

Aula 06

1. Criar migration “partidas”
2. Rodar a migration “partidas”
3. Criar nova rota GET /partidas
4. Criar nova rota POST partida

1. Criar migration “partidas”

Para criar a migration, rode o seguinte comando no terminal

```
npx sequelize-cli model:generate --name Partida --attributes jogador_brancas:integer,jogador
```

Feito isso, abra a pasta /migrations na raiz do projeto e revise o arquivo gerado pelo sequelize-cli. Você deve alterar o arquivo que no final vai ficar com o seguinte código:

```
'use strict';

module.exports = {
  up: async (queryInterface, Sequelize) => {
    await queryInterface.createTable('partidas', {
      id: {
        allowNull: false,
        autoIncrement: true,
        primaryKey: true,
        type: Sequelize.INTEGER
      },
      jogador_brancas: {
        type: Sequelize.INTEGER,
        allowNull: false,
      },
      jogador_pretas: {
        type: Sequelize.INTEGER,
        allowNull: false,
      },
      data: {
        type: Sequelize.DATE,
        allowNull: false,
        defaultValue: Sequelize.NOW,
      },
      resultado: {
        type: Sequelize.STRING,
        allowNull: false,
        validate: {
          isIn: [['brancas', 'pretas', 'empate']],
        },
      },
    });
  },
  down: async (queryInterface, Sequelize) => {
    await queryInterface.dropTable('partidas');
  }
};
```

```

    }
  });
},
down: async (queryInterface, Sequelize) => {
  await queryInterface.dropTable('partidas');
}
};

```

Ainda, para não ficar forçando o **truncate** e **criação da tabela** todas as vezes que a API for executada, altere o código no arquivo `src/index.ts` para:

```

sequelize.sync().then(() => {
  console.log('Tabelas sincronizadas. ');
});

```

A opção `force: true` recria as tabelas sempre que você roda a aplicação, o que apaga os dados existentes.

2. Rodar a migration “partidas”

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`

3. Criar nova rota GET /partidas

Feita a migration podemos implementar a rota GET /partidas no `index.ts`:

```

// Rota para listar todas as partidas
app.get('/partidas', async (req: Request, res: Response) => {
  const partidas = await Partida.findAll();
  res.json(partidas);
});

```

4. Criar nova rota POST partida

Já para conseguir postar uma partida a partir do postman, implemente o seguinte código no arquivo `index.ts`

```

// Rota para criar uma nova partida
app.post('/partidas', async (req: Request, res: Response) => {
  const { jogador_brancas, jogador_pretas, data, resultado } = req.body;

  if (!jogador_brancas || !jogador_pretas || !resultado) {
    res.status(400).json({ error: 'jogador_brancas, jogador_pretas e Resultado são obrigatórios' });
    return;
  }
}

```

```
if !data {
  const novaPartida = await Partida.create({
    jogador_branças,
    jogador_pretas,
    resultado,
  });
}
else {
  const novaPartida = await Partida.create({
    jogador_branças,
    jogador_pretas,
    data,
    resultado,
  });
}

res.status(201).json(novaPartida);
});
```