



BidYourAuction

Manual de Utilizador

Leilões Online

2018284515	Ana Rita Rodrigues	analr@student.dei.uc.pt	PL5
2018295474	Bruno Faria	brunofaria@student.dei.uc.pt	PL5
2018233092	Dylan Perdigão	dgp@student.dei.uc.pt	PL5

Coimbra

2021

Índice

Introdução	3
Funcionalidades	4
Registrar de um utilizador	4
Autenticar um utilizador/administrador	5
Criar um leilão	6
Listar leilões existentes	7
Pesquisar leilões existentes	8
Consultar detalhes de um leilão	9
Listar leilões onde um utilizador tenha atividade	10
Licitar	11
Editar detalhes de um leilão	12
Escrever mensagem no mural	13
Listar notificações	14
Verificar se um leilão acabou	15
Cancelar um leilão	16
Banir definitivamente um utilizador	17
Obter estatísticas	18
Outras Mensagens de Erros	19

Introdução

O presente documento surge como **Manual do Utilizador para apoio ao cliente** aquando o manuseamento da aplicação através da troca de request / response REST, focando-se nos seguintes objetivos:

- **Descrição do sistema**, detalhando as suas principais características;
- **Instruções passo a passo das funcionalidades que constam da aplicação**, de modo a capacitar os utilizadores da correta utilização do sistema;

Funcionalidades

Registar de um utilizador

O registo de um utilizador é feito a partir do seguinte request com o respetivo body:

```
POST http://localhost:8080/dbproj/user
```

```
{
  "username": "maria",
  "email": "maria@email.com",
  "password": "password"
}
```

Onde o `username`, `email` e `password` são respetivamente o nome de utilizador, o endereço de correio eletrónico e a palavra-passe. A resposta resultante devolve o identificador da pessoa (`userId`) na base de dados:

```
{
  "userId": 7
}
```

São efetuadas verificações de input que podem resultar numa resposta com código de erro caso o *body* do *request* seja inválido.

```
{
  "erro": 401
}
```

Autenticar um utilizador/administrador

A autenticação de um utilizador ou de um administrador é feita pela seguinte *request* e *respetivo body*:

```
PUT http://localhost:8080/dbproj/user
```

```
{
  "username": "mara",
  "password": "password"
}
```

Como podemos observar o *body* do *request* é composto pelo *username* e pela *password* do nosso utilizador ou administrador. Em caso de sucesso é devolvido o *token* de identificação (*authToken*) que servirá para executar os *requests* subjacentes uma vez que necessitam de um utilizador com *login* feito. O token é devolvido da seguinte forma:

```
{
  "authToken": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOiJlY2MjI0ODU0NTgsImhhdCI6MTYyMjQ4MTg1OAwic3ViIjoibWFyY28ifQ.Vc-i5eThHZ0BL90vLmQnburMrdx-Rk3uhOW_9d7MrbI"
}
```

Caso o *login* esteja inválido é devolvido um erro com uma mensagem indicando que as credenciais estão erradas.

```
{
  "erro": 401,
  "message": "Wrong credentials"
}
```

Criar um leilão

A criação de um leilão necessita que o utilizador esteja com *login* feito. A partir daí o *request* e o *body* são os seguintes:

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

```
{
  "artigoId": 69,
  "precoMinimo": 10000.00,
  "titulo": "Lingote de ouro",
  "descricao": "Ouro puro (24 quilates)",
  "dataFim": "2021-06-10 23:59"
}
```

No *request* não é necessário especificar nenhum campo, no entanto, no body deve-se colocar o identificador do artigo (código EAN / ISBN), o preço mínimo que o produto pode ser licitado inicialmente, o título, a descrição e a data de fim, sendo que a data de início é considerada a data de criação do leilão.

Em caso de sucesso é devolvido o identificador do leilão criado:

```
{
  "leilãoId": 1
}
```

Listar leilões existentes

A listagem de leilões que estão a decorrer passa pelo seguinte request:

```
GET http://localhost:8080/dbproj/leiloes
```

```
[
  {
    "descricao": "Desc v2",
    "leilaoId": -1
  },
  {
    "descricao": "Desc v1",
    "leilaoId": -2
  },
  {
    "descricao": "Ouro puro (24 quilates)",
    "leilaoId": 1
  }
]
```

Como resposta recebemos uma lista de identificadores e descrições. Para cada leilão é escolhida e apresentada a descrição correspondente à versão mais recente.

Pesquisar leilões existentes

Para pesquisar leilões no sistema é preciso usar o seguinte *request*:

```
GET http://localhost:8080/dbproj/leiloes/{keyword}
```

Onde o campo `{keyword}` permite pesquisar semelhança com o título ou com a descrição de um leilão. É devolvido uma lista com os resultados que tenham uma descrição ou código que combine com a palavra-chave introduzida.

```
[
  {
    "descricao": "Ouro puro (24 quilates)",
    "leilaoId": 1
  }
]
```

Caso não haja similaridade com a palavra-chave é devolvido o seguinte erro:

```
{
  "Ups": "Sem resultados para esta pesquisa!"
}
```


Consultar detalhes de um leilão

Os detalhes de um determinado leilão podem ser consultados com o seguinte *request*:

```
GET http://localhost:8080/dbproj/leilao/{leilaoId}
```

Onde `{leilaoId}` é o identificador do leilão destinado a ser consultado. A resposta devolvida pelo servidor contém a data de fim (`dataFim`), a última versão da descrição do leilão (`descricao`), o identificador do leilão (`leilaoId`), uma lista com as licitações (`licitacoes`), uma lista com as mensagens do mural (`mensagens`) e o título (`titulo`).

```
{
  "dataFim": "Thu, 10 Jun 2021 23:59:00 GMT",
  "descricao": "Ouro puro (24 quilates)",
  "leilaoId": 1,
  "licitacoes": [],
  "mensagens": [],
  "titulo": "Ouro"
}
```

Listar leilões onde um utilizador tenha atividade

Um utilizador pode consultar os seus leilões onde tenha efetuado licitações com o seguinte *request*:

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

O servidor responde com uma lista contendo a última versão das descrições e os identificadores dos leilões.

```
[
  {
    "descricao": "Desc v1",
    "leilaoId": -2
  },
  {
    "descricao": "Desc v2",
    "leilaoId": -1
  }
]
```

Licitar

Uma licitação é efetuada da seguinte forma:

```
POST http://localhost:8080/dbproj/licitar/{leilaoId}/{licitacao}
```

Onde {leilaoId} e {licitacao} são respetivamente o identificador do leilão e a quantia da licitação a realizar nesse leilão. Em caso de sucesso é devolvido o identificador da licitação realizada ({licitacaoId}).

```
{  
  "licitacaoId": 4  
}
```

Caso uma licitação tenha um valor demasiado baixo é devolvida a seguinte mensagem de erro:

```
{  
  "erro": "Licitação demasiado baixa."  
}
```

Se o erro for outro é devolvido o código de erro 401:

```
{  
  "erro": 401  
}
```

Editar detalhes de um leilão

A edição de um leilão é feita através do seguinte *request* e respetivo *body*:

```
PUT http://localhost:8080/dbproj/leilao/{leilaoId}
```

```
{
  "titulo": "Agua do Mondegoo",
  "descricao": "Agua fresca do rio Mondegoo"
}
```

Onde `{leilaoId}` no request representa o identificador do leilão a ser editado, com o título e a descrição a serem alteradas numa nova versão das propriedades textuais de um leilão.

```
{
  "Ativo": true,
  "Criador": "marco",
  "DataFim": "Thu, 10 Jun 2021 23:59:00 GMT",
  "DataIni": "Mon, 31 May 2021 17:35:57 GMT",
  "Descricao": "Agua fresca do rio Mondegoo",
  "Titulo": "Agua do Mondegoo",
  "codigo": 69,
  "leilaoId": 1,
  "precoMin": 10000.0
}
```

Escrever mensagem no mural

Para escrever uma mensagem no mural de um leilão determinado por `{leilaoId}`, usa-se o seguinte *request* e *body*:

```
POST http://localhost:8080/dbproj/feed/{leilaoId}
```

```
{
  "message" : "O que justifica o preco do artigo?",
  "type" : "question"
}
```

No body, escreve-se o conteúdo da mensagem no campo `message` e o tipo de mensagem no campo `type`. Os tipo de mensagem válidos são:

- `question`
- `clarification`
- `comment`

A resposta devolve o identificador da mensagem que foi escrita no mural.

```
{
  "messageId": 1
}
```

Listar notificações

A listagem das notificações é feita através do seguinte *request*:

```
GET http://localhost:8080/dbproj/inbox
```

O servidor responde com uma lista contendo para cada notificação a data e hora de envio da (*data*) e com o seu respetivo conteúdo (*mensagem*):

```
[
  {
    "data": "Mon, 31 May 2021 17:54:44 GMT",
    "mensagem": "Nova mensagem no feed do seu leilão 1"
  }
]
```

Verificar se um leilão acabou

O seguinte *request* serve para atualizar a base de dados dos término dos leilões, ou seja, sempre que este *request* é chamado todos os leilões cuja data de fim seja inferior à data do momento ficam inativos e dá-se o apuramento do vencedor e da respetiva licitação (que o fez ganhar o leilão) através de um *trigger*.

```
PUT http://localhost:8080/dbproj/leilao/checkFinish
```

Independentemente se algum leilão tenha terminado ou não, é sempre retornada uma mensagem de sucesso.

```
{  
  "status": "success"  
}
```

Cancelar um leilão

O cancelamento de um leilão é chamado com o seguinte *request*:

```
PUT http://localhost:8080/dbproj/leilao/cancel/{leilaoId}
```

Onde o `{leilaoId}` é o identificador do leilão a ser cancelado. Em caso de sucesso é devolvido o estado (`Estado`) e o identificador do leilão cancelado (`leilaoId`):

```
{  
  "Estado": "Cancelado",  
  "leilaoId": "-1"  
}
```

Caso seja um utilizador a tentar cancelar o leilão, a seguinte mensagem de erro é enviada:

```
{  
  "erro": "Sem permissões de administrador!"  
}
```


Banir definitivamente um utilizador

O banimento de um utilizador é feito através do seguinte request:

```
PUT http://localhost:8080/dbproj/ban/{username}
```

Sendo o campo `{username}` o nome do utilizador a banir. O banimento só pode ser efetuado depois do administrador ter a sessão iniciada. Um exemplo de resposta devolvida é:

```
{
  "adminId": 9,
  "userId": -5
}
```

Onde é devolvido o identificador do administrador que baniu e o identificador do utilizador banido por ele.

Mais concretamente, ao banir o utilizador, ele é adicionado numa tabela da base de dados (*admin_participant*) que irá relacioná-lo ao administrador que o baniu. Ao efetuar essa ação é acionado um trigger que:

- muda o valor da coluna *isbanned* do utilizador banido para *True* na tabela *participant*;
- muda o valor da coluna *iscancelled* dos leilões criados pelo utilizador banido para *True* na tabela *auctions*;
- muda o valor da coluna *isinvalided* das licitações efetuadas pelo utilizador banido e das licitações superiores para *True* na tabela *bid*;
- muda o valor da maior licitação para a maior licitação do utilizador banido na tabela *bid*;
- escreve no mural (tabela *feed_message*) um comentário a lamentar o incomodo pelo banimento do utilizador;
- envia notificações aos utilizadores participando nos leilões afetados (tabela *notification*).

Obter estatísticas

A obtenção de estatísticas sobre o top 10 dos utilizadores com mais leilões criados, o top 10 utilizadores que mais leilões venceram e número total de leilões nos últimos 10 dias é somente feita pelo administrador com conta iniciada. O request associada é a seguinte:

```
http://localhost:8080/dbproj/stats
```

A resposta associada segue a seguinte formatação:

```
{
  "numeroDeLeiloesNosUltimosDezDias": 5,
  "topCriadores": [
    {
      "leiloesCriados": 3,
      "username": "dylantoban"
    },
    {
      "leiloesCriados": 1,
      "username": "marco"
    }
  ],
  "topVencedores": []
}
```

Onde são devolvidos:

- o número de leilões ativos nos últimos 10 dias (`numeroDeLeiloesNosUltimosDezDias`);
- uma lista (`topCriadores`) dos 10 utilizadores que mais leilões criaram. Devolvendo o username e número de leilões criados pelo mesmo.
- uma lista (`topVencedores`) com os 10 maiores vencedores de leilões. Devolvendo o username e número de leilões vencidos.

Outras Mensagens de Erros

Caso um request tenha uma formatação errada é devolvida a página de erro “404 not Found”:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<title>404 Not Found</title>
<h1>Not Found</h1>
<p>The requested URL was not found on the server. If you entered the URL manually please
check your spelling and try
again.</p>
```

Quando um leilão não existe é retornado o seguinte erro:

```
{
  "erro": "O leilão não existe!"
}
```

Quando um leilão foi cancelado é retornado o seguinte erro:

```
{
  "erro": "O leilão já está cancelado!"
}
```

Em caso de insucesso, será retornado o erro 404 ou 401. Quando um utilizador não tem permissões para criar um leilão, como o caso do administrador, surge o erro 401.

```
{
  "erro": 401
}
```

Caso qualquer um dos inputs inseridos não estiver no formato correto é devolvido o erro 404.

```
{
  "erro": 404
}
```