Projeto Backend Framework

Sistema de Gerenciamento de Agência de Intercâmbio (SGAI)

Gabriel Galvão dos Anjos 01563414 João
Victor Freire de Souza 01525280 Juliana
Camilo de Barros 01532960 Leonardo
Nadolny Magalhães 01559557
Matheus Vitor Fernandes Moreira 01523037
Claudio Henrique Maia Soares de Souza 24010216
Talisson Marcelo Ludugero Silva 01360881

Recife - PE

Título

Sistema de Gerenciamento de Agência de Intercâmbio (SGAI)

Descrição

O Sistema de Gerenciamento de Agência de Intercâmbio (SGAI) é uma API desenvolvida para facilitar e otimizar o processo de gerenciamento de alunos e a conexão com instituições de ensino no exterior. Este projeto tem como objetivo fornecer uma solução eficiente para agências de intercâmbio, permitindo-lhes acompanhar, administrar e aprimorar a experiência de estudantes que desejam estudar em instituições de ensino em países estrangeiros. A API SGAI oferece um conjunto abrangente de recursos para simplificar tarefas administrativas e melhorar a comunicação entre agências de intercâmbio, alunos e instituições de ensino no exterior.

Objetivos

- Automatizar o registro e a gestão de alunos interessados em programas de intercâmbio.
- Facilitar a busca e seleção de programas de estudo oferecidos por instituições estrangeiras.
- Permitir o registro e consulta de instituições de ensino no exterior, incluindo informações como nome, localização e cursos oferecidos.
- Fornecer comunicação eficiente entre agências de intercâmbio, alunos e instituições de ensino.

Escopo

No escopo inicial do projeto, a API SGAI incluirá as seguintes funcionalidades:

- Implementação de pelo menos uma classe de modelagem para representar entidades relevantes, como Aluno, Programa de Estudo e Instituição de Ensino.
- Desenvolvimento de pelo menos uma classe de repositório para acesso aos dados do banco de dados.
- Criação de pelo menos uma classe de controle para lidar com as operações da API.
- Implementação de uma suíte de testes para as funcionalidades entregues até o momento, utilizando Postman.

Este escopo inicial estabelece as bases da API, permitindo o gerenciamento de alunos, programas de estudo e instituições de ensino. À medida que o projeto avança, novas funcionalidades e melhorias podem ser adicionadas para expandir o escopo e atender a requisitos adicionais.

Público-Alvo

O público-alvo da API SGAI inclui:

- Agências de Intercâmbio: Gerentes e equipes administrativas de agências de intercâmbio que utilizam o sistema para gerenciar alunos e processos de intercâmbio.
- Estudantes: Alunos que desejam participar de programas de estudo no exterior e usam a API para se inscrever, acompanhar seu progresso e se comunicar com a agência.
- Instituições de Ensino no Exterior: Universidades, escolas de idiomas e outras instituições de ensino que colaboram com agências de intercâmbio e utilizam a API para receber inscrições de alunos.

Tecnologias Utilizadas

A criação de uma API robusta e eficiente para gerenciar uma agência de intercâmbio é um desafio empolgante que requer a escolha criteriosa das tecnologias certas. Neste projeto de backend, utilizaremos um conjunto de tecnologias modernas e poderosas para atender às necessidades específicas de uma agência de intercâmbio e garantir uma experiência de usuário fluida e confiável. A seguir, destacamos as principais tecnologias que serão empregadas na construção da nossa API:

1. **Python:** A linguagem de programação escolhida para o desenvolvimento da API é o Python. Python é amplamente conhecido por sua simplicidade, legibilidade e rica comunidade de desenvolvedores. Ele é uma escolha popular para a criação de APIs devido à sua facilidade de uso e à grande quantidade de bibliotecas e frameworks disponíveis.
2. **Flask:** Utilizaremos o Flask como o framework web para construir nossa API. O Flask é conhecido por sua simplicidade e flexibilidade, tornando-o ideal para o desenvolvimento de APIs RESTful. Ele oferece recursos essenciais para lidar com rotas, solicitações HTTP e respostas, facilitando a criação de endpoints para nossa API.
3. **SQLite:** Para o armazenamento de dados, optamos pelo SQLite, um banco de dados leve e

incorporado. SQLite é uma escolha adequada para projetos de pequena a média escala e é particularmente útil para desenvolvimento rápido de protótipos. Ele permitirá que armazenemos e gerenciemos informações relacionadas a alunos, programas de estudo e outras entidades essenciais para a agência de intercâmbio.

Em resumo, nossa escolha de tecnologias é guiada pela necessidade de construir uma API eficiente, segura e escalável para atender às demandas de gerenciamento de uma agência de intercâmbio. A combinação de Python, Flask, SQLite e outras tecnologias permitirá que nossa API ofereça uma plataforma confiável para conectar agências, alunos e instituições de ensino no exterior de forma eficaz e eficiente.

Arquitetura

A arquitetura de uma API desempenha um papel fundamental na sua eficiência, escalabilidade e capacidade de atender às necessidades dos usuários. Para o nosso projeto de Gerenciamento de Agência de Intercâmbio, optamos por uma abordagem RESTful (Representational State Transfer), uma arquitetura amplamente adotada para o desenvolvimento de APIs web que se baseia em princípios simples e poderosos. Abaixo, fornecemos uma visão geral da arquitetura da nossa API:

- **Arquitetura RESTful:** A API SGAI segue a arquitetura RESTful, que se baseia em princípios como recursos, URIs (Uniform Resource Identifiers), métodos HTTP e representações de recursos. Isso significa que cada entidade da nossa API, como alunos e programas de estudo, é tratada como um recurso identificável por meio de URIs únicas.

- **Recursos:** Os recursos são as entidades que nossa API manipula, como alunos, programas de estudo e instituições de ensino. Cada recurso é identificado por uma URI específica, como */alunos* para alunos e */programas* para programas de estudo.

- **Métodos HTTP:** A API utiliza os métodos HTTP padrão, como GET, POST, PUT e DELETE, para executar operações nos recursos. Por exemplo, o método GET é usado para recuperar informações sobre um aluno específico, enquanto o método POST é usado para criar um novo aluno.

- **Estado Stateless:** Uma característica fundamental do REST é que ele é stateless, o que significa que cada solicitação de um cliente para o servidor deve conter todas as informações necessárias para entender e processar a solicitação. Isso simplifica a escalabilidade e a manutenção da API.

- **Representações de Recursos:** As representações dos recursos são formatos de dados, como JSON ou XML, que são usados para transmitir informações entre o cliente e o servidor. A API SGAI utiliza JSON como formato padrão para representar dados, tornando as respostas da API facilmente legíveis e compatíveis com a maioria das linguagens de programação.

Em resumo, a arquitetura RESTful é a espinha dorsal da nossa API, proporcionando uma base sólida e eficiente para conectar agências, alunos e instituições de ensino no exterior. A organização do código por meio de classes aprimora a manutenção do projeto, garantindo que ele possa evoluir com facilidade à medida que novas funcionalidades são adicionadas para atender às demandas em constante evolução do nosso sistema de gerenciamento de intercâmbio educacional.

Rotas e Endpoints

Nossa API de Gerenciamento de Agência de Intercâmbio é projetada para oferecer um conjunto completo de funcionalidades para interagir com alunos, programas de estudo e instituições de ensino no exterior. Abaixo, apresentamos uma descrição detalhada de cada rota e endpoint da API, incluindo os métodos HTTP suportados e as operações executadas em cada endpoint:

1. `GET /alunos`
    a. **Método HTTP:** GET
    b. **Descrição:** Esta rota permite a recuperação de todos os alunos cadastrados no sistema.
    c. **Funcionalidade:** Retorna uma lista de todos os alunos registrados na agência de intercâmbio

2. `POST /alunos`
    a. **Método HTTP:** POST
    b. **Descrição:** Essa rota permite a criação de um novo aluno no sistema.
    c. **Funcionalidade:** Cria um novo registro de aluno (ID) com base nos dados fornecidos no corpo da requisição (nome, origem, destino).

3. `GET /aluno/<int:id>`
    a. **Método HTTP:** GET
    b. **Descrição:** Essa rota permite a consulta de um aluno específico com base em seu ID.
    c. **Funcionalidade:** Retorna os detalhes de um aluno específico, identificado pelo ID fornecido na URL.

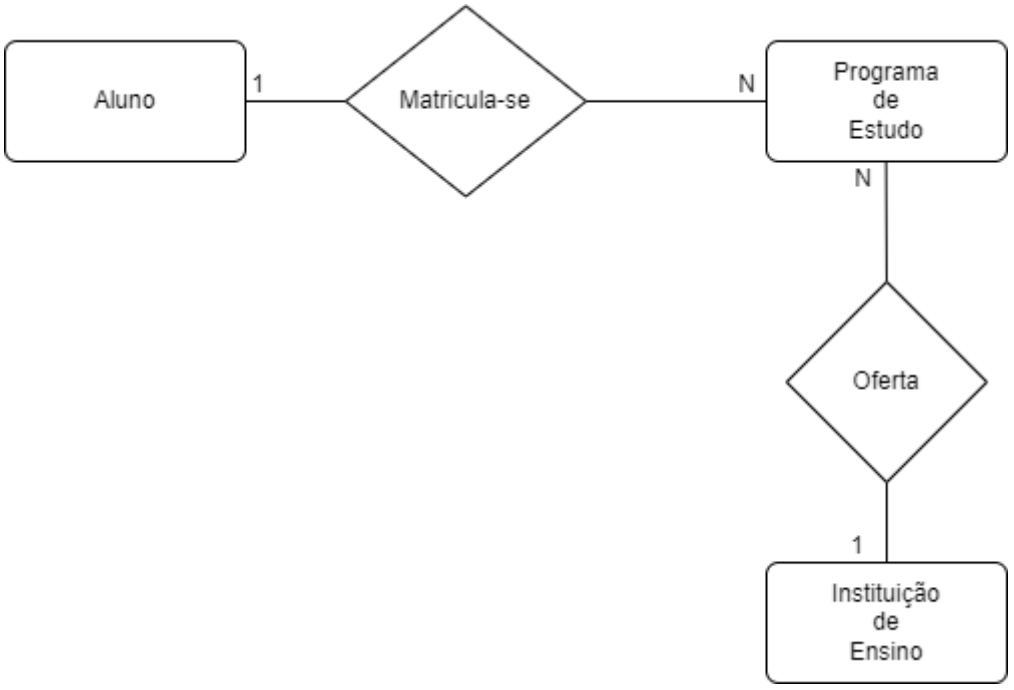
4. `PUT /aluno/<int:id>`
    a. **Método HTTP:** PUT
    b. **Descrição:** Essa rota permite a atualização dos dados de um aluno específico.
    c. **Funcionalidade:** Atualiza as informações de um aluno com base no ID fornecido na URL da requisição, substituindo os dados existentes pelos novos dados fornecidos na solicitação.

5. `DELETE /aluno/<int:id>`
    a. **Método HTTP:** DELETE
    b. **Descrição:** Essa rota permite a exclusão de um aluno específico com base em seu ID.
    c. **Funcionalidade:** Remove permanentemente um aluno do sistema com base no ID fornecido na URL.

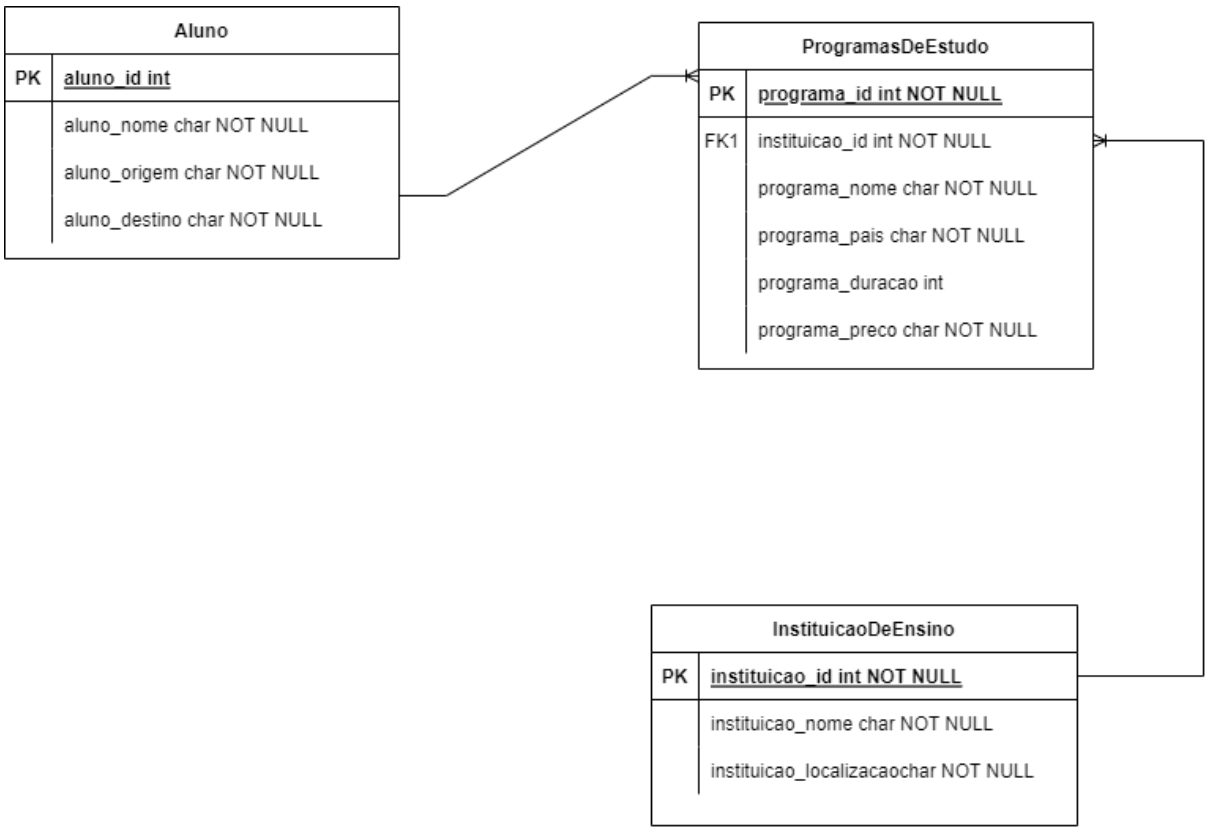
Essas rotas e endpoints formam a espinha dorsal da nossa API de gerenciamento de agência de intercâmbio, permitindo a interação completa com alunos e programas de estudo. Ao seguir os padrões RESTful, a API oferece uma estrutura consistente e eficaz para gerenciar os dados e atender às necessidades dos usuários.

Banco de Dados

Modelo Conceitual



Modelo Lógico



Formato de dados

Em um projeto de desenvolvimento de API, a escolha do formato de dados desempenha um papel fundamental na troca de informações entre os clientes (aplicativos, sistemas, usuários) e o servidor. Um dos formatos mais amplamente adotados e versáteis para esse fim é o JSON (JavaScript Object Notation).

Campos e Validações

Em qualquer API, a qualidade e a integridade dos dados são essenciais para o funcionamento correto e seguro da aplicação. Para atingir esse objetivo, é fundamental estabelecer campos de entrada bem definidos, realizar validações rigorosas e implementar tratamento adequado de erros. Este documento aborda detalhes sobre como os campos e as validações são tratados em nossa API de gerenciamento de agência de intercâmbio.

Campos de Entrada

Os campos de entrada em nossas solicitações API seguem uma estrutura clara e consistente. Cada endpoint da API especifica quais campos são necessários para uma operação bem-sucedida. Por exemplo, ao criar um novo aluno, os campos "nome", "origem" e "destino" são obrigatórios. As informações de cada campo são detalhadas na documentação da API para garantir que os clientes saibam quais dados devem fornecer.

Validações Necessárias

A fim de manter a integridade dos dados, implementamos validações rigorosas em nossos endpoints da API. Essas validações incluem:

1. **Validação de Formato:** Verificamos se os campos estão no formato correto, como nomes, que devem ser texto (letras ou espaços).

2. **Validação de Duração:** No caso de programas de estudo, garantimos que a duração seja um valor numérico positivo, representando meses ou anos.

3. **Validação de Preço:** Para programas de estudo, o preço deve ser um valor numérico positivo ou zero.

Tratamento de Erros

Erros podem ocorrer em qualquer sistema, e nossa API não é exceção. Para garantir que os clientes recebam feedback claro e útil, implementamos um sistema de tratamento de erros consistente. Os principais pontos incluem:

1. **Códigos de Erro Descritivos:** Cada erro é acompanhado de um código de erro descritivo, permitindo uma identificação rápida do problema.

2. **Mensagens de Erro Claras:** As mensagens de erro são formuladas de maneira clara e informativa, indicando qual validação ou condição específica não foi atendida.

3. **Códigos de Status HTTP Adequados:** Utilizamos códigos de status HTTP padrão para indicar o resultado de uma solicitação (por exemplo, 400 para solicitações inválidas, 404 para recursos não encontrados).

4. **Respostas JSON Estruturadas:** As respostas de erro seguem um formato JSON estruturado,

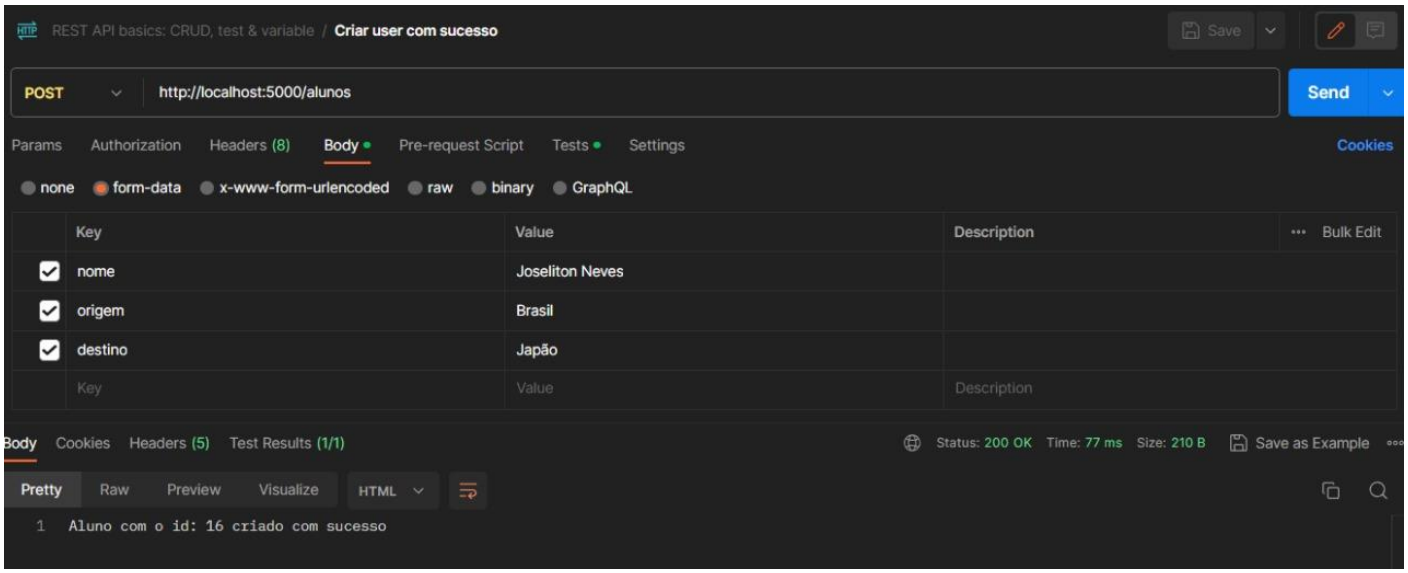
incluindo detalhes sobre o erro, código, mensagem e, quando apropriado, informações adicionais.

Em resumo, a API de gerenciamento de agência de intercâmbio se esforça para garantir que os campos de entrada sejam bem definidos, as validações sejam rigorosas e que os erros sejam tratados de maneira clara e informativa. Isso ajuda a manter a confiabilidade e a usabilidade da API, permitindo que os desenvolvedores interajam com nossos serviços de maneira eficaz e confiável.

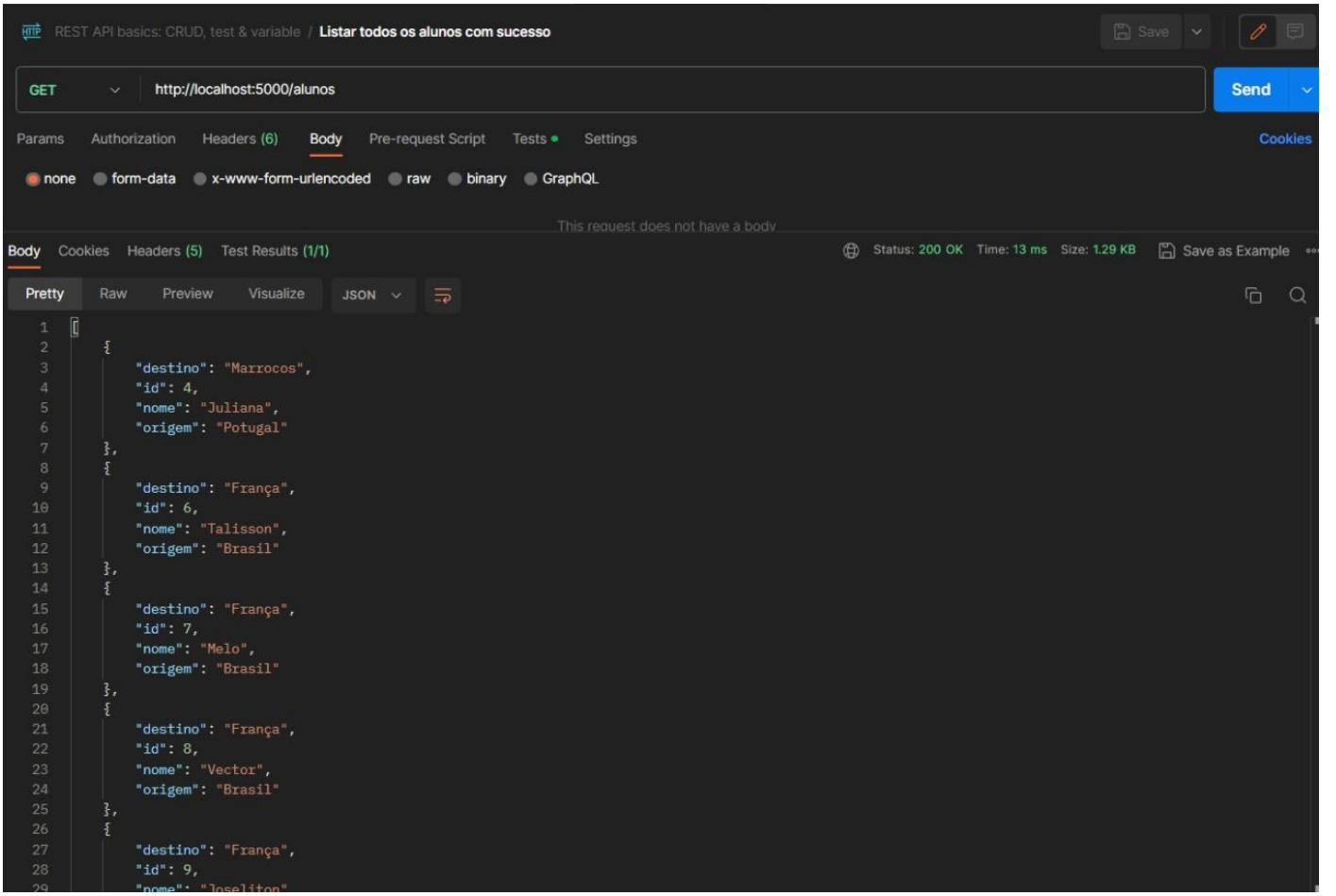
Exemplos de Uso

Nossa API de gerenciamento de agência de intercâmbio é projetada para ser intuitiva e fácil de usar. Para ajudar os desenvolvedores a se familiarizarem com o funcionamento da API, fornecemos uma série de exemplos de uso que demonstram como fazer solicitações para cada endpoint, juntamente com exemplos de respostas correspondentes.

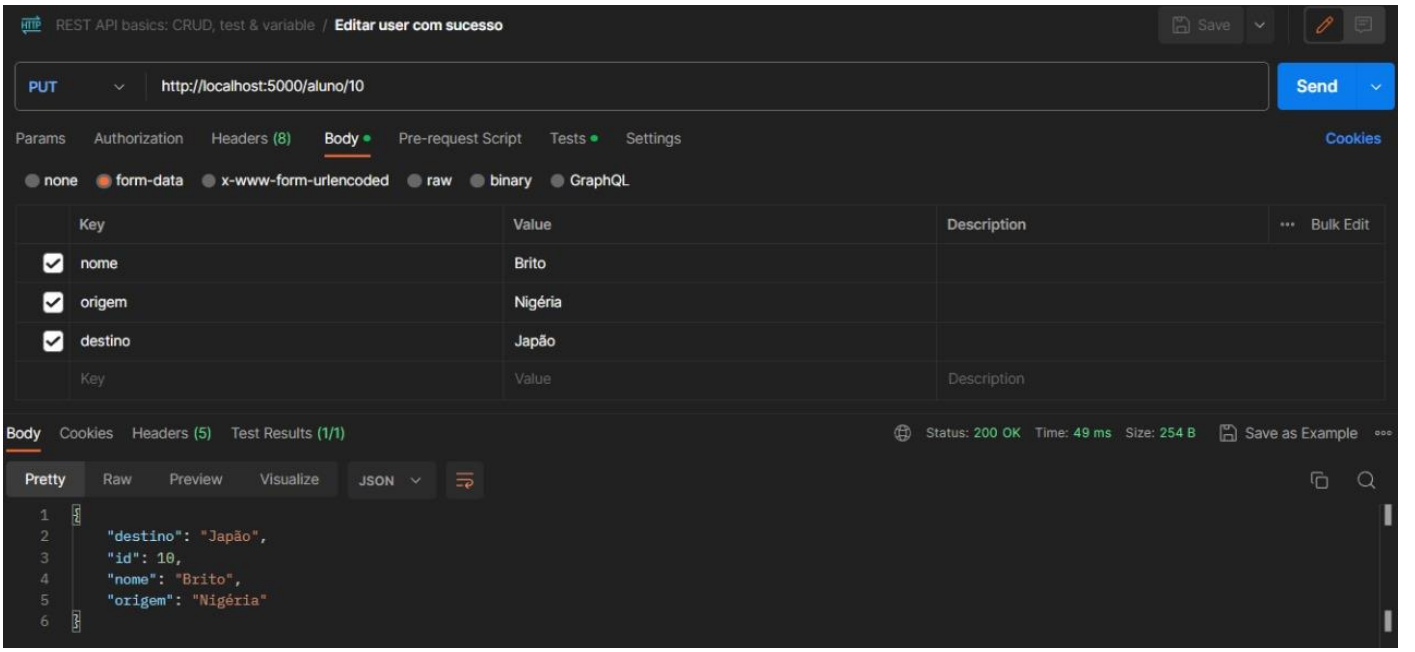
POST /alunos



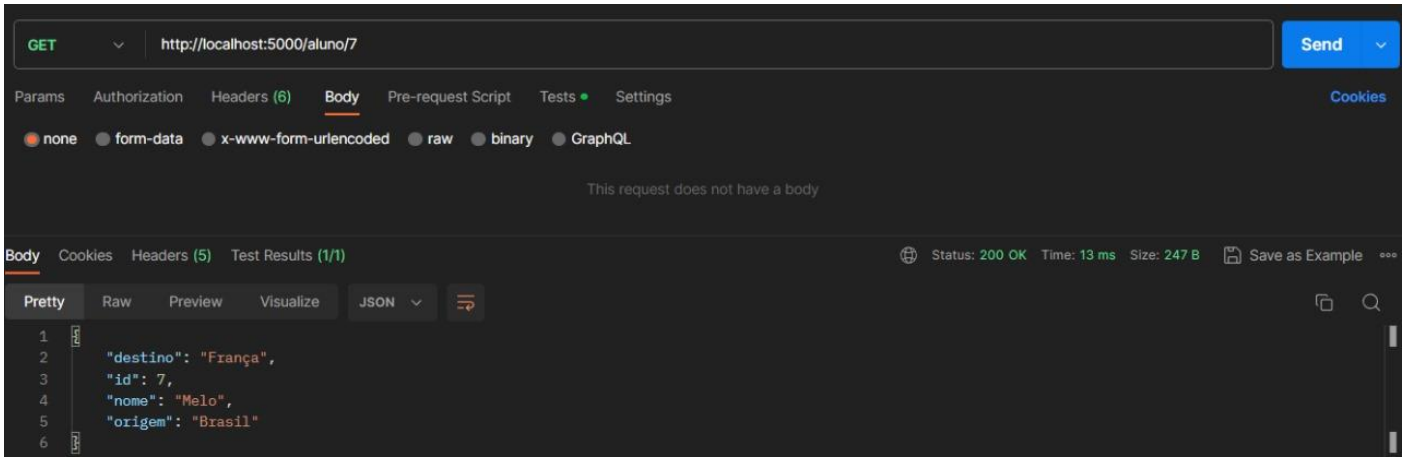
GET /alunos



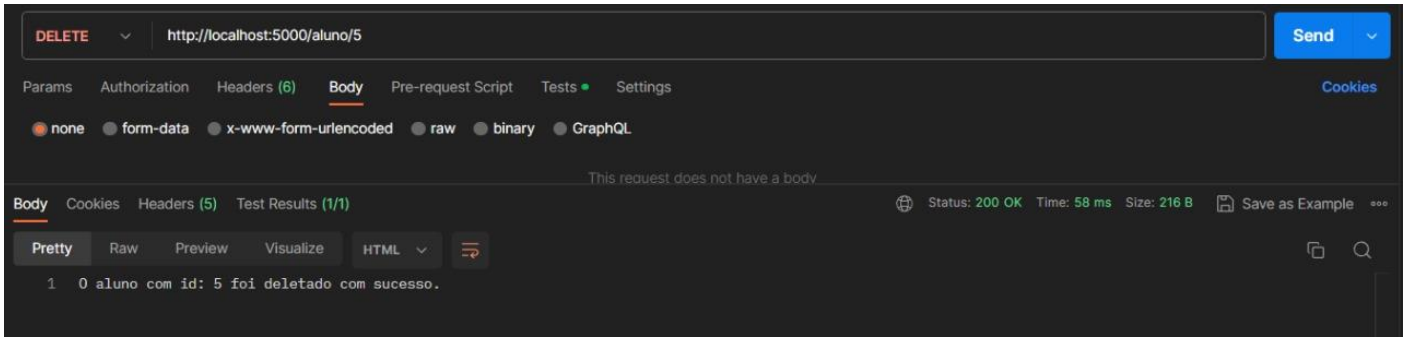
PUT /aluno/<id>



GET aluno/<id>



DELETE /aluno/<id>



Tratamento de Erros

O tratamento de erros desempenha um papel fundamental na experiência do usuário ao interagir com nossa API de gerenciamento de agência de intercâmbio. Para assegurar que os usuários recebam feedback claro e informativo, implementamos um sistema de tratamento de erros robusto e consistente. Este documento aborda detalhes sobre como os diferentes tipos de erros são tratados e comunicados aos usuários.

Códigos de Erro e Mensagens Descritivas

Nossa API utiliza códigos de erro e mensagens descritivas para identificar e comunicar problemas específicos. Cada erro é acompanhado de um código de erro único que permite uma identificação rápida do problema. As mensagens de erro são formuladas de maneira clara e informativa, indicando qual validação ou condição específica não foi atendida.

Códigos de Status HTTP

Utilizamos códigos de status HTTP padrão para indicar o resultado de uma solicitação. Isso ajuda os clientes a entenderem o estado da solicitação de forma imediata. Alguns dos códigos de status comuns incluem:

- **200 OK:** Indica que a solicitação foi bem-sucedida e a resposta contém os dados solicitados.
- **201 Created:** Usado quando uma nova entidade é criada com sucesso.

- **400 Bad Request:** Indica que a solicitação do cliente é inválida, muitas vezes devido a dados ausentes ou formatos incorretos.
- **401 Unauthorized:** Usado quando a autenticação é necessária, mas as credenciais fornecidas são inválidas ou ausentes.
- **403 Forbidden:** Indica que o cliente não tem permissão para acessar o recurso solicitado.
- **404 Not Found:** Usado quando o recurso solicitado não foi encontrado no servidor.

Respostas JSON Estruturadas

As respostas de erro seguem um formato JSON estruturado para garantir que os desenvolvedores possam analisá-las facilmente. Um exemplo de resposta de erro em JSON pode ser:

```
{
  "error_code": "400",
  "message": "Os dados fornecidos são inválidos. Verifique os campos e tente novamente."
}
```

Considerações de Segurança

A segurança é uma prioridade fundamental em nossa API de gerenciamento de agência de intercâmbio. Implementamos diversas medidas de segurança para proteger os dados e garantir a confidencialidade, integridade e disponibilidade das informações. Neste documento, detalhamos algumas das considerações de segurança mais importantes que orientam o funcionamento seguro de nossa API.

1. Proteção contra Ataques de Injeção

Um dos principais riscos de segurança em qualquer aplicação web é a possibilidade de ataques de injeção, como Injeção SQL ou Injeção de Scripts. Para mitigar esses riscos:

Prevenção de Injeção SQL: Utilizamos consultas parametrizadas e ORMs (Object-Relational Mapping) para interagir com o banco de dados, evitando assim a possibilidade de injeção SQL.

Validação de Dados de Entrada: Realizamos validações rigorosas em todos os dados de entrada, garantindo que apenas dados válidos e seguros sejam processados.

2. Autenticação e Autorização

A autenticação e autorização são elementos cruciais da segurança da API;

Autenticação: Exigimos autenticação adequada para acessar recursos protegidos. Os usuários devem fornecer credenciais válidas, como tokens de autenticação, para acessar dados sensíveis.

Autorização: Implementamos um sistema de controle de acesso baseado em funções (Role-Based Access Control - RBAC) para garantir que os usuários tenham permissões apropriadas para acessar recursos específicos.

3. Proteção de Dados Sensíveis

Comunicação Segura: Utilizamos protocolos de criptografia, como HTTPS, para garantir que os dados sejam transmitidos de forma segura entre os clientes e o servidor.

4. Atualizações e Correções de Segurança

Mantemos nossa API atualizada com as últimas correções de segurança e patches. Isso inclui atualizações de dependências e correções de vulnerabilidades conhecidas.

Em resumo, as considerações de segurança são uma parte integral de nossa API de gerenciamento de agência de intercâmbio. Estamos comprometidos em proteger os dados de nossos usuários e garantir um ambiente seguro para todas as interações. Continuamos a monitorar e aprimorar nossas práticas de segurança para atender às ameaças em constante evolução e manter a confiança de nossos usuários.

Desempenho e Escalabilidade

O desempenho e a escalabilidade são aspectos críticos para a eficácia e a confiabilidade de nossa API de gerenciamento de agência de intercâmbio. Neste documento, abordaremos considerações sobre como otimizamos o desempenho e planejamos a escalabilidade da API para atender às crescentes demandas de nossos usuários.

1. Otimização de Desempenho

A otimização de desempenho é uma prioridade em nossa API. Implementamos diversas estratégias para garantir que as solicitações sejam processadas de maneira rápida e eficiente:

Índices de Banco de Dados: Usamos índices eficazes em nosso banco de dados para acelerar consultas e reduzir o tempo de resposta.

Compactação de Dados: Minimizamos o tamanho das respostas JSON usando compactação, reduzindo a largura de banda necessária e melhorando o tempo de carregamento.

Otimização de Consultas: Revisamos regularmente as consultas SQL para garantir que sejam eficientes e que usem índices quando apropriado.

2. Escalabilidade Horizontal

Entendemos que o aumento na demanda pode ocorrer com o tempo. Para garantir que nossa API seja escalável, adotamos uma abordagem de escalabilidade horizontal:

Divisão de Serviços: Componentes da API são divididos em serviços independentes, permitindo que cada parte seja escalada individualmente conforme necessário.

3. Monitoramento e Análise de Desempenho

Monitoramos constantemente o desempenho da API para identificar gargalos e oportunidades de melhoria;

Análise de Logs: Analisamos logs de servidor para identificar problemas, erros e possíveis melhorias de desempenho.

4. Otimização de Banco de Dados

Nosso banco de dados é otimizado para garantir que as consultas sejam rápidas e eficazes:

Particionamento de Tabelas: Utilizamos particionamento de tabelas para dividir grandes conjuntos de dados, melhorando o desempenho das consultas.

Gerenciamento de Índices: Monitoramos e otimizamos índices de banco de dados para garantir consultas eficientes.

Em resumo, estamos comprometidos em fornecer uma API de alto desempenho e escalável para atender às necessidades crescentes de nossos usuários. Nossas práticas de otimização de desempenho e escalabilidade garantem que a API continue funcionando de maneira eficiente, mesmo quando enfrentamos picos de tráfego e aumento na demanda. Estamos constantemente refinando e aprimorando nossas estratégias para oferecer a melhor experiência possível aos nossos usuários.

Documentação da API

A documentação da API é uma parte essencial de nossa plataforma de gerenciamento de agência de intercâmbio. Ela serve como um guia completo para desenvolvedores, permitindo que eles compreendam, utilizem e integrem nossos serviços de maneira eficaz. Neste documento, descrevemos como nossa documentação da API será mantida e disponibilizada para a comunidade de desenvolvedores.

1. Atualizações Regulares

Para garantir que nossa documentação esteja sempre atualizada e precisa, seguimos um processo de revisão e atualização regular. Isso inclui:

- **Acompanhamento de Mudanças:** Monitoramos continuamente as mudanças em nossa API, incluindo atualizações de endpoints, campos de entrada e saída, e novos recursos.
- **Revisão de Conteúdo:** Realizamos revisões periódicas do conteúdo da documentação para garantir que ele reflita com precisão o estado atual da API.
- **Incorporação de Feedback:** Valorizamos o feedback dos desenvolvedores e incorporamos suas sugestões e observações à documentação.

2. Documentação Abrangente

Nossa documentação é abrangente e inclui informações detalhadas sobre:

- **Endpoints da API:** Descrevemos cada endpoint disponível, incluindo seus métodos HTTP suportados, parâmetros de entrada e exemplos de uso.
- **Formato de Dados:** Fornecemos informações sobre o formato de dados esperado nas solicitações e respostas da API, geralmente usando JSON.
- **Autenticação e Autorização:** Explicamos os métodos de autenticação suportados e como os desenvolvedores podem obter tokens de autenticação.
- **Campos e Validações:** Detalhamos os campos de entrada, validações necessárias e tratamento de erros.
- **Exemplos de Uso:** Fornecemos exemplos práticos de como fazer solicitações para cada endpoint e exemplos correspondentes de respostas.

- **Tratamento de Erros:** Explicamos como diferentes tipos de erros serão tratados e comunicados aos usuários.

- **Considerações de Segurança:** Detalhamos as medidas de segurança implementadas na API.

- **Desempenho e Escalabilidade:** Descrevemos como otimizamos o desempenho da API e como ela pode ser escalada conforme a demanda aumenta.

Cronograma do Projeto

O projeto de desenvolvimento da API de gerenciamento de agência de intercâmbio é uma jornada desafiadora que requer um planejamento sólido e um cronograma bem estruturado. Dado que o projeto começou em 8 de agosto e a primeira entrega está prevista para 27 de setembro, aqui está um cronograma que cobre as atividades e marcos importantes durante esse período:

Fase 1: Planejamento Inicial (8 de agosto a 14 de agosto)

Semana 1 (8 a 14 de agosto):

- Reunião de Kick-off do Projeto
- Definição de Objetivos e Escopo
- Escolha das Tecnologias e Ferramentas
- Design Inicial da API
- Alocação de Tarefas para as Equipes

Fase 2: Desenvolvimento e Implementação (15 de agosto a 19 de setembro)

Semana 2 (15 a 21 de agosto):

- Configuração do Ambiente de Desenvolvimento
- Desenvolvimento da Classe de Modelagem
- Início da Implementação da Classe de Repositórios

Semana 3 (22 a 28 de agosto):

- Continuação do Desenvolvimento da Classe de Repositórios
- Início da Implementação da Classe de Controles
- Preparação da Suite de Testes

Semana 4 (29 de agosto a 4 de setembro):

- Desenvolvimento e Testes da Classe de Controles
- Implementação de Roteiros de Teste no Postman ou SoapUI
- Revisão de Código e Correções

Semana 5 (5 a 11 de setembro):

- Finalização da Implementação da API
- Testes de Integração e Testes de Unidade
- Depuração e Resolução de Problemas
- Documentação Inicial da API

Semana 6 (12 a 18 de setembro):

- Preparação para a Primeira Entrega
- Finalização do Documento e API (20 de setembro)

Fase 3: Finalização e Ajustes (20 de setembro a 27 de setembro)

Semana 7 (20 a 27 de setembro):

- Últimos Ajustes na API
- Testes Finais e Validação
- Preparação para a Entrega (27 de setembro)