). No processo de autenticação, o utilizador envia um pedido HTTP à API com as suas credenciais. Consoante as credenciais, a API envia de volta uma mensagem com os tokens de acesso e refresh que são guardados na localstorage do cliente.

Sempre que o cliente efetuar um novo pedido, através da implementação de um intercetor, o angular interceta os pedidos e quando necessário, cria um clone do mesmo com a adição do token de acesso aos headers do pedido, prosseguindo com o pedido clone.

Visto que o token de acesso tem um tempo de vida curto por motivos de segurança, quando o mesmo expira, o django vai retornar um erro com o código 401 (Não Autorizado). Quando o angular recebe este erro, envia automaticamente um novo pedido com o token de refresh para atualizar o seu token de acesso.

Por fim, no processo de logout, a localstorage do cliente é limpa.

6. Credenciais Contas

6.1 Login Dono Instituto:

username: DiogoCarvalho

password: dfacpass

6.2 Login Administrador:

username: projeto

password: projeto

7. Notas

Na modelação dos dados, decidimos que os serviços, produtos e staff estariam dependentes de um utilizador invés de dependentes de um instituto, visto que assim permitimos que um mesmo produto/serviço/membrostaff seja reutilizado para diferentes institutos de um mesmo dono.

8. Deploy

Conseguimos efetuar com sucesso o deploy separado das componentes do projeto. No caso da aplicação django deployed no pythonanywhere, encontra-se não só a Django Rest API como também o projeto desenvolvido no primeiro trabalho prático. Seguem-se os links de ambas as componentes.

<https://rafaelfbaptista.pythonanywhere.com/>

<https://projeto-tpw.herokuapp.com/home>

9. Conclusão

Este projeto foi bastante útil no desenvolvimento das capacidades pessoais dos elementos do grupo de trabalho no trabalho com as tecnologias web Angular Framework e Django Rest Framework.

Visto que conseguimos aproveitar os conhecimentos adquiridos com esta cadeira para efetuar a componente web de um sistema para um projeto de uma outra cadeira neste mesmo curso, consideramos que adquirimos os conhecimentos necessários com sucesso.