Desafio Sanar Backend
Documentação

Autor: Ubiratan Correia Barbosa Neto

Descrição do projeto

O objetivo do desafio foi a construção de um serviço web RESTful que integre o SanarFlix com a MundiPagg. Foi requisitado também algumas funcionalidades obrigatórias, que constam na descrição do desafio. O projeto foi realizado em Python 3.6 e será detalhado neste documento suas informações e todas as suas funcionalidades.

Requisitos

Para a execução da API, é necessário Python 3.6, além das bibliotecas listadas abaixo (e suas dependências):

- jsonpickle==0.7.1
- CacheControl==0.11.7
- requests==2.18.4
- Flask_SQLAlchemy==2.4.0
- Flask==1.1.1
- Flask_RESTful==0.3.7
- python_dateutil==2.8.0

Caso possua o ‘virtualenv’ instalado, é possível utilizar o ambiente virtual ‘env’ dentro do diretório do projeto, que já possui todas as bibliotecas instaladas.

Informações sobre a API

Execução da API

Para iniciar o servidor, é necessário estar dentro do diretório raiz do projeto e executar o script ‘app.py’. A partir desse momento, a API passará a estar em funcionamento no endereço ‘http://127.0.0.1:5000’ (localhost, usando a porta 5000) e será possível realizar requisições para ela (detalhadas em seções futuras).

Persistência dos Dados

Dados dos clientes e dos planos são armazenados numa base de dados no arquivo ‘app.db’, presente no diretório raiz do projeto. A base possui duas tabelas. A tabela ‘user’ contém informações do usuário e da sua assinatura realizada. A tabela ‘plan’ contém informações dos planos oferecidos para assinatura. Este arquivo, caso não exista, é criado na primeira inicialização da API. Caso a API seja executada novamente, os dados não serão apagados nem sobrescritos. Caso o arquivo seja apagado, o comportamento da API é o mesmo da primeira inicialização.

Sobre os Planos

Existem 4 planos disponíveis para assinatura, e suas informações são armazenadas na base de dados na primeira execução da API (ou na primeira execução após a deleção da base de dados). Cada um deles possui um identificador único, que consiste de uma string, que identifica o plano a ser usado ao realizar uma assinatura. Os planos disponíveis são:

- Plano SanarFlix Mensal: Este plano possui o valor de R$24,50 por mês, e seu identificador é a string ‘plan_mens’.
- Plano SanarFlix Mensal com Período de Teste Gratuito: Este plano é similar ao primeiro, porém a mensalidade só passa a ser cobrada após 7 dias. Seu identificador é a string ‘plan_mens_teste’.
- Plano SanarFlix Trimestral: Este plano custa R$69,90 a cada três meses, e seu identificador é a string ‘plan_trim’
- Plano SanarFlix Promocional Yellowbook: Este plano promocional custa R$164,40 no primeiro mês, e inclui o livro Yellowbook. Após o primeiro mês, ele custa o mesmo preço da assinatura mensal padrão. É identificado pela string ‘plan_promo_yellowbook’

Estes identificadores são utilizados para informar à API qual o plano correto a ser assinado. Mais informações sobre isso serão dadas posteriormente.

Funcionalidades

Nesta seção serão apresentadas as funcionalidades da API construída, com detalhes acerca de como utilizar cada uma delas. Exemplos de uso de cada uma também serão dados, utilizando o comando 'curl'. É possível importar estes comandos para serem executados no Postman.

Realizar a assinatura de um cliente

Para realizar essa operação, é necessário fazer uma requisição POST para a API no endereço 'http://127.0.0.1:5000/api/users/'. O corpo da requisição (no formato JSON) deve conter as seguintes informações:

```
{
  "cliente":
        {
          "nome": <nome do cliente : string>,
          "email": <email do cliente : string>
        },
  "produto":
        {
          "tipo": <descrição do plano : string>,
          "plano_id": <identificador do plano (descrito na seção de planos) : string>
        },
  "cartao":
        {
          "nome_cartao" :<nome como consta no cartão : string>,
          "numero": <numero do cartão : string>,
          "expiracao_mes": <mês de expiração : integer>,
          "expiracao_ano": <ano de expiração : integer>,
          "cvv": <código de verificação : string>
        }
 }
```

Um exemplo pode ser visto abaixo:

```
{
  "cliente":
        {
          "nome": “Joao Cardoso”,
          "email": "joao123@gmail.com"
        },
  "produto":
        {
          "tipo": “Plano Mensal”,
          "plano_id": “plan_mens”
        },
  "cartao":
        {
          "nome_cartao" : “JOAO CARD”,
          "numero": “4584441896453869”,
          "expiracao_mes": 12,
          "expiracao_ano": 19,
          "cvv": 591
        }
 }
```

É possível realizar essa requisição através do seguinte comando:

```
$ curl http://127.0.0.1:5000/api/users/ -X POST -d "$request_json"
```

Onde ‘request_json’ é um JSON no formato acima. A requisição retorna, em caso de sucesso, um JSON contendo as seguintes informações:

```
{
  “nome”: <nome do cliente>,
  “id”: <identificador do cliente para a API>,
  “mundi_subscription_id”: <identificador da assinatura atual para a MundiPagg>,
  “plano”: <nome do plano assinado>
  “primeiros 6 digitos do cartao”: <os primeiros dígitos do cartão utilizado>,
  “ultimos 4 digitos do cartao”: <os últimos dígitos do cartão utilizado>
}
```

Em caso de falha na requisição, uma mensagem de erro será retornada no JSON.

Obter informações de um cliente

Essa operação é realizada fazendo uma requisição GET para a API no endereço ‘http://127.0.0.1:5000/api/users/<user_id : string>’, onde ‘<user_id>’ corresponde ao identificador do cliente para a API. Este identificador é obtido como retorno de uma requisição POST. Supondo que existe um cliente assinante já cadastrado, com identificador ‘1’, é possível realizar essa requisição da forma mostrada abaixo:

$ curl http://127.0.0.1:5000/api/users/1 -X GET

Em caso de falha, o JSON conterá uma mensagem de erro relatando o problema. Em caso de sucesso, é retornado o seguinte JSON:

{
  “id”: <identificador do cliente para a API>,
  “nome”: <nome do cliente>,
  “plano”: <nome do plano assinado>,
  “mundi_customer_id”: <identificador do cliente para a MundiPagg>,
  “mundi_subscription_id”: <identificador da assinatura atual para a MundiPagg>
}

Obter informações detalhadas de um cliente

Essa operação retorna informações mais detalhadas do cliente em relação à anterior. Pode ser realizada através de uma requisição GET para a API no endereço ‘http://127.0.0.1:5000/api/users/details/<user_id : string>’, onde ‘<user_id>’ corresponde ao identificador do cliente para a API. Supondo que existe um cliente assinante já cadastrado, com identificador ‘1’, é possível realizar essa requisição da forma mostrada abaixo:

$ curl http://127.0.0.1:5000/api/users/details/1 -X GET

Em caso de falha, o JSON conterá uma mensagem de erro relatando o problema. Em caso de sucesso, é retornado o seguinte JSON:

```
{
  “cliente”:
        {
          “id”: <identificador do cliente para a API>
          “mundi_customer_id”: <identificador do cliente para a MundiPagg>,
          “nome”: <nome do cliente>
          “email”: <email do cliente>
        },
  “assinatura”:
        {
          “cartao”:
                {
                  “nome_cartao”: <nome como consta no cartão>
                  “primeiros 6 digitos”: <primeiros dígitos do cartão>
                  “ultimos 4 digitos”:  <últimos dígitos do cartão>
                  “expiracao_mes”: <mês de expiração>
                  “expiracao_ano”: <ano de expiração>
                },
          “produto”:
                  [
                    {
                      “nome”: <nome do produto ou serviço>
                      “quantidade”: <quantidade provida pela assinatura>
                      “ciclos”: <quantidade de ciclos (Nulo para assinaturas)>
                      “preco”: <preço do produto ou serviço>
                    }
                  ]
          “codigo”: <descrição da assinatura>
          “mundi_subscription_id”: <identificador da assinatura para a MundiPagg>
          “status”: <status da assinatura (cancelada ou ativa)>
          “moeda”: <moeda utilizada para o pagamento>
          “tipo de pagamento”: <tipo de pagamento>
        }
}
```

Cancelamento de assinatura de um cliente

Essa operação é realizada fazendo uma requisição DELETE para a API no endereço ‘http://127.0.0.1:5000/api/users/<user_id : string>’, onde ‘<user_id>’ corresponde ao identificador do cliente para a API. Com um cliente cadastrado, com identificador ‘1’, é possível realizar essa requisição da forma abaixo:

$ curl http://127.0.0.1:5000/api/users/1 -X DELETE

Em caso de falha, o JSON conterá uma mensagem de erro relatando o problema. Em caso de sucesso, é retornado o seguinte JSON:

```
{
  “id”: <identificador do cliente para a API>,
  “nome”: <nome do cliente>,
  “plano”: '<nome do plano>,
  “mundi_subscription_id”: <identificador da assinatura na MundiPagg,
  “status”: <status da assinatura (deverá constar como ‘canceled’)>
}
```

Alteração do cartão da assinatura

Essa operação é realizada fazendo uma requisição PUT para a API no endereço ‘http://127.0.0.1:5000/api/users/<user_id : string>’, onde ‘<user_id>’ corresponde ao identificador do cliente para a API. O corpo da requisição deve conter as seguintes informações:

```
{
  "cartao":
        {
          "nome_cartao" :<nome como consta no cartão : string>,
          "numero": <numero do cartão : string>,
          "expiracao_mes": <mês de expiração : integer>,
          "expiracao_ano": <ano de expiração : integer>,
          "cvv": <código de verificação : string>
        }
 }
```

É possível realizar essa requisição, supondo a existência de um cliente com assinatura ativa com identificador ‘1’, através da linha de código em Python:

$ curl http://127.0.0.1:5000/api/users/1 -X PUT -d "$request_json"

Onde ‘request_json’ é um JSON no formato acima. Em caso de sucesso, será retornado um JSON contendo as seguintes informações:

{
“nome”: <nome do cliente>,
“id”: <identificador do cliente para a API>,
“plano”: <nome do plano>,
“mundi_subscription_id”: <identificador da assinatura para a MundiPagg>,
“primeiros 6 digitos do cartao”: <primeiros dígitos do cartão>,
“ultimos 4 digitos do cartao”: <últimos dígitos do cartão>
}

Nova assinatura de um cliente

Essa operação só pode ser realizada se o cliente é um usuário antigo da SanarFlix que cancelou sua última assinatura. Mesmo sem possuir uma assinatura ativa, suas informações são mantidas na base de dados da API. Esta operação cria uma nova assinatura para esse cliente. Essa operação é realizada fazendo uma requisição PUT para a API no endereço ‘http://127.0.0.1:5000/api/users/<user_id : string>’, onde ‘<user_id>’ corresponde ao identificador do cliente para a API. O corpo da requisição deve conter as seguintes informações:

{
"produto":
{
"tipo": <descrição do plano : string>,
"plano_id": <identificador do plano (descrito na seção de planos) : string>
},
"cartao":
{
"nome_cartao" :<nome como consta no cartão : string>,
"numero": <numero do cartão : string>,
"expiracao_mes": <mês de expiração : integer>,
"expiracao_ano": <ano de expiração : integer>,
"cvv": <código de verificação : string>
}
}

É possível realizar essa requisição, supondo a existência de um cliente com assinatura ativa com identificador ‘1’, através do comando abaixo:

$ curl http://127.0.0.1:5000/api/users/1 -X PUT -d "$request_json"

Onde ‘request_json’ é um JSON no formato acima. Em caso de sucesso, será retornado um JSON contendo as seguintes informações:

```
{
  “nome”: <nome do cliente>,
  “id”: <identificador do cliente para a API>,
  “mundi_subscription_id”: <identificador da assinatura para a MundiPagg>,
  “plano”: <nome do plano>,
  “primeiros 6 digitos do cartao”: <primeiros dígitos do cartão>,
  “ultimos 4 digitos do cartao”: <últimos dígitos do cartão>
}
```

Testes

A API foi testada de duas formas distintas. Os scripts de teste estão localizados no diretório raiz do projeto. Inicialmente testes foram realizados para verificar as funcionalidades da API e sua conexão com a MundiPagg. Este teste foi realizado com uma instância de teste que simula a realização de requisições para a API. Os primeiros testes podem ser executados com o script ‘api_tests.py’. É importante lembrar que estes testes necessitam que a base de dados contenha as informações dos planos que podem ser assinados. Deve-se iniciar a API pela primeira vez antes de realizar estes testes, para criar a base de dados. Entretanto, não é necessário que a API esteja em funcionamento durante os testes em si. Em seguida, estes testes foram repetidos, porém realizando requisições reais utilizando a biblioteca ‘requests’ do Python, para verificar o funcionamento integral do sistema com a API em execução. Para executar estes testes, inicialmente a API deve estar em funcionamento. Eles podem ser executados com o script ‘requests_tests.py’.

Casos de Uso Requisitados

Os casos de uso requisitados pelo desafio podem ser executados pelo script ‘casos_de_uso.py’, presente no diretório raiz. Este script mostra cada um dos casos, descrevendo as requisições feitas para realizar a operação desejada. As requisições mostradas são realizadas no mesmo instante da execução do script, mostrando também o retorno de cada uma delas. É necessário que a API esteja em

funcionamento para executar o script. Um arquivo de texto de nome ‘casos_de_uso.txt’ também está presente no diretório, que mostra uma execução do script realizada antes do envio do projeto, para fins de validação. Abaixo é mostrado cada um dos casos de uso presentes no script de forma mais legível.

Caso de Uso 1: Mario assinou o SanarFlix Mensal.

Deve-se realizar uma requisicao POST para "http://127.0.0.1:5000/api/users/", passando no corpo o JSON com as informações do usuário mostrado abaixo:

```
{
  “cliente”:
        {
          “nome”: 'Mario',
          “email": “mariosantos@gmail.com”
        },
  “produto”:
        {
          “tipo”: “plano”,
          “plano_id”: “plan_mens”
        },
  “cartao”:
        {
          “nome_cartao”: “joao”,
          “numero”: “4584441896453869”,
          “expiracao_mes”: 12,
          “expiracao_ano”: 19,
          “cvv”: “591”
        }
}
```

O retorno dessa requisição pode ser visto abaixo:

```
{
  “nome”: “Mario”,
  “mundi_subscription_id”: “sub_gP6V8p2F4s3Ampkr”,
  “id”: 1,
  “plano”: “Plano SanarFlix Mensal”,
  “primeiros 6 digitos do cartao”: “458444”,
  “ultimos 4 digitos do cartao”: “3869”
}
```

Caso de Uso 2: Juliana assinou o SanarFlix com período de teste de 7 dias

Deve-se realizar uma requisicao POST para "http://127.0.0.1:5000/api/users/", passando no corpo o JSON com as informações do usuário mostrado abaixo:


```
{
  “cliente”:
        {
          “nome”: “Juliana”,
          “email": “juliana123@gmail.com”
        },
  “produto”:
        {
          “tipo”: “plano”,
          “plano_id”: “plan_mens_teste”
        },
  “cartao”:
        {
          “nome_cartao”: “juliana”,
          “numero”: “4584441896453869”,
          “expiracao_mes”: 12,
          “expiracao_ano”: 19,
          “cvv”: “591”
        }
}
```


O retorno dessa requisição pode ser visto abaixo:


```
{
  “nome”: “Juliana”,
  “mundi_subscription_id”: “sub_aRjp9JYhxTOwX2oN”,
  “id”: 2,
  “plano”: “Plano SanarFlix Mensal com 7 Dias de Teste”,
  “primeiros 6 digitos do cartao”: “458444”,
  “ultimos 4 digitos do cartao”: “3869”
}
```

Caso de Uso 3: Pedro assinou o SanarFlix Trimestral

Deve-se realizar uma requisicao POST para "http://127.0.0.1:5000/api/users/", passando no corpo o JSON com as informações do usuário mostrado abaixo:


```
{
  “cliente”:
        {
          “nome”: “Pedro”,
          “email": “pedro147@gmail.com”
        },
  “produto”:
        {
          “tipo”: “plano”,
          “plano_id”: “plan_trim”
        },
  “cartao”:
        {
          “nome_cartao”: “pedro”,
          “numero”: “4584441896453869”,
          “expiracao_mes”: 12,
          “expiracao_ano”: 19,
          “cvv”: “591”
        }
}
```


O retorno dessa requisição pode ser visto abaixo:


```
{
  “nome”: “Pedro”,
  “mundi_subscription_id”: “sub_XnLAOLdT56F3qBeY”,
  “id”: 3,
  “plano”: “Plano SanarFlix Trimestral”,
  “primeiros 6 digitos do cartao”: “458444”,
  “ultimos 4 digitos do cartao”: “3869”
}
```

Caso de Uso 4: Marcos deseja mudar o cartão de sua assinatura

Inicialmente, Marcos deve ser um assinante do SanarFlix. Realizando um procedimento similar aos anteriores, realizamos uma assinatura para Marcos (suponhamos uma assinatura mensal). É atribuído a ele o identificador ‘4’. Prosseguindo, para alterar o seu cartão, deve-se realizar uma requisicao PUT para "http://127.0.0.1:5000/api/users/4", passando no corpo o JSON com as informações do usuário mostrado abaixo:

```
{
  “cartao”:
        {
          “nome_cartao”: “marcos”,
          “numero”: “4532912167490007”,
          “expiracao_mes”: 1,
          “expiracao_ano”: 28,
          “cvv”: “123”
        }
}
```

O retorno dessa requisição pode ser visto abaixo:

```
{
  “nome”: “Marcos”,
  “mundi_subscription_id”: “sub_pXzPlozoSYfeRkNE”,
  “id”: 4,
  “plano”: “Plano SanarFlix Mensal”,
  “primeiros 6 digitos do cartao”: “453291”,
  “ultimos 4 digitos do cartao”: “0007”
}
```

Caso de Uso 5: Luiz assinou o SanarFlix Promocional com o Livro Yellowbook

Deve-se realizar uma requisicao POST para "http://127.0.0.1:5000/api/users/", passando no corpo o JSON com as informações do usuário mostrado abaixo:

```
{
  “cliente”:
        {
          “nome”: “Luiz”,
          “email": “luiz258@gmail.com”
        },
  “produto”:
        {
          “tipo”: “plano”,
          “plano_id”: “plan_promo_yellowbook”
        },
  “cartao”:
        {
          “nome_cartao”: “luiz”,
          “numero”: “4584441896453869”,
          “expiracao_mes”: 12,
          “expiracao_ano”: 19,
          “cvv”: “591”
        }
}
```


O retorno dessa requisição pode ser visto abaixo:


```
{
  “nome”: “Luiz”,
  “mundi_subscription_id”: “sub_q6ebvlZxSqcG50yz”,
  “id”: 5,
  “plano”: “Plano SanarFlix Promocional Com Livro Yellowbook”,
  “primeiros 6 digitos do cartao”: “458444”,
  “ultimos 4 digitos do cartao”: “3869”
}
```


Caso de Uso 6: Ricardo deseja cancelar a sua assinatura

Inicialmente, Marcos deve ser um assinante do SanarFlix. Realizando um procedimento similar aos anteriores, realizamos uma assinatura para Marcos (suponhamos uma assinatura trimestral). É atribuído a ele o identificador ‘6’. Para realizar seu cancelamento, é preciso efetuar uma requisição DELETE para "http://127.0.0.1:5000/api/users/6", sem informações adicionais no corpo. O retorno dessa requisição pode ser visto abaixo:

```
{
  “nome”: “Ricardo”,
  “mundi_subscription_id”: “sub_GgEolqnSVt0NkmMX”,
  “id”: 6,
  “plano”: “Plano SanarFlix Trimestral”,
  “status”: “canceled”
}
```