Para atualizar e atualizar os dados dos alunos a partir de uma planilha Excel para o banco de dados PostgreSQL, podemos usar uma biblioteca xlsxpara ler o arquivo e typeormpara inserir ou atualizar os dados.

**🔧 Passos para Implementação:**

1. **Instalar dependências** :

npm install xlsx multer

1. **Criar rota para upload e importação** : Adicione a seguinte rota em src/routes/alunoRoutes.ts:

import { Router } from "express";

import multer from "multer";

import { AlunoController } from "../controllers/AlunoController";

const router = Router();

const upload = multer({ dest: "uploads/" }); // Define pasta temporária

const alunoController = new AlunoController();

router.post("/importar", upload.single("file"), alunoController.importarAlunos);

export { router as alunoRoutes };

1. **Crie o driver para processar a planilha** : Adicione src/controllers/AlunoController.ts:

import { Request, Response } from "express";

import \* as xlsx from "xlsx";

import fs from "fs";

import { AlunoService } from "../services/AlunoService";

export class AlunoController {

async importarAlunos(req: Request, res: Response) {

if (!req.file) {

return res.status(400).json({ error: "Nenhum arquivo enviado" });

}

const filePath = req.file.path;

const workbook = xlsx.readFile(filePath);

const sheetName = workbook.SheetNames[0];

const sheet = workbook.Sheets[sheetName];

const alunos = xlsx.utils.sheet\_to\_json(sheet);

const alunoService = new AlunoService();

await alunoService.importarAlunos(alunos);

fs.unlinkSync(filePath); // Remove o arquivo temporário após a leitura

return res.json({ success: "Dados importados com sucesso" });

}

}

1. **Criar o serviço para salvar os alunos no banco** : Adicione src/services/AlunoService.ts:

import { AppDataSource } from "../ormconfig";

import { Aluno } from "../entities/Aluno";

export class AlunoService {

async importarAlunos(alunos: any[]) {

const alunoRepo = AppDataSource.getRepository(Aluno);

for (const dado of alunos) {

let aluno = await alunoRepo.findOne({ where: { matricula: dado.MATRICULA\_ALUNO } });

if (aluno) {

// Atualizar dados existentes

aluno.nome = dado.NOME\_ALUNO;

aluno.telefone = dado.TELEFONE\_ALUNO;

aluno.email = dado.EMAIL\_ALUNO;

aluno.nome\_responsavel\_01 = dado.NOME\_RESPONSÁVEL\_01;

aluno.telefone\_responsavel\_01 = dado.TELEFONE\_RESPONSÁVEL\_01;

aluno.email\_responsavel\_01 = dado.EMAIL\_RESPONSÁVEL\_01;

aluno.nome\_responsavel\_02 = dado.NOME\_RESPONSÁVEL\_02;

aluno.telefone\_responsavel\_02 = dado.TELEFONE\_RESPONSÁVEL\_02;

aluno.email\_responsavel\_02 = dado.EMAIL\_RESPONSÁVEL\_02;

} else {

// Criar novo aluno

aluno = alunoRepo.create(dado);

}

await alunoRepo.save(aluno);

}

}

}

Agora, para importar os dados, basta fazer uma requisição POST para:

http://localhost:3000/alunos/importar

enviando um arquivo Excel( .xlsx) no campo file.

O campo file é o nome do campo no formulário usado para enviar o arquivo Excel via multipart/form-data. Se você estiver testando com **Postman** , **Insomnia** ou um front-end, você deve garantir que o upload seja feito corretamente.

**📌 Como enviar o arquivo via Postman:**

1. Abra o **Carteiro** .
2. Selecione o método **POST** .
3. No campo URL, insira:

http://localhost:3000/alunos/importar

1. Vá até a aba **Body** e selecione **form-data** .
2. No campo **Key** , insira file(exatamente esse nome).
3. No campo **Type** , selecione **File** .
4. Clique no botão para adicionar um arquivo e selecione sua planilha Excel ( .xlsx).
5. Pressione **Enviar** .

**📌 Enviando via Front-End com JavaScript:**

Se estiver fazendo isso em um site com **HTML + JavaScript** , você pode criar um formulário assim:

<form id="uploadForm">

<input type="file" id="fileInput" name="file" accept=".xlsx" />

<button type="submit">Enviar</button>

</form>

<script>

document.getElementById("uploadForm").addEventListener("submit", async (event) => {

event.preventDefault();

const fileInput = document.getElementById("fileInput").files[0];

if (!fileInput) {

alert("Selecione um arquivo!");

return;

}

const formData = new FormData();

formData.append("file", fileInput);

const response = await fetch("http://localhost:3000/alunos/importar", {

method: "POST",

body: formData

});

const result = await response.json();

console.log(result);

alert(result.success || result.error);

});

</script>

Se você estiver enviando via **Insomnia** , o processo é o mesmo do Postman:

* Usar **POST**
* Vá para **Corpo → Formulário Multipartes**
* Adicione file como chave e selecione o arquivo

Isso garante que o back-end receba corretamente o arquivo enviado no campo file. 🚀 Avise se precisar de mais ajuda!