



UNIVERSIDADE D
COIMBRA

Projeto de Base de Dados
Licenciatura em Engenharia Informática
3º Ano – 2º Semestre
2020/2021

MANUAL DE UTILIZADOR
Aplicação Web

Realizado por:
Inês Teixeira - 2018296643
Pedro Tavares - 2018280907
Rui Tapadinhas- 2018283200

1. Login

Request:

Método : PUT

Url (eg): <http://192.168.4.6:5000/login/>

Body (json)(eg):

```
{  
  "user_name": "ines",  
  "password": "password"  
}
```

Explicação:

user_name: inserir username escolhido ao se registar

password : inserir a password escolhida ao se registar

Response:

Body(json)(eg):

```
{  
  "authToken": 4316  
}
```

Explicação:

authToken: token necessário para a execução das funcionalidades(serve como verificação)

2. Registo Utilizadores

Request:

Método : POST

Url (eg): <http://192.168.4.6:5000/registoUtilizador>

Body (json)(eg):

```
{  
  "user_name": "ines",  
  "email": "ines@hotmail.com",  
  "nome": "ines teixeira",  
  "password": "password",  
  "genero": "feminino",  
  "nif": 12365478,  
  "data_nasc": "18-12-2000",  
  "estado": "solteiro",  
  "contacto": 96258741,  
  "is_admin": true  
}
```

Explicação:

user_name: inserir um username
email: inserir um email ao qual quer associar a conta
nome: inserir o seu nome
password : inserir a password escolhida ao se registar
genero: inserir o seu gênero
nif: inserir o seu nif
data_nasc : inserir a sua data de nascimento
estado: inserir o seu estado civil
contacto: inserir o seu número telefónico
is_admin: inserir true se for admin, inserir false se não for admin

Response:

Body(json)(eg):

Explicação:

3. Criar artigo

Request:

Método : POST

Url (eg): <http://192.168.4.6:5000/artigo/>

Body (json)(eg):

```
{  
  "token": 3663,  
  "id_artigo": 1321,  
  "codigoisbn": 101212,  
  "nome_artigo": "computador",  
  "categoria": "tecnologia",  
  "descricao": "Computador sistema operativo windows com 20 polegas",  
  "utilizador_user_name": "pedro"  
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada
id_artigo: id do artigo a adicionar
codigoisbn: código isbn do artigo a adicionar
nome_artigo: nome do artigo a adicionar
categoria: categoria do artigo a adicionar
descricao: descrição do artigo a adicionar

utilizador_user_name: nome do utilizador a quem o artigo a adicionar pertence

Response:

Body(json)(eg):

```
{  
  "artigoId": 1321  
}
```

Explicação:

artigoId: id do artigo que foi adicionado

4. Criar leilão

Request:

Método : POST

Url (eg): <http://192.168.4.6:5000/leilao/>

Body (json)(eg):

```
{  
  "id_artigo": 132246,  
  "id_leilao": 12346,  
  "data_ini": "01-04-2021 12:30:30",  
  "data_fim": "14-06-2021 13:30:30",  
  "preco_base": 2,  
  "token": 16838  
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada

id_artigo: id do artigo a associar

id_leilao: id leilao do artigo a adicionar

nome_artigo: nome do artigo a adicionar

data_ini: data de começo do leilao

data_fim : data de fim do leilao

preco_base: valor inicial do artigo

Response:

Body(json)(eg):

```
{
```

```
"leilaoId": 12346
}
```

Explicação:

leilaoId: id do leilao que foi adicionado

5. Listar todos os leilões a decorrer

Request:

Método : GET

Url (eg): <http://192.168.4.6:5000/leilao/>

Body (json)(eg):

```
{
  "token": 10275
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada

Response:

Body(json)(eg):

```
{
  "lista_leiloes": [
    {
      "descricao": "abstrata",
      "leilaoId": 123
    },
    {
      "descricao": "abstrata",
      "leilaoId": 124
    },
    {
      "descricao": "carro",
      "leilaoId": 12346
    },
    {
      "descricao": "Computador sistema operativo windows com 20 polegas",
      "leilaoId": 11236
    }
  ]
}
```

Explicação:

lista_leiloes: lista de dicionários em que cada um contém a descrição e o id de um leilão estando este leilão a decorrer.

6. Pesquisar os leilões existentes

Request:

Método : GET

Url (eg): <http://192.168.4.6:5000/leiloes/keyword>

Body (json)(eg):

```
{
  "token": 10275
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada

Response:

Body(json)(eg):

```
[
  {
    "descricao": "abstrata",
    "leilaoid": 123
  },
  {
    "descricao": "abstrata",
    "leilaoid": 124
  }
]
```

Explicação:

Lista que contém a descrição e o id de cada leilão que tenha a keyword inserida na descrição ou o id igual a keyword.

7. Consultar detalhes de um leilão

Request:

Método : GET

Url (eg): <http://192.168.4.6:5000leilao/123>

Body (json)(eg):

```
{
  "token": 10275
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada

Response:

Body(json)(eg):

```
[
  {
    "categoria": "veiculo",
    "data_fim": "Tue, 15 Jun 2021 15:03:08 GMT",
    "data_ini": "Tue, 01 Jun 2021 23:03:02 GMT",
    "descricao": "carro",
    "leilaold": 12,
    "nome_artigo": "carro",
    "preco_base": 1500
  },
  {
    "Comentarios": []
  },
  {
    "Licitacoes": []
  },
  {
    "Estado": "Terminado"
  }
]
```

Explicação:

Lista os dados de um leilão, seus comentários, licitações e estado.

8. Listar todos os leilões em que um utilizador tenha atividade

Request:

Método : GET

Url (eg): <http://192.168.4.6:5000/leiloes/user/>

Body (json)(eg):

```
{
  "token": 10275
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada

Response:

Body(json)(eg):

```
{
  "leiloes": [
    {
      "categoria": "arte",
      "data_fim": "Wed, 02 Jun 2021 14:12:03 GMT",
      "data_ini": "Fri, 28 May 2021 21:30:12 GMT",
      "descricao": "abstrata",
      "leilaold": 123,
      "nome_artigo": "quadro",
      "preco_base": 1500
    },
    {
      "categoria": "arte",
      "data_fim": "Wed, 02 Jun 2021 14:12:03 GMT",
      "data_ini": "Fri, 28 May 2021 21:30:12 GMT",
      "descricao": "abstrata",
      "leilaold": 123,
      "nome_artigo": "quadro",
      "preco_base": 1500
    }
  ]
}
```

Explicação:

_____ Devolve um dicionário com chave “leilões” e com valor igual a uma lista de dicionários com todos os dados de cada tabela leilão.

9. Efetuar licitação

Request:

Método : GET

Url (eg): http://192.168.4.6:5000/leiloes/<id_leilao>/<preco_licitacao>

Body (json)(eg):

```
{  
  "token": 10275  
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada

No url é indicado o leilão e o preço da licitação para essa licitação e o nome de utilizador é obtido a partir do token.

Response:

Body(json)(eg):

```
"Sucesso"
```

Explicação: Apenas é retornada uma resposta positiva em caso de sucesso.

10. Editar leilão

Request:

Método : PUT

Url (eg): <http://192.168.4.6:5000/leilao/editar/11>

Body (json)(eg):

```
{  
  "data_ini": "22-06-2020 12:30:30",  
  "data_fim": "22-06-2021 13:30:30",  
  "preco_base": 20,  
  "nome_artigo": "livro bd",  
  "categoria": "livros",  
  "descricao": "livro sobre bases de dados",  
  "token": 20246  
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada
Todos os pares para além do token são opcionais e só são enviados nos
parametros que se pretende que sejam alterados.

Response:**Body(json)(eg):**

```
{  
  "Leilão já foi licitado": "impossível editar"  
}
```

Explicação:

_____ É retornada uma resposta de sucesso ou de insucesso apenas para verificação
do sucesso da operação.

11. Escrever Mural

Request:**Método :** POST**Url (eg):** http://192.168.4.6:5000/escreve_msg_mural**Body (json)(eg):**

```
{  
  "type": "comentario",  
  "texto": "gostei do artigo",  
  "id_leilao": 123,  
  "token": 6473  
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada
Todos os dados da mensagem para o mural.

Response:**Body(json)(eg):**

```
{  
  "Mensagem publicada com sucesso"  
}
```

Explicação:

_____ É retornada uma resposta de sucesso ou de insucesso apenas para verificação
do sucesso da operação.

12. Terminar Leilão

Request:

Método : GET

Url (eg): <http://192.168.4.6:5000/terminarLeiloes>

Body (json)(eg):

```
{  
  "token": 15077  
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada

Response:

Body(json)(eg):

"Leiloes terminados!!"

Explicação:

É retornada uma resposta de sucesso ou de insucesso apenas para verificação do sucesso da operação.

13. Cancelar Leilão

Request:

Método : GET

Url (eg): <http://192.168.4.6:5000/cancelarLeilao/123>

Body (json)(eg):

```
{  
  "token": 15077  
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada

Response:

Body(json)(eg):

"Leilao Cancelado!!": "12346"

Explicação:

_____É retornada uma resposta de sucesso ou de insucesso apenas para verificação do sucesso da operação com o id do leilão que foi cancelado.

14. Top 10 vencedores

Request:

Método : GET

Url (eg): http://192.168.4.6:5000/top10_vencedores

Body (json)(eg):

```
{
  "token": 15077
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada

Response:

Body(json)(eg):

```
[
  {
    "total de artigos comprados": 7,
    "user": ""
  },
  {
    "total de artigos comprados": 5,
    "user": "ines"
  }
]
```

Explicação:

Retorna um dicionário

15. Top 10 artigos

Request:

Método : GET

Url (eg): http://192.168.4.6:5000/top10_artigos

Body (json)(eg):

```
{  
  "token": 15077  
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada

Response:

Body(json)(eg):

```
[  
  {  
    "total de artigos leiloados": 5,  
    "user": "pedro"  
  },  
  {  
    "total de artigos leiloados": 3,  
    "user": "tapadinhas"  
  },  
  {  
    "total de artigos leiloados": 3,  
    "user": "ines"  
  }  
]
```

Explicação:

16. Top 10 leiloadores

Request:

Método : GET

Url (eg): http://192.168.4.6:5000/top10_leiloadores

Body (json)(eg):

```
{
```

```
"token": 15077
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada

Response:

Body(json)(eg):

```
[
  {
    "total de artigos leiloados": 6,
    "user": "pedro"
  },
  {
    "total de artigos leiloados": 3,
    "user": "tapadinhas"
  },
  {
    "total de artigos leiloados": 3,
    "user": "ines"
  }
]
```

Explicação:

_____Retorna os top

17. Top 10 leiloes

Request:

Método : GET

Url (eg): http://192.168.4.6:5000/top10_leiloes

Body (json)(eg):

```
{
  "token": 15077
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada

Response:

Body(json)(eg):

```
[
  {
    "total_leiloes": [
      4
    ]
  }
]
```

Explicação:

18. Receber notificações

Request:

Método : GET

Url (eg): http://192.168.4.6:5000/notificacoes/

Body (json)(eg):

```
{
  "token": 15077
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada

Response:

Body(json)(eg):

```
{
  "0": {
    "Data": "Mon, 29 May 2000 21:02:06 GMT",
    "Mensagem": "init notificacao"
  },
  "4": {
    "Data": "Sat, 29 May 2021 21:02:06 GMT",

```

```

      "Mensagem": "Nova mensagem no mural do leilão 123"
    },
    "6": {
      "Data": "Sat, 29 May 2021 21:02:06 GMT",
      "Mensagem": "Nova mensagem no mural do leilão 123"
    },
    "8": {
      "Data": "Mon, 31 May 2021 21:52:13 GMT",
      "Mensagem": "Nova mensagem no mural no leilão 123"
    },
    "14": {
      "Data": "Mon, 31 May 2021 22:02:36 GMT",
      "Mensagem": "O leilao 123 que licitou foi cancelado."
    }
  }
}

```

Explicação:

19. Banir utilizador

Request:

Método : POST

Url (eg): <http://192.168.4.6:5000/admin/ban/ines>

Body (json)(eg):

```

{
  "token": 15077
}

```

Explicação:

token: verifica se houve um login antes da operação ser realizada

Response:

Body(json)(eg):

```

{
  "Leilões atualizados": [
    123,
    124
  ],
}

```



```
"User banned": "ines"
}
```

Explicação:

É retornado um dicionário com uma lista dos leilões que sofreram alterações e um par com o utilizador que foi banido.

20. Listar

Request:

Método : GET

APP ROUTE: /comentarios/<leilaoId>

Body (json)(eg):

```
{
  "token": 15077
}
```

Explicação:

token: verifica se houve um login antes da operação ser realizada

Response:

Lista todos os comentarios de uma leilao .