

Documentação da API de Gastos Residenciais

Visão Geral do Projeto

A API de Gastos Residenciais é um serviço desenvolvido em **Spring Boot** que permite o **cadastro e gerenciamento de pessoas** e suas **respectivas transações financeiras** (despesas e receitas). O objetivo é facilitar o **controle financeiro** permitindo que os usuários registrem e visualizem suas movimentações de forma organizada.

Estrutura do Projeto

O projeto utiliza **Maven** para gerenciamento de dependências e possui o seguinte arquivo `pom.xml`:

Dependências Utilizadas:

- **spring-boot-starter-data-jpa**: Permite integração com bancos de dados usando JPA.
- **spring-boot-starter-web**: Fornece funcionalidades para criação de APIs REST.
- **spring-boot-devtools**: Ferramenta para recarregamento automático no desenvolvimento.
- **h2**: Banco de dados em memória usado para testes e desenvolvimento.
- **spring-boot-starter-test**: Inclui bibliotecas como JUnit e Mockito para testes automatizados.

Configurações do Projeto:

- **Java Version**: 21
 - **Group ID**: `com.example`
 - **Artifact ID**: `DesafioMaxiprod`
 - **Version**: `0.0.1-SNAPSHOT`
 - **Name**: `DesafioMaxiprod`
 - **Description**: `Demo project for Spring Boot`
-

Testes com Postman

A API pode ser testada utilizando **Postman**, uma ferramenta para desenvolvimento e testes de APIs. Com ela, é possível enviar requisições HTTP para os endpoints da API e visualizar as respostas.

Classes do Domínio

1. CadPessoas

Representa uma pessoa no sistema e possui os seguintes atributos:

- **id**: Identificador único.
- **nome**: Nome da pessoa.
- **idade**: Idade da pessoa.

Possui **métodos getters e setters** para acesso e modificação dos atributos.

2. CadTransacoes

Representa uma transação financeira (despesa ou receita) com os atributos:

- **idTransacoes**: Identificador único da transação.
- **descricao**: Descrição da transação.
- **valor**: Valor da transação.
- **data**: Data da transação.
- **tipo**: Tipo da transação (**RECEITA** ou **DESPESA**).
- **pessoas**: Referência para a pessoa associada.

3. Tipo (Enumeração)

Define os tipos de transações disponíveis:

- **RECEITA**: Entrada de dinheiro.
 - **DESPESA**: Saída de dinheiro.
-

Repositórios

1. CadPessoasRepository

Interface responsável pela persistência e recuperação de dados da entidade **CadPessoas**. Ela estende `JpaRepository<CadPessoas, Integer>`, fornecendo **operações CRUD automáticas**.

2. CadTransacoesRepository

Repositório da entidade **CadTransacoes**, que também estende `JpaRepository<CadTransacoes, Integer>`.

DTOs (Data Transfer Objects)

1. CadPessoasResumoDto

Utilizado para exibir um resumo das informações financeiras de uma pessoa.

Atributos:

- **id**: Identificador da pessoa.
- **nome**: Nome da pessoa.
- **idade**: Idade da pessoa.
- **totalReceita**: Soma de todas as receitas associadas.
- **totalDespesa**: Soma de todas as despesas associadas.
- **saldo**: Cálculo de **totalReceita - totalDespesa**.

Construtor

Recebe um objeto `CadPessoas`, os totais de receita e despesa, e calcula o saldo automaticamente.

2. CadPessoasDto

Representa as informações básicas de uma pessoa para transferência de dados.

Atributos:

- **id**: Identificador da pessoa.
- **nome**: Nome da pessoa.
- **idade**: Idade da pessoa.

Construtores

- **Construtor com CadPessoas**: Converte um objeto `CadPessoas` em DTO.

- **Construtor com parâmetros:** Permite criar um DTO diretamente.
- **Construtor padrão:** Para criação vazia.

3. CadTransacoesDto

DTO utilizado para representar uma transação financeira.

Atributos:

- **idTransacoes:** Identificador da transação.
- **descricao:** Descrição da transação.
- **valor:** Valor da transação.
- **tipo:** Tipo (**RECEITA** ou **DESPESA**).
- **idPessoa:** ID da pessoa associada.

Construtores

- **Construtor com CadTransacoes:** Converte um objeto `CadTransacoes` em DTO.
 - **Construtor com parâmetros:** Permite criar um DTO diretamente.
 - **Construtor padrão:** Para criação vazia.
-

1. Endpoints da API

1.1. Pessoas

Método	Endpoint	Descrição
GET	<code>/cadastro pessoas</code>	Retorna todas as pessoas cadastradas.
GET	<code>/cadastro pessoas/{id}</code>	Retorna uma pessoa pelo ID.
POST	<code>/cadastro pessoas</code>	Cadastra uma nova pessoa.
DELETE	<code>/cadastro pessoas/{id}</code>	Remove uma pessoa pelo ID.

1.2. Transações

Método	Endpoint	Descrição
GET	<code>/cadastrotransacoes</code>	Retorna todas as transações cadastradas.
GET	<code>/cadastrotransacoes/{id}</code>	Retorna uma transação pelo ID.
POST	<code>/cadastrotransacoes</code>	Cadastra uma nova transação.
DELETE	<code>/cadastrotransacoes/{id}</code>	Remove uma transação pelo ID.

CadPessoasService

Descrição

Serviço responsável pelo gerenciamento de pessoas.

Endpoints

Criar uma nova pessoa

- URL: `/cadastropessoas`
- Método: `POST`

Corpo da Requisição:

```
{  
  "nome": "string",  
  "idade": "integer"  
}
```

-
- Resposta: `201 Created`

Buscar uma pessoa por ID

- URL: `/cadastropessoas/{id}`
- Método: GET

Resposta:

```
{  
  "id": "integer",  
  "nome": "string",  
  "idade": "integer"  
}
```

•

Listar todas as pessoas

- URL: `/cadastropessoas`
- Método: GET

Resposta:

```
[  
  {  
    "id": "integer",  
    "nome": "string",  
    "idade": "integer"  
  }  
]
```

•

Atualizar uma pessoa

- URL: `/cadastropessoas/{id}`
- Método: PUT

Corpo da Requisição:

```
{  
  "nome": "string",  
  "idade": "integer"  
}
```

•

- Resposta: 204 No Content

Deletar uma pessoa

- URL: `/cadastropessoas/{id}`

- **Método:** DELETE
 - **Resposta:** 204 No Content
-

CadTransacoesService

Descrição

Serviço responsável pelo gerenciamento de transações financeiras.

Endpoints

Criar uma nova transação

- **URL:** /cadastrotransacoes
- **Método:** POST

Corpo da Requisição:

```
{  
  "descricao": "string",  
  "valor": "double",  
  "tipo": "RECEITA" | "DESPESA",  
  "idPessoa": "integer"  
}
```

-
- **Resposta:** 201 Created

Buscar uma transação por ID

- **URL:** /cadastrotransacoes/{id}
- **Método:** GET

Resposta:

```
{  
  "idTransacoes": "integer",  
  "descricao": "string",  
  "valor": "double",  
  "tipo": "RECEITA" | "DESPESA",  
  "pessoas": {  
    "id": "integer",  
    "nome": "string",  
    "idade": "integer"  
  }  
}
```

```
}  
}
```

-

Listar todas as transações

- URL: `/cadastrotransacoes`
- Método: `GET`

Resposta:

```
[  
  {  
    "idTransacoes": "integer",  
    "descricao": "string",  
    "valor": "double",  
    "tipo": "RECEITA" | "DESPESA",  
    "pessoas": {  
      "id": "integer",  
      "nome": "string",  
      "idade": "integer"  
    }  
  }  
]
```

Atualizar uma transação

- URL: `/cadastrotransacoes/{id}`
- Método: `PUT`

Corpo da Requisição:

```
{  
  "descricao": "string",  
  "valor": "double",  
  "tipo": "RECEITA" | "DESPESA",  
  "idPessoa": "integer"  
}
```

-

- Resposta: `204 No Content`

Deletar uma transação

- **URL:** /cadastrotransacoes/{id}
- **Método:** DELETE
- **Resposta:** 204 No Content