

Relatório da Aplicação

Alunos

Guilherme Henrique Luiz - 1A226881906.

Karen P. da Silva Moreira - 1A226881958

Descrição da Aplicação

Aplicação desenvolvida com intuito de promover a relação de três tabelas armazenadas em um banco de dados nos serviços da PlanetScale. Os dados gerenciados são providos de um cenário fictício de uma biblioteca, tendo-se por objetivo o gerenciamento dos dados das tabelas Alunos, Livros & Empréstimos.

Documentação da API do Servidor

Visão Geral

Esta documentação descreve a API do servidor, que gerencia operações relacionadas a uma biblioteca, empréstimos de livros e informações sobre alunos. O servidor foi desenvolvido usando Node.js, Express.js, e MySQL para a persistência de dados.

Base URL: `http://localhost:3000`

Visão Geral

Para rodar o projeto local você precisa seguir os seguintes passos:

- Abrir o caminho da pasta server no terminal e rodar o seguinte comando:

```
# Caminho:  
# ./server/  
  
# Comando:  
node server.js  
  
# Assim o servidor irá rodar e você pode acessar a aplicação
```

- Abrir o arquivo 'home.html' no navegador

Endpoints

Livros

1. Obter Todos os Livros

- URL: `/api/library/get/books`
- Método: `GET`
- Descrição: Retorna todos os livros presentes na biblioteca.

2. Obter Um Livro

- URL: `/api/library/get/book/:id`
- Método: `GET`
- Parâmetros da URL:
 - `id` : ID do livro a ser recuperado.
- Descrição: Retorna as informações de um livro específico com base no ID fornecido.

3. Adicionar Livro

- URL: `/api/library/add/book`
- Método: `POST`
- Corpo da Requisição (JSON):

```
{
  "id": 1,
  "titulo": "O Livro dos Exemplos",
  "autor": "Autor Exemplo"
}
```

- Descrição: Adiciona um novo livro à biblioteca.

4. Atualizar Livro

- URL: `/api/library/update/book/:id`

- **Método:** PATCH
- **Parâmetros da URL:**
 - `id`: ID do livro a ser atualizado.
- **Corpo da Requisição (JSON):**

```
{
  "titulo": "Novo Título",
  "autor": "Novo Autor"
}
```

- **Descrição:** Atualiza as informações de um livro específico com base no ID fornecido.

5. Excluir Livro

- **URL:** `/api/library/delete/book/:id`
- **Método:** DELETE
- **Parâmetros da URL:**
 - `id`: ID do livro a ser excluído.
- **Descrição:** Exclui um livro específico com base no ID fornecido.

Empréstimos

6. Obter Todos os Empréstimos

- **URL:** `/api/registers/get/registers`
- **Método:** GET
- **Descrição:** Retorna todos os registros de empréstimos.

7. Obter Um Empréstimo

- **URL:** `/api/registers/get/register/:id`
- **Método:** GET
- **Parâmetros da URL:**
 - `id`: ID do empréstimo a ser recuperado.

- **Descrição:** Retorna as informações de um empréstimo específico com base no ID fornecido.

8. Adicionar Empréstimo

- **URL:** `/api/registers/add/register`
- **Método:** `POST`
- **Corpo da Requisição (JSON):**

```
{
  "id_aluno": 1,
  "id_livro": 1,
  "data_emprestimo": "2023-01-01",
  "data_devolucao": "2023-01-15"
}
```

- **Descrição:** Registra um novo empréstimo na biblioteca.

9. Atualizar Empréstimo

- **URL:** `/api/registers/update/register/:id`
- **Método:** `PATCH`
- **Parâmetros da URL:**
 - `id`: ID do empréstimo a ser atualizado.
- **Corpo da Requisição (JSON):**

```
{
  "data_devolucao": "2023-02-01"
}
```

- **Descrição:** Atualiza as informações de um empréstimo específico com base no ID fornecido.

10. Excluir Empréstimo

- **URL:** `/api/registers/delete/register/:id`
- **Método:** `DELETE`
- **Parâmetros da URL:**

- `id`: ID do empréstimo a ser excluído.
- **Descrição:** Exclui um empréstimo específico com base no ID fornecido.

Alunos

11. Obter Todos os Alunos

- **URL:** `/api/students/get/students`
- **Método:** `GET`
- **Descrição:** Retorna todos os alunos cadastrados.

12. Obter Um Aluno

- **URL:** `/api/students/get/student/:id`
- **Método:** `GET`
- **Parâmetros da URL:**
 - `id`: ID do aluno a ser recuperado.
- **Descrição:** Retorna as informações de um aluno específico com base no ID fornecido.

13. Adicionar Aluno

- **URL:** `/api/students/add/student`
- **Método:** `POST`
- **Corpo da Requisição (JSON):**

```
{
  "id": 1,
  "nome": "Nome do Aluno",
  "matricula": "12345"
}
```

- **Descrição:** Adiciona um novo aluno ao sistema.

14. Atualizar Aluno

- **URL:** `/api/students/update/student/:id`
- **Método:** `PATCH`

- **Parâmetros da URL:**
 - `id`: ID do aluno a ser atualizado.
- **Corpo da Requisição (JSON):**

```
{
  "matricula": "54321"
}
```

- **Descrição:** Atualiza as informações de um aluno específico com base no ID fornecido.

15. Excluir Aluno

- **URL:** `/api/students/delete/student/:id`
- **Método:** `DELETE`
- **Parâmetros da URL:**
 - `id`: ID do aluno a ser excluído.
- **Descrição:** Exclui um aluno específico com base no ID fornecido.

Manipulação de Erros

Em caso de qualquer erro durante o processamento de uma solicitação, a API retornará uma resposta com status `500 Internal Server Error`, e a mensagem de erro será incluída no corpo da resposta. As mensagens de erro fornecem detalhes para facilitar a depuração.

Exemplos de Uso

Exemplo de Solicitação para Obter Todos os Livros

```
GET <http://localhost:3000/api/library/get/books>
```

Exemplo de Solicitação para Adicionar um Livro

```
POST <http://localhost:3000/api/library/add/book>
Content-Type: application/json
```

```
{
  "id": 2,
  "titulo": "O Novo Livro",
  "autor": "Autor Desconhecido"
}
```

Exemplo de Resposta

para Obter Todos os Livros

```
[
  {
    "id": 1,
    "titulo": "O Livro dos Exemplos",
    "autor": "Autor Exemplo"
  },
  {
    "id": 2,
    "titulo": "O Novo Livro",
    "autor": "Autor Desconhecido"
  }
]
```

Considerações Finais

Instale as dependências necessárias usando `npm install express cors mysql2 dotenv`.

Observações

A Aplicação tinha o objetivo de ser disponibilizada através do Vercel, porém os métodos POST, PATCH & DELETE não estão funcionando fora do 'localhost', sendo assim a documentação está considerando o 'localhost' como base e não o Vercel.

Nos pasta raiz terão menções (arquivos e pastas) que fazem o trabalho funcionar no Vercel, sendo elas:

```
.vercel
pasta 'api'
vercel.json
```

Caso tenha interesse de visualização da versão web, ela está disponível no seguinte [link](#):

Node JS Project with PlanetScale

<https://library-dwb.vercel.app/>