

Introdução

Este teste é composto pela construção de uma aplicação SPA, composta por um backend em Spring Boot e um frontend em Angular, tendo como funcionalidade uma tela de cadastro de países.

Ambos são descritos nas próximas páginas.

Tecnologia

O frontend deve ser desenvolvido em Angular (versão 17 ou posterior).

O backend deve ser desenvolvido em Spring Boot versão 3.1 ou posterior, com Java versão 21.

Apresentação do resultado

Os códigos-fonte das aplicações devem ser disponibilizados em repositórios github privados, compartilhados com o usuário mngsgoncalves.

Avaliação do resultado

Na avaliação do resultado será considerado:

- Atendimento aos requisitos apresentados
- Organização do código e documentação
- Atenção a implementações de validações dos dados e processos
- Usabilidade da interface

Dúvidas

Dúvidas sobre o teste podem ser encaminhadas para goncalves@mngs.com.br.

Backend

Modelo de dados

Persistidos:

Pais (id (chave primária sequencial), nome (único), sigla (único), gentilico (único))

Token (id (chave primária sequencial), token, login, expiracao (timestamp), administrador (boolean))

Usuario (id (chave primária sequencial), login (único), senha (hash), nome, administrador (boolean))

Para resposta de requisições apenas (em memória):

UsuarioAutenticado (login, nome, token, administrador (boolean), autenticado (boolean))

Endpoints

/usuario – ações relativas a objetos Usuario

Ação	Descrição	Método	Parâmetros	Retorno
/autenticar	Autenticar o usuário – consulta a tabela de usuários por um registro com os dados passados.	POST	login – login do usuário senha – senha do usuário	Objeto UsuarioAutenticado indicando se o usuário foi autenticado, e em caso positivo, com o token gerado na autenticação, incluindo também a informação se o usuário é administrador.
/renovar-ticket	Renova um ticket gerado anteriormente	GET	token – token de autenticação gerado anteriormente	true se reautenticou o token, false se não conseguiu reautenticar (token não existe, por exemplo)

/pais – ações relativas a objetos Pais

Ação	Descrição	Método	Parâmetros	Retorno
/listar	Lista os países cadastrados	GET	token – token de autenticação gerado anteriormente	Lista dos países ou HTTP_ERROR 401 se não autenticado
/salvar	Inclui/altera um país cadastrado. Se recebe um Pais com id 0, inclui um novo país, caso contrário altera o país com id informado	POST	token – token de autenticação gerado anteriormente, passado na URL corpo da requisição (body em formato JSON) - entidade Pais	O objeto pais salvo atualizado ou HTTP_ERROR 401 se não autenticado ou o usuário não é administrador
/pesquisar	Retorna os países cujo nome contém o texto informado (case insensitive)	GET	token – token de autenticação gerado anteriormente filtro – texto a ser considerado para o filtro	Lista dos países ou HTTP_ERROR 401 se não autenticado
/excluir	Remove o país de id informado	GET	token – token de autenticação gerado anteriormente id – id do país a excluir	true se excluiu, false se não encontrou ou HTTP_ERROR 401 se não autenticado ou o usuário não é administrador

Dados iniciais

O backend deve vir com os seguintes dados iniciais ao ser carregado:

```
insert into usuario(id,login,senha,nome,administrador) values
(Seq_USUARIO.NEXTVAL, 'convidado', 'manager', 'Usuário convidado', 0);
insert into usuario(id,login,senha,nome,administrador) values
(Seq_USUARIO.NEXTVAL, 'admin', 'suporte', 'Gestor', 1);
insert into pais(id,nome,sigla,gentilico) values (Seq_PAIS.NEXTVAL,
'Brasil', 'BR', 'Brasileiro');
insert into pais(id,nome,sigla,gentilico) values (Seq_PAIS.NEXTVAL,
'Argentina', 'AR', 'Argentino');
insert into pais(id,nome,sigla,gentilico) values (Seq_PAIS.NEXTVAL,
'Alemanha', 'AL', 'Alemão');
```

Os tipos/nomes de campos são meramente ilustrativos. Não é obrigatório que a inserção seja via SQL. Atentar que a senha deve ser salva como **hash**.

Funcionamento da autenticação e tokens

Ao solicitar a autenticação do usuário, para gerar um token, o sistema deve:

1. Conferir login/senha
2. Gerar objeto Token, com um valor de token gerado de alguma forma dinâmica, que não seja previsível
3. Persistir o objeto Token

O token deve expirar em 5 minutos a partir da sua geração

Ao solicitar a renovação de um token:

1. O sistema deve validar a existência do token
2. O sistema deve atualizar a expiração do mesmo para 5 minutos no futuro

Infraestrutura

Utilizar OpenAPI para publicar a documentação da interface REST e disponibilizar navegação via Swagger-UI.

Utilizar banco de dados H2 ou HSQL.

Persistência via JPA utilizando Spring Data.

Frontend

A estrutura/fluxo esperado da SPA a ser desenvolvida em Angular, como frontend, é:

1. Tela de login, solicitando login e senha
 - a. Autenticação feita contra o backend
 - b. Deve salvar o token no storage local da aplicação no navegador, assim como a informação de se o usuário é administrador ou não (vem nas informações de resposta da autorização)
2. Tela principal com menu superior simples, com uma opção "Países". Na parte superior da tela o nome do usuário deve ser apresentado (vem nas informações de resposta da autorização)
3. Tela para consulta/edição de países
 - a. Só permite editar/excluir/incluir se for administrador
 - b. Deve exibir lista dos países em tabela paginada, com paginação no cliente e ordenação também no cliente
 - c. Deve permitir editar/incluir/excluir países
 - d. Os campos devem ter as validações adequadas aos dados (como sigla do estado ser 2 caracteres textuais, por exemplo)
4. Como o token expira, deve ser renovado antes de cada requisição ou capturar os retornos 401 do backend e reautorizar o token automaticamente, tornando o processo transparente ao usuário.