

Aula 07

1. Criar o modelo Jogador
 2. Criar migration “jogadores”
 3. Rodar migration “jogadores”
 4. Criar a nova rota GET jogadores
 5. Criar a nova rota POST jogador
 6. Criar as novas rotas PUT jogador e DELETE jogador
-

1. Criar o modelo Jogador

Na pasta `./src/models` crie o arquivo `jogador.ts` com o seguinte código:

```
import { DataTypes, Model } from 'sequelize';
import { sequelize } from '../database';

export class Jogador extends Model {
  public id!: number;
  public nome!: string;
}

Jogador.init({
  {
    id: {
      type: DataTypes.INTEGER,
      autoIncrement: true,
      primaryKey: true,
    },
    nome: {
      type: DataTypes.STRING,
      allowNull: false,
    },
  },
  {
    sequelize,
    modelName: 'Jogador',
    tableName: 'jogadores',
    timestamps: false,
  }
});
```

2. Criar migration “jogadores”

Para criar a migration, rode o seguinte comando no terminal

```
npx sequelize-cli model:generate --name Jogador --attributes nome:string
```

Abra o arquivo de migração que foi gerado automaticamente pelo comando `model:generate`, que estará dentro da pasta `migrations/`. O nome do arquivo será algo como `XXXXXXXXXXXXX-create-jogador.js`, onde `XXXXXXXXXXXXX` é um timestamp.

Verifique e ajuste a migração para garantir que todas as colunas e tipos estejam corretos. O arquivo deve se parecer com isso:

```
'use strict';

module.exports = {
  up: async (queryInterface, Sequelize) => {
    await queryInterface.createTable('jogadores', {
      id: {
        allowNull: false,
        autoIncrement: true,
        primaryKey: true,
        type: Sequelize.INTEGER,
      },
      nome: {
        type: Sequelize.STRING,
        allowNull: false,
      },
      // Adicione mais colunas se necessário
    });
  },
  down: async (queryInterface, Sequelize) => {
    await queryInterface.dropTable('jogadores');
  }
};
```

3. Rodar a migration “jogadores”

Agora rode a migration com o comando `npx sequelize-cli db:migrate`. Caso tenha dado algo errado e precise reverter a execução da migration, use o comando `'npx sequelize-cli db:migrate:undo`

4. Criar a nova rota GET jogadores

Atualize o código do arquivo `index.ts` com o seguinte trecho de código:

```
// Rota GET para buscar todos os jogadores
app.get('/jogadores', async (req: Request, res: Response) => {
  try {
    const jogadores = await Jogador.findAll();
    res.status(200).json(jogadores);
  } catch (error) {
```

```

    res.status(500).json({ error: 'Erro ao buscar jogadores' });
  }
});

```

5. Criar a nova rota POST jogador

Para criar a rota de POST jogador, basta atualizarmos o `index.ts` com o seguinte trecho de código:

```

// Rota POST para criar um novo jogador
app.post('/jogador', async (req: Request, res: Response) => {
  const { nome } = req.body;

  try {
    const novoJogador = await Jogador.create({ nome });
    res.status(201).json(novoJogador);
  } catch (error) {
    res.status(500).json({ error: 'Erro ao criar jogador' });
  }
});

```

6. Criar as novas rotas PUT jogador e DELETE jogador

De forma semelhante, para criarmos as rotas PUT e DELETE jogador utilize o seguinte trecho de código:

```

// Rota PUT para atualizar um jogador existente
app.put('/jogador/:id', async (req: Request, res: Response) => {
  const { id } = req.params;
  const { nome } = req.body;

  try {
    const jogador = await Jogador.findById(id);

    if (!jogador) {
      res.status(404).json({ error: 'Jogador não encontrado' });
      return;
    }

    jogador.nome = nome;
    await jogador.save();

    res.status(200).json(jogador);
  } catch (error) {
    res.status(500).json({ error: 'Erro ao atualizar jogador' });
  }
});

```

```

// Rota DELETE para remover um jogador existente
app.delete('/jogador/:id', async (req: Request, res: Response) => {
  const { id } = req.params;

  try {
    const jogador = await Jogador.findByPk(id);

    if (!jogador) {
      res.status(404).json({ error: 'Jogador não encontrado' });
      return;
    }

    await jogador.destroy();
    res.status(204).send(); // Resposta sem conteúdo após deletar com sucesso
  } catch (error) {
    res.status(500).json({ error: 'Erro ao deletar jogador' });
  }
});

```