

 Esta documentação ainda não foi publicada, esta é apenas uma prévia. As alteraçõ...

# Labebank - Sistema Bancário

Sistema simplificado, gerenciamento de usuários no banco, com contas bancárias flexíveis do sistema

EndFragment

EndFragment

## PEGUE testar

http://localhost:3003/teste

Verificação se Apl está funcionando

Solicitação de exemplo

testar

```
curl --location --request GET 'http://localhost:3003/teste'
```

## PEGUE Pegar saldo

http://localhost:3003/users/:id

StartFragment

Rota de requisição: "/users/:id"

Entradas → id do usuário selecionado.

Validação de Entrada → nenhuma

Regras de Negócio → id deve existir na lista de contas cadastradas

Saídas possíveis:

- Cada erro deve retornar o seu código de status de respeito e uma mensagem descrevendo a situação.

- Para sucesso, deve retornar o status de exibição e o saldo do usuário selecionado

 Esta documentação ainda não foi publicada, esta é apenas uma prévia. As alteraçõ...

Eu iria

2

Solicitação de exemplo

Pegar saldo

```
curl --location --request GET 'http://localhost:3003/users/2'
```

## COLOCAR Adicionar Saldo

http://localhost:3003/users/:id

StartFragment

Rota de requisição: "/users/:id"

Entradas → id do usuário selecionado e novo saldo a ser adicionado.

Validação de Entrada → Aumento do saldo deve ser do tipo number e maior que 0

Regras de Negócio → id deve existir na lista de contas cadastradas

Saídas possíveis:

- Cada erro deve retornar o seu código de status de respeito e uma mensagem descrevendo a situação.
- Para sucesso, deve retornar o status de adição, uma mensagem de sucesso e os dados do usuário selecionado e atualizado.

EndFragment

### PATH VARIABLES

id

2

BODY raw

```
{  
  "saldo":1000
```

 Esta documentação ainda não foi publicada, esta é apenas uma prévia. As alteraçõ...

Example Request

Adicionar Saldo

```
curl --location --request PUT 'http://localhost:3003/users/2' \  
--data-raw '{  
  "saldo":1000  
}'
```

## PUT Pagar conta

http://localhost:3003/users/:id/pay

StartFragment

Rota de requisição: "/users/:id"

Entradas → id do usuário selecionado e novo saldo a ser adicionado.

Validação de Input → Aumento do saldo deve ser do tipo number e maior que 0

Regras de Negócio → id deve existir na lista de contas cadastradas

Saídas possíveis:

- Cada erro deve retornar o seu respectivo status code e uma mensagem descrevendo a situação.
- Para sucesso, deve retornar o status de adição, uma mensagem de sucesso e os dados do usuário selecionado e atualizado.


### PATH VARIABLES

id

2

### BODY raw

```
{  
  "valorDaConta": 1000.00,  
  "descricaoContaPagar": "Imposto"  
}
```

 Esta documentação ainda não foi publicada, esta é apenas uma prévia. As alteraçõ...

```
{
  "valorDaConta": 1000.00,
  "descricaoContaPagar": "Imposto"
}
```

## POST Criar conta

http://localhost:3003/users

StartFragment

Entradas → Nome do usuário, CPF e data de nascimento.

Validação de Input:

- nome, CPF e data de nascimento devem existir.
- Todas as entradas devem ser do tipo string.

Regras de Negócio:

- CPF é único para cada usuário.
- O usuário deve ter idade maior ou igual a 18 anos (não se preocupa quanto a variações no dia e mês).
- O nome do usuário deve ter ao menos 3 caracteres.
- a id é ativada automaticamente
- a lista de extrato deve iniciar vazia
- o saldo deve iniciar com o valor 0

Saídas possíveis:

- Cada erro deve retornar o seu código de status de respeito e uma mensagem descrevendo a situação.
- Para sucesso, deve retornar o status de criação, mensagem de sucesso e lista de usuários atualizados.

EndFragment

CORPO cru

```
{
  "nome": "Aderval",
  "cpf": "11144488896",
  "aniversario": "10/11/95"
```

 Esta documentação ainda não foi publicada, esta é apenas uma prévia. As alteraçõ...

Solicitação de exemplo

Criar conta

```
curl --location --request POST 'http://localhost:3003/users' \  
--data-raw '{  
  "nome": "Aderval",  
  "cpf": "11144488896",  
  "aniversario": "10/11/95"
```