

Instruções para testes funcionais

Criação de uma nova pessoa

É possível criar uma nova pessoa através da API Rest.

Path:

POST /api/pessoa

Request:

```
{
  "nome": "Novo Usuario",
  "cpf": "98767856490",
  "dataNascimento": "12/12/2012"
}
```

Response: HTTP/200

```
{
  "idPessoa": 4,
  "nome": "Novo Usuario",
  "cpf": "98767856490",
  "dataNascimento": "12/12/2012"
}
```

Criação de uma nova Conta

Para criar uma conta é necessário ter uma pessoa cadastrada, pois uma conta sempre pertencerá a uma pessoa.

path:

POST /api/conta

request:

```
{
  "idPessoa": 4,
  "saldo": 100.00,
  "limiteSaqueDiario": 100.00,
  "ativa": true
}
```

Response: HTTP/200

```
{
  "idConta": 3,
  "idPessoa": 4,
  "saldo": 100,
  "limiteSaqueDiario": 100,
  "ativa": true
}
```

Response: HTTP/500

caso o usuário tente criar uma nova conta sem uma pessoa válida.

```
{
  "status": "INTERNAL_SERVER_ERROR",
  "message": "Pessoa inexistente no banco de dados",
  "errors": []
}
```

Realização de operações na Conta

Para realizar as operações na conta, eu criei um request com atributos que são preenchidos de acordo com a necessidade dependendo da operação desejada pelo usuário.

Request para operações na conta:

```
{
  "idConta": int,
  "valor": BigDecimal,
  "dataInicio": Date,
  "dataFim": Date,
}
```

As datas sempre devem ser enviadas no formato **"19/09/2018"** padrão usado por nós. Os campos de data no request, só serão utilizados na requisição de extrato por período.

Response de operações na conta

No fim de toda operação com sucesso na conta, a API retorna o extrato diário com as informações da conta e as transações realizadas no dia.

Depósito

path:

POST /api/conta/deposito

request:

```
{
  "idConta": 3,
  "valor": 100
}
```

response: HTTP/200

```
{
  "conta": {
    "idConta": 2,
    "idPessoa": 2,
    "saldo": 600,
    "limiteSaqueDiario": 100,
    "ativa": true,
    "dataCriacao": "2018-09-19T21:04:55.000Z"
  },
  "extrato": [
    {
      "idTransacao": 11,
      "idConta": 2,
      "valor": 100,
      "tipoOperacao": "D",
      "dataTransacao": "2018-09-20T02:45:55.912Z"
    }
  ]
}
```

Saque

path:

POST /api/conta/saque

request:

```
{
  "idConta": 3,
  "valor": 100
}
```

response: HTTP/200

```
{
  "conta": {
    "idConta": 2,
    "idPessoa": 2,
    "saldo": 600,
    "limiteSaqueDiario": 100,
    "ativa": true,
    "dataCriacao": "2018-09-19T21:04:55.000Z"
  },
  "extrato": [
    {
      "idTransacao": 11,
      "idConta": 2,
      "valor": 100,
      "tipoOperacao": "S",
      "dataTransacao": "2018-09-20T02:45:55.912Z"
    }
  ]
}
```

Bloqueio / Desbloqueio da Conta

path:

POST /api/conta/bloqueio

POST /api/conta/desbloqueio

request:

```
{
  "idConta":3
}
```

response: HTTP/200

```
{
  "conta": {
    "idConta": 3,
    "idPessoa": 4,
    "saldo": 100,
    "limiteSaqueDiario": 100,
    "ativa": true or false,
    "dataCriacao": "2018-09-19T22:24:24.000Z"
  }
}
```

Extrato

Se não for passado a data, ele retorna sempre o extrato do dia, caso seja passado a dataInicial e dataFinal é feito uma busca por período.

path:

POST /api/conta/extrato

request:

```
//extrato diario
{
  "idConta":3
}
```

ou

```
//extrato por periodo
{
  "idConta":2,
  "dataInicio": "19/09/2018",
```

```
"dataFim": "20/09/2018"
}
```

response

```
{
  "conta": {
    "idConta": 3,
    "idPessoa": 4,
    "saldo": 100,
    "limiteSaqueDiario": 100,
    "ativa": true or false,
    "dataCriacao": "2018-09-19T22:24:24.000Z"
  }
}
```

Pegar apenas as informações cadastrais da Conta

path

```
GET /api/conta/{idConta}
```

response

```
{
  "idConta": 2,
  "idPessoa": 2,
  "saldo": 600,
  "limiteSaqueDiario": 100,
  "ativa": true,
  "dataCriacao": "2018-09-19T21:04:55.000Z"
}
```

Responses : HTTP/500

Possíveis erros durante a realização de operações na conta

```
{
  "status": "INTERNAL_SERVER_ERROR",
  "message": "Conta inválida ou inexistente",
  "errors": []
}
```

```
{
  "status": "INTERNAL_SERVER_ERROR",
  "message": "Saldo Insuficiente",
  "errors": []
}
{
  "status": "INTERNAL_SERVER_ERROR",
  "message": "O limite de saque diário foi excedido",
  "errors": []
}
{
  "status": "INTERNAL_SERVER_ERROR",
  "message": "A conta encontra-se bloqueada",
  "errors": []
}
{
  "status": "INTERNAL_SERVER_ERROR",
  "message": "O valor passado na operação é inválido",
  "errors": []
}
```