

Avaliação Técnica

Python Developer

Solidaridad Network

Julho/2017

Recentemente, o Banco de Tangamandápio resolveu trocar os sistemas de seus caixas eletrônicos por uma interface web e você é responsável por escrever uma parte do backend. O novo sistema irá se conectar a uma API RESTful para obter as informações, enquanto toda a parte visual rodará em um navegador no caixa eletrônico.

O backend deve ser escrito em Python e disponibilizar alguns endpoints para que o caixa eletrônico possa se conectar. A lista deles é:

GET /conta/

Retorna uma lista de contas, cada uma com as seguintes informações:

id	ID da conta.
nome	Nome da conta.
saldo	Saldo da conta.

POST /conta/

Cria uma nova conta quando enviamos os seguintes dados:

nome	Nome da conta a ser criada.
saldo	Saldo da conta a ser criada.

Após a criação da conta, retorna as seguintes informações:

id	ID da conta criada.
nome	Nome da conta criada.
saldo	Saldo da conta criada.

GET /conta/{id}/

Retorna as seguintes informações da conta identificada por {id}:

nome	Nome da conta.
saldo	Saldo da conta.

POST /conta/{id}/saque/

Realiza um saque na conta identificada por {id} ao passarmos a seguinte informação:

valor	Valor do saque.
-------	-----------------

Retorna as seguintes informações relativas ao saque, além de devidamente alterar o estoque de cédulas do caixa eletrônico:

1	Quantidade de cédulas de 1 pataca do saque.
2	Quantidade de cédulas de 2 patacas do saque.
5	Quantidade de cédulas de 5 patacas do saque.
10	Quantidade de cédulas de 10 patacas do saque.
20	Quantidade de cédulas de 20 patacas do saque.
50	Quantidade de cédulas de 50 patacas do saque.
100	Quantidade de cédulas de 100 patacas do saque.

GET /caixa/

Retorna as seguintes informações do caixa eletrônico:

1	Quantidade de cédulas de 1 pataca.
2	Quantidade de cédulas de 2 patacas.
5	Quantidade de cédulas de 5 patacas.
10	Quantidade de cédulas de 10 patacas.
20	Quantidade de cédulas de 20 patacas.
50	Quantidade de cédulas de 50 patacas.
100	Quantidade de cédulas de 100 patacas.

POST /caixa/

Altera o estoque de cédulas do caixa eletrônico quando enviamos os seguintes dados:

1	Quantidade de cédulas de 1 pataca.
2	Quantidade de cédulas de 2 patacas.
5	Quantidade de cédulas de 5 patacas.
10	Quantidade de cédulas de 10 patacas.
20	Quantidade de cédulas de 20 patacas.
50	Quantidade de cédulas de 50 patacas.
100	Quantidade de cédulas de 100 patacas.

Retorna as seguintes informações do caixa eletrônico:

1	Quantidade de cédulas de 1 pataca.
2	Quantidade de cédulas de 2 patacas.
5	Quantidade de cédulas de 5 patacas.
10	Quantidade de cédulas de 10 patacas.
20	Quantidade de cédulas de 20 patacas.
50	Quantidade de cédulas de 50 patacas.
100	Quantidade de cédulas de 100 patacas.

Saques

Os saques devem ter como objetivo minimizar o número de cédulas entregues ao cliente, ou seja, as cédulas de valores maiores devem ser priorizadas. Vamos supor que o caixa eletrônico tenha o seguinte estoque:

Cédula	1	2	5	10	20	50	100
Qtidade	2	10	5	3	4	1	5

Se um cliente realizar um saque de 197 patacas, ele irá receber 6 cédulas:

- 1 cédula de 100;
- 1 cédula de 50;
- 2 cédulas de 20;
- 1 cédula de 5;
- 1 cédula de 2.

Mas repare que caso o estoque do caixa eletrônico seja:

Cédula	1	2	5	10	20	50	100
Qtidade	2	10	5	3	4	0	5

E o cliente realizar o mesmo saque de 197 patacas, ele irá receber 8 cédulas:

- 1 cédula de 100;
- 4 cédulas de 20;
- 1 cédula de 10;
- 1 cédula de 5;
- 1 cédula de 2.

Requisitos

1. Implementar os endpoints da API descrita acima em Python utilizando o framework Django;
2. Utilizar SQLite como base de dados, para ser portátil;
3. A API deve utilizar JSON (`Content-Type: application/json`) tanto na entrada de dados, quanto na saída;
4. Realizar os saques da maneira descrita, atualizando corretamente o saldo da conta e o estoque de cédulas do caixa eletrônico;
 - (a) O saque não pode ser realizado se não houver dinheiro suficiente na conta informada;
 - (b) O saque também não pode ser realizado se não houver cédulas suficientes no caixa eletrônico.

Dicas

- Utilize a interface disponível em `banco.html` para testar a sua API. Ela implementa os métodos descritos neste documento.
- Remova a verificação de tokens CSRF para que o Django aceite pedidos sem estes.
- Utilize o pacote `django-cors-headers` para permitir chamadas de qualquer origem. Ele adiciona o cabeçalho `Access-Control-Allow-Origin`.

Para se destacar

Os itens descritos abaixo **não são obrigatórios**, mas o esforço extra chamará nossa atenção:

1. Ao realizar um saque, apresentar uma outra possibilidade de combinação de cédulas. Esta funcionalidade existe hoje, por exemplo, nos caixas eletrônicos do Banco do Brasil.
2. Implementar um endpoint `/conta/{id}/extrato/`, que exiba o histórico de transações.
3. Implementar um endpoint `/conta/{id}/transferencia/`, que permita transferir dinheiro entre contas.

As funcionalidades acima não estão previstas na interface dada. Portanto, será necessário testar via `curl`, Postman¹ ou realizar alterações na mesma.

¹<https://www.getpostman.com/>