# Deletando Transações

Nesta aula, vamos aprender a criar rotas e controladores para deletar transações na nossa API.

**Objetivos da Aula:**

- Compreender como criar rotas para deletar transações
- Configurar controladores para manipular as requisições de deleção de transações

1. **Configurando a Rota para Deletar Transações**

**Atualizar o Arquivo transactions.js**

Abra o arquivo **transactions.js** dentro da pasta **routes**:

Vamos adicionar uma nova rota **DELETE** para deletar transações.

**Adicione o seguinte código ao arquivo transactions.js**:

**Código JavaScript**

```
const express = require('express'); // Importa o framework Express
const router = express.Router(); // Cria um novo roteador
const transactionsController = require('../controllers/transactionsController'); // Importa o controlador de transações

// Definindo uma rota para obter todas as transações
router.get('/', transactionsController.getAllTransactions);

// Definindo uma rota para adicionar uma nova transação
router.post('/', transactionsController.addTransaction);

// Definindo uma rota para atualizar uma transação existente (substituição completa)
router.put('/:id', transactionsController.updateTransactionPut);

// Definindo uma rota para atualizar uma transação existente (atualização parcial)
router.patch('/:id', transactionsController.updateTransactionPatch);



// Definindo uma rota para deletar uma transação existente
router.delete('/:id', transactionsController.deleteTransaction);



// Exportando o roteador
module.exports = router;
```

**Explicação do Código**:

- **router.delete('/:id', transactionsController.deleteTransaction);:** Define uma rota DELETE para deletar uma transação específica.

2. **Configurar o Controlador para Deletar Transações**

Abra o arquivo **transactionsController.js** dentro da pasta **controllers**:

Vamos adicionar uma nova função para deletar transações.

**Adicione o seguinte código ao arquivo transactionsController.js**:

**Código JavaScript**

```
const db = require('../config/db'); // Importa a conexão com o banco de dados


// Função para obter todas as transações
const getAllTransactions = (req, res) => {
  db.query('SELECT * FROM transactions', (err, results) => {
    if (err) {
      console.error('Erro ao obter transações:', err);
      res.status(500).send('Erro ao obter transações');
      return;
    }
    res.json(results);
  });
};


// Função para adicionar uma nova transação
const addTransaction = (req, res) => {
  const { date, amount, description, category, account, user_id } = req.body;
  db.query(
    'INSERT INTO transactions (date, amount, description, category, account, user_id) VALUES
(?, ?, ?, ?, ?, ?)',
    [date, amount, description, category, account, user_id],
    (err, results) => {
      if (err) {
        console.error('Erro ao adicionar transação:', err);
        res.status(500).send('Erro ao adicionar transação');
        return;
      }
      res.status(201).send('Transação adicionada com sucesso');
    }
  );
};
```

```
// Função para atualizar uma transação existente (substituição completa)
const updateTransactionPut = (req, res) => {
  const { id } = req.params;
  const { date, amount, description, category, account, user_id } = req.body;
  db.query(
    'UPDATE transactions SET date = ?, amount = ?, description = ?, category = ?, account = ?,
user_id = ? WHERE id = ?',
    [date, amount, description, category, account, user_id, id],
    (err, results) => {
      if (err) {
        console.error('Erro ao atualizar transação:', err);
        res.status(500).send('Erro ao atualizar transação');
        return;
      }
      res.send('Transação atualizada com sucesso');
    }
  );
};




// Função para atualizar uma transação existente (atualização parcial)
const updateTransactionPatch = (req, res) => {
  const { id } = req.params;
  const fields = req.body;
  const query = [];
  const values = [];

  for (const [key, value] of Object.entries(fields)) {
    query.push(`${key} = ?`);
    values.push(value);
  }

  values.push(id);

  db.query(
    `UPDATE transactions SET ${query.join(', ')} WHERE id = ?`,
    values,
    (err, results) => {
      if (err) {
        console.error('Erro ao atualizar transação:', err);
        res.status(500).send('Erro ao atualizar transação');
        return;
      }
      res.send('Transação atualizada com sucesso');
    }
  );
};
```

```
// Função para deletar uma transação existente
const deleteTransaction = (req, res) => {
  const { id } = req.params;
  db.query('DELETE FROM transactions WHERE id = ?', [id], (err, results) => {
    if (err) {
      console.error('Erro ao deletar transação:', err);
      res.status(500).send('Erro ao deletar transação');
      return;
    }
    res.send('Transação deletada com sucesso');
  });
};



module.exports = {
  getAllTransactions,
  addTransaction,
  updateTransactionPut,
  updateTransactionPatch,
  deleteTransaction
};
```

**Explicação do Código**:

- **const deleteTransaction = (req, res) => {...};:** Define uma função arrow chamada deleteTransaction que recebe dois parâmetros, req (requisição) e res (resposta). Esta função será usada para deletar uma transação existente no banco de dados.
- **const { id } = req.params;:** Obtém o id da transação a partir dos parâmetros da URL.
- **db.query('DELETE FROM transactions WHERE id = ?', [id], (err, results) => {...});:** Executa uma query SQL para deletar a transação no banco de dados. A função callback recebe dois parâmetros, err (erro) e results (resultados).

**Parte Prática (1 hora):**

**Ação:**

1. Atualizar o arquivo **transactions.js** para definir a rota de deleção de transações.
2. Atualizar o arquivo **transactionsController.js** para definir o controlador de deleção de transações.
3. Testar a rota **DELETE** para **/api/transactions/:id** para garantir que transações podem ser deletadas.

**Teste Prático com Insomnia:**

Vamos usar o Insomnia para testar a rota DELETE para /api/transactions/:id.

**Testando a Rota DELETE para Deletar Transações**

1. **Abra o Insomnia**.
2. **Crie uma Nova Requisição**:
    - No menu lateral esquerdo, clique no botão "+" ao lado de "Debug" para adicionar uma nova requisição.
    - Dê um nome à sua requisição, por exemplo, "Delete Transaction".
    - Selecione o método HTTP como "DELETE".
    - Clique em "Create".

3. **Configure a URL da Requisição**:
    - Na barra de URL, insira http://localhost:3000/api/transactions/1 (onde 1 é o id da transação que deseja deletar).

4. **Envie a Requisição**:
    - Clique no botão "Send" para enviar a requisição.

- **Verifique a Resposta**:
    - Verifique o painel de resposta no Insomnia.
    - A resposta deve indicar que a transação foi deletada com sucesso.

**Exemplo de Resposta Esperada para Deleção**

**Código JSON**

```
{
  "message": "Transação deletada com sucesso"
}
```

**Resolução de Problemas Comuns**

**Erro 404 (Not Found)**

- Verifique se o servidor está rodando corretamente.
- Certifique-se de que a URL está correta e que a rota /api/transactions/:id está configurada no server.js.

**Erro 500 (Internal Server Error)**

- Verifique os logs do servidor para identificar a causa do erro.
- Certifique-se de que a conexão com o banco de dados está configurada corretamente.