

Bases de Dados 2020/21

Projeto
“Leilões Online”

User Manual

Ana Beatriz Marques, 2018274233

Ana Mateus, 2018282874

Gonçalo Coelho, 2018265127

Índice

1. Introdução -----	3
2. Entrada no sistema -----	4
2.1 Registo de utilizador	
2.2 Login	
3. Funcionalidades do utilizador-----	5
3.1 Criação Leilão	
3.2 Listar leilões	
3.3 Pesquisa leilão	
3.4 Consultar detalhes leilão	
3.5 Listar envolvimento	
3.6 Licitar num leilão	
3.7 Editar leilão	
3.8 Escrever no mural	
3.9 Leitura mensagens	
4. Funcionalidades de administrador -----	8
4.1 Cancelar leilão	
4.2 Banir user	
4.3 Estatística	
Contactos -----	8

1. Introdução

Após a correta instalação dos programas (descrito no INSTALLATION MANUAL), iniciamos o programa abrindo o terminal na pasta do projeto e correndo o seguinte comando:

```
nodemon src/dbproj.js  
ou, em alternativa:  
npm run dev
```

Isto inicia a base de dados associada. De seguida, usamos o Postman para enviar comandos para o sistema. Caso necessário, uma breve explicação de como usar o postman pode ser encontrada em:

<https://www.youtube.com/watch?v=t5n07Ybz7yI>

2. Entrada no sistema

Ao entrar no sistema, é necessário selecionar entre as seguintes opções:

- Registrar uma nova conta (1.1)
- Entrar numa conta existente (1.2)

2.1 Registrar uma nova conta introduzindo os valores associados ao utilizador: username, email e password.

req POST <http://localhost:8080/dbproj/user>

```
1 {  
2   ... "username": "ana12",  
3   ... "email": "ana12@student.dei.uc.pt",  
4   ... "password": "pass",  
5   ... "admin": "false"  
6 }
```

res: User created

2.2 Entrar numa conta existente introduzindo o login: username e password.

req PUT <http://localhost:8080/dbproj/user>

Inserir:

```
1 {  
2   ... "username": "anaadmin2",  
3   ... "password": "pass"  
4 }
```

res: {"authToken": <código de autenticação>}

Notas:

Por motivos de segurança, sempre que existe entrada de um utilizador, é iniciado um timer que realiza automaticamente logout consoante o tempo designado. Caso tenha esgotado o tempo, basta re-fazer login no sistema sem qualquer complicação.

Também por este motivo, é necessário em todos os requests alterar o authToken para o atual do login antes de submeter o mesmo.

3. Funcionalidades do utilizador

Após a entrada no sistema, temos várias funcionalidades disponíveis para todos os utilizadores:

3.1 Criar novo leilão de um determinado artigo e inseri-lo diretamente na BD, com os todas as informações necessárias do leilão.

req POST http://localhost:8080/dbproj/leilao

Inserir no body:

```
1 {  
2   .... "authToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6I...",  
3   .... "titulo": "novo leilao",  
4   .... "descricao": "uma descricao do artigo",  
5   .... "artigoid": 1000000023,  
6   .... "minpreco": 100,  
7   .... "datacomeco": "2021-05-31 16:50:06-07",  
8   .... "datafim": "2021-05-31 20:10:06-07"  
9 }
```

res {"leilaoId": id do leilão}

3.2 Listar todos os leilões existentes.

req GET <http://localhost:8080/dbproj/leiloes>

res {"leilaid": id do leilão 1, "leilaid": id do leilão 2, (...) }

3.3 Pesquisar leilões existentes através do seu artigo (código EAN/ISBN) ou descrição.

req GET <http://localhost:8080/dbproj/leiloes/{keyword}>

res {"leilaid": id do leilão, (...) }

3.4 Consultar todos os detalhes de um leilão pelo seu id.

req GET <http://localhost:8080/dbproj/leilao/{leilaid}>

res {"leilaid": id do leilão, "titulo": "Titulo do leilão", (...)}

3.5 Listar todos os leilões na qual o user esteve envolvido (criador do leilão ou licitou no leilão).

req GET <http://localhost:8080/dbproj/user/historico>

res {leilaid : id do leilão, (...) }

3.6 Efetuar uma licitação num leilão ativo. O valor da licitação tem de ser superior à maior licitação válida atual, ou caso não exista nenhuma, superior ao valor mínimo imposto pelo leiloador.

req GET <http://localhost:8080/dbproj/{leilaid}/{licitacao}>

3.7 Editar detalhes de um leilão (apenas disponível para o dono do leilão).

req PUT <http://localhost:8080/dbproj/leilao/{leilaid}>

Inserir no body:

```
1 {  
2   ... "authToken": "eyJhbGciOiJIUzI1Ni"  
3   , ... "titulo": "novo titulo",  
4   ... "descricao": "nova descricao"  
5 }
```

```
res {"leilaoId": id do leilão}
```

3.8 Escrever mensagens no mural de um leilão. O mural é o aglomerado de comentários, questões ou esclarecimentos associados ao leilão de forma pública.

```
req PUT http://localhost:8080/dbproj/leilao/{leilaid}/mural
```

Inserir no body:

```
1 {  
2   ... "authToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTJ5bnVpdC9yYXZlcnQxMjM0NTY3ODk=",  
3   ... "texto": "uma mensagem de teste"  
4 }
```

```
res {"success": "Nova mensagem no mural do leilão X."}
```

3.9 Ler mensagens privadas recebidas pelo utilizador.

```
reg GET http://localhost:8080/dbproj/user
```

4. Funcionalidades de administrador

Para uma melhor gerência do sistema, temos também funções exclusivas para admins:

4.1 Cancelar um leilão.

req PUT <http://localhost:8080/dbproj/leilao/{leilaoId}/cancel>
res {"leilaoId": id do leilão}

4.2 Banir um utilizador, cancelando todas as licitações e leilões do mesmo.

req PUT <http://localhost:8080/dbproj/user/admin/ban>
{"userToBan": username}

4.3 Obter as estatísticas da aplicação.

- top 10 utilizadores com mais leilões criados;
- top 10 utilizadores que mais leilões venceram;
- total de leilões ativos nos últimos 10 dias;

req GET <http://localhost:8080/dbproj/estatisticas>

Qualquer dúvida contactar-nos através de um dos seguintes emails:

*anafm@student.dei.uc.pt
almateus@student.dei.uc.pt
gdcoelho@student.dei.uc.pt*