Centro Universitário Estácio Juiz de Fora - campus Rio Branco

Análise e Desenvolvimento de Sistemas – Turma 3001

Projeto de Desenvolvimento Rápido de Aplicações em Python:

Documentação da API

Trabalho apresentado ao professor Anderson Barboza da Cruz, na disciplina Desenvolvimento Web, pelos(as) alunos(as), Davi Rodrigues, Gustavo Rodrigues, Jamil Salomão, Tarcísio Alves

Juiz de Fora, 09 de Outubro de 2024

Sumário

1. Introdução

O objetivo dessa documentação é principalmente demonstrar como utilizar a API e fornecer uma visão geral das rotas disponíveis para o sistema do projeto, métodos HTTP, necessidade de autenticação, formatos de corpo de requisição, com possíveis respostas.

2. Endpoints

Primeiramente para utilizar, inicie o servidor Flask pelo arquivo app.js no source do projeto, após isso você receberá uma mensagem semelhante a essa logo abaixo indicando que o servidor está pronto:

Server is running on http://localhost:3001

LEMBRETE: Para utilizar baixe o aplicativo Postman ou Insomnia.

Postman: https://www.postman.com/downloads/
Insomnia: https://insomnia.rest/download

Autenticação:

- A maioria dos endpoints requer que o usuário esteja autenticado. A autenticação é feita usando JWT.

Headers:

- `Content-Type: application/json`

2.1. Rotas de Usuário

1. Criar Usuário:

- **URL:** /api/create-user
- **Método:** POST
- **Corpo:**

  {
  "email": "teste@gmail.com",
  "password": "teste123",
  "name": "Arnaldo",
  "profession": "Analista",
  "monthlyIncome": 1220.0
  }
- **Resposta:**

  - 201 Ok - Usuário criado com sucesso;

  - 400 Solicitação Inválida – Campos obrigatórios ausentes;

  - 500 Erro Interno do Servidor – Erro ao criar usuário.

2. Login do Usuário:

- **URL:** /api/create-user
- **Método:** POST
- **Corpo:**

  {
  "email": "teste@gmail.com",
  "password": "teste123",
  "name": "Arnaldo",
  "monthlyIncome": 1220.0
  }
- **Resposta:**

  - 200 OK – Usuário logado com sucesso;

  - 400 Solicitação Inválida – Campos obrigatórios ausentes;

  - 500 Erro Interno do Servidor - Mensagem de erro.

3. Logout de Usuário:

- **URL:** /api/logout
- **Método:** POST
- **Autenticação:** Requerida
- **Resposta:**

  - 200 OK – Usuário deslogado com sucesso;

  - 500 Erro Interno do Servidor - Mensagem de erro.

4. Atualizar Dados do Usuário:

- **URL:** /api/update-user
- **Método:** PUT
- **Autenticação:** Requerida
- **Corpo:**

```
{
  "name": "Arnaldo",
  "email": "teste@gmail.com",
  "profession": "Analista",
  "monthlyIncome": 1450.0
}
```

- **Resposta:**

  - 200 OK - Dados do usuário atualizados com sucesso;

  - 400 Solicitação Inválida - Campos obrigatórios ausentes;

  - 500 Erro Interno do Servidor - Mensagem de erro.

5. Atualizar Senha do Usuário:

- **URL:** /api/reset-password
- **Método:** POST
- **Corpo:**

  {

  "email": " teste@gmail.com "

  }
- **Resposta:**

  - 200 OK – E-mail para troca de senha enviado com sucesso;

  - 400 Solicitação Inválida - Campos obrigatórios ausentes;

  - 500 Erro Interno do Servidor - Mensagem de erro.

2.2. Rotas de Categoria de Gastos

1. Criar Categoria de Gastos:

- **URL:** /api/create-financial-payments
- **Método:** POST
- **Autenticação:** Requerida
- **Corpo:**

  {

  "type_account": "Contas de Lazer",

  "meta": 400

  }
- **Resposta:**

  - 201 OK – Categoria de gastos criado com sucesso.

  - 400 Solicitação Inválida - Campos obrigatórios ausentes.

  - 500 Erro Interno do Servidor - Mensagem de erro.

2. Retornar Lista de Gasto do Usuário

- **URL:** /api/get-financial-payments
- **Método:** GET
- **Autenticação:** Requerida
- **Resposta:**

  - 200 OK

  - 500 Erro Interno do Servidor - Mensagem de erro.

3. Renomear Categoria de Gasto:

- **URL:** /api/rename-type-account
- **Método:** PUT
- **Autenticação:** Requerida
- **Corpo:**

  {

  "type_account": "Conta Corrente",

  "new_type_account": "Conta Corrente 2"

  }
- **Resposta:**

  - 200 OK – Categoria de Gasto renomeada com sucesso.

  - 400 Solicitação Inválida - Campos obrigatórios ausentes.

  - 500 Erro Interno do Servidor - Mensagem de erro.

4. Atualizar Categoria de Gasto:

- **URL:** /api/update-meta-type-account
- **Método:** PUT
- **Autenticação:** Requerida
- **Corpo:**

  {

  "type_account": "Conta Corrente",

  "new_meta": 312.00

  }
- **Resposta:**

  - 200 OK – Meta de Gasto atualizada com sucesso

  - 400 Solicitação Inválida - Campos obrigatórios ausentes.

  - 500 Erro Interno do Servidor - Mensagem de erro.

5. Deletar Categoria de Gasto:

- **URL:** /api/delete-type-account
- **Método:** DELETE
- **Autenticação:** Requerida
- **Corpo:**

  {

  "type_account": "Contas de Lazer"

  }
- **Resposta:**

  - 200 OK – Categoria de Gasto e todas as suas contas deletadas com sucesso

  - 400 Solicitação Inválida - Campos obrigatórios ausentes.

  - 500 Erro Interno do Servidor - Mensagem de erro.

2.3 Rotas de Lógica das Categorias de Gastos

1. Soma Total das Categorias de Gasto:

- **URL:** /api/sum-total-type-account-payments
- **Método:** GET
- **Autenticação:** Requerida
- **Resposta:**

  - **200 OK** – Total por Categoria de gasto recuperado com sucesso.

  - **500 Erro Interno do Servidor** - Mensagem de erro.

2. Soma do Saldo Total:

- **URL:** /api/sum-total-balance
- **Método:** GET
- **Autenticação:** Requerida
- **Resposta:**

  - **200 OK –** Saldo Total recuperado com sucesso.

  - **500 Erro Interno do Servidor** - Mensagem de erro.

3. Verificar Tipo de Conta Não Paga:

- **URL:** /api/verify-type-account-not-pay
- **Método:** GET
- **Autenticação:** Requerida
- **Resposta:**

  - **200 OK** – Tipo de Conta sem pagamento recuperada com sucesso.

  - **500 Erro Interno do Servidor** - Mensagem de erro.

4. Verificar Data de Pagamento das Contas:

- **URL:** /api/verify-accounts-payment-date
- **Método:** GET
- **Autenticação:** Requerida
- **Resposta:**

  - **200 OK** – Data de Pagamento da conta recuperada com sucesso.

  - **500 Erro Interno do Servidor** - Mensagem de erro.

5. Todos os Pagamentos por Tipo de Conta:

- **URL:** /api/all-payments-per-type-account
- **Método:** GET
- **Autenticação:** Requerida
- **Resposta:**

  - 200 OK – Todos os Pagamentos por tipo conta recuperados com sucesso.

  - 500 Erro Interno do Servidor - Mensagem de erro.

2.4 Rotas de Pagamentos

1. Criar Pagamento:

- **URL:** /api/create-financial-accounts
- **Método:** POST
- **Autenticação:** Requerida
- **Corpo:**

```
{
  "type_account": "Conta Corrente 2",
   "name_account": "Luz",
   "description": "Não posso esquecer de pagar",
   "status": false,
   "priority": "Alta",
   "paymentDate": "2024-09-08",
   "price": 322.22
}
```

- **Resposta:**

  - 201 OK – Pagamento criado com sucesso.

  - 400 Solicitação Inválida - Campos obrigatórios ausentes.

  - 500 Erro Interno do Servidor - Mensagem de erro.

2. Atualizar Pagamento:

- **URL:** /api/update-financial-accounts
- **Método:** PUT
- **Autenticação:** Requerida
- **Corpo:**

```
{
  "type_account": "Conta Corrente 2",
  "name_account": "Luz",
  "description": "Não posso esquecer de pagar",
  "status": false,
  "priority": "Baixa",
  "paymentDate": "2024-09-09",
  "price": 402.40
}
```

- **Resposta:**

  - 200 OK – Pagamento atualizado com sucesso.

  - 400 Solicitação Inválida - Campos obrigatórios ausentes.

  - 500 Erro Interno do Servidor - Mensagem de erro.

3. Deletar Pagamento:

- **URL:** /api/delete-financial-accounts
- **Método:** DELETE
- **Autenticação:** Requerida
- **Corpo:**

  {

  "type_account": "Conta Corrente 2",

  "name_account": "Luz"

  }
- **Resposta:**

  - 200 OK – Pagamento deletado com sucesso.

  - 400 Solicitação Inválida - Campos obrigatórios ausentes.

  - 500 Erro Interno do Servidor - Mensagem de erro.

3. Observações finais

Algumas observações finais vão para as opções do CORS, github do projeto e alguns sites de ajuda para Methods HTTP e Status HTTP:

CORS Options:

As permissões de CORS estão configuradas para permitir solicitações do http://localhost:3000 (pode ser mudado para outras origens) com credenciais:

CORS(app, resources={ r"/" : { "origins" : "" }})

Projeto no GitHub:

Para acessar o repositório do projeto no GitHub, visite Project-Financial-Management.

Mais Informações:

Para mais informações sobre os status HTTP, consulte MDN HTTP Status.

Para mais informações sobre os métodos HTTP, consulte MDN HTTP Methods.