

GREEN BANK API

DOCS



S O B R E A A P I

GreenBankAPI é uma simples aplicação bancária. Dentro dela é possível registrar uma conta, realizar transações básicas e gerar relatórios baseados nessas transações.

Dentro dessa documentação você terá todas as informações necessárias para usar a API de forma correta.

REGISTRAR CONTA

.../api/register

Endpoint responsável por registrar as contas no banco.

Espera um JSON contendo três parâmetros:

```
{  
  "document": 92786448891,  
  "account_holder": "rafa@live.com",  
  "password": "1234567"  
}
```

document (int): CPF válido do titular da conta

account_holder (string): email do titular da conta

password (string): senha da conta de no mínimo 7 dígitos

RESPOSTAS

Retorna 201 caso a conta tenha sido criada corretamente

Retorna 422 caso algum erro aconteça

REGISTRAR CONTA

POST .../api/register

INFO

Não é possível criar mais de uma conta com o mesmo CPF.

O CPF precisa ser válido.

A senha precisa ter no mínimo 7 dígitos.

Ao se registrar o usuário começa com um saldo de R\$1000.

REALIZAR LOGIN

POST .../api/login

Endpoint responsável por logar um usuário.
Espera um JSON contendo dois parâmetros:

```
{  
  "document": 12151194884,  
  "password": "1234567"  
}
```

document (int): CPF válido do titular da conta

password (string): senha da conta de no mínimo 7 dígitos

RESPOSTAS

Retorna 202 caso o login tenha sido
realizado com sucesso

Retorna 401 caso algum erro aconteça

REALIZAR LOGIN

.../api/login

INFO

Se caso tenha sucesso, a requisição retornará um token (exemplo):

e5787f7fb83194@fd5ca59\$78

Ele é essencial para todas as requisições envolvendo transações e relatórios.

Por conta disso, é necessário salva-lo para utilizar em futuras requisições.

O token será invalidado após logout. Cada conta só pode ter um token válido por sessão.

Não é possível se logar duas vezes em dois dispositivos diferentes, é necessário dar logout antes de se logar novamente.

LOGOUT

GET .../api/logout

Endpoint responsável por deslogar um usuário. Espera o documento do usuário enviado como uma query string (exemplo):

```
/api/logout?document=43082810837
```

`document (int)`: CPF válido do titular da conta

RESPOSTAS

Retorna 200 caso o logout tenha sido realizado com sucesso

Retorna 417 caso o usuário não tiver uma sessão ativa

REALIZAR TRANSAÇÕES

POST .../api/logout

Endpoint responsável por realizar todas as transações, espera um POST request com um JSON contendo no mínimo 3 parâmetros para transações que envolvem somente uma conta:

```
{  
  "payer": 12151194884,  
  "transaction_type": "deposit",  
  "amount": "1500"  
}
```

payer (int): CPF da conta
amount(float): valor total da transação
transaction_type(string): tipo de transação

no momento apenas três tipos são suportadas:

"withdraw" para saques
"deposit" para depósitos
"wire_transfer" para transferência

REALIZAR TRANSAÇÕES

POST .../api/logout

Para a transação de tipo "wire_transfer" é necessário passar um quarto argumento para a requisição:

```
{  
  "payer": 12151194884,  
  "transaction_type": "deposit",  
  "amount": "1500",  
  "payee": 43082810837  
}
```

payee (int): CPF da conta destino

RESPOSTAS

Retorna 200 caso a transação tenha sido realizada com sucesso

Retorna 412 caso a transação tenha falhado.

REALIZAR TRANSAÇÕES

POST .../api/logout

TOKEN DE AUTENTICAÇÃO

Em toda requisição relacionada a este endpoint é necessário passar o token retornado ao logar como um parametro dentro do header da requisição.

No header basta adicionar uma key nomeada token e seu valor dentro dela.

Sem isso, as transações não serão autenticadas e por isso falharão (exemplo):

≡ token

e5787f7fb83194@fd5ca59\$78

GERAR RELATÓRIO

GET .../api/report

Assim como as transações, o endpoint report depende do token disponibilizado durante o login, insira-o como um parametro dentro do header da requisição para que os relatórios sejam gerados.

O relatório pode ser gerado baseado em três períodos diferentes:

Em um dia específico:

Para isso, basta passar o dia, o mês, o ano e o documento logado como query parameters. Exemplo:

```
http://localhost:4000/api/report?document=43082810837&year=2022&month=11&day=02
```

Em um mês específico:

Para isso, basta passar o mês, o ano e o documento logado como query parameters. Deixe o day como nil.

Exemplo:

```
http://localhost:4000/api/report?document=43082810837&year=2022&month=11&day=nil
```

Em um ano específico:

Para isso, basta passar apenas o ano e o documento como query parameters. Deixe o day e o mês como nil.

Exemplo:

```
http://localhost:4000/api/report?document=43082810837&year=2022&month=nil&day=nil
```

GERAR RELATÓRIO

GET .../api/report

Para gerar um relatório baseado em todas as transações realizadas no banco independente do período, basta passar o documento logado como query parameter e deixar todos os outros campos (month, day, year) como nil. Exemplo:

```
http://localhost:4000/api/report?document=43082810837&year=nil&month=nil&day=nil
```

RESPOSTAS

Retorna 201 e o valor total transacionado caso o relatório tenha sido gerado com sucesso.