Durante a aula usamos o formato **embedding** para unirmos os dados de autor ao livro correspondente.

Confira abaixo um exemplo simples do mesmo processo, mas usando o formato **referencing**.

### No arquivo Livro.js:

Ao invés do autorSchema, adicionamos a propriedade autor tendo como valor um **ID** do tipo ObjectId do MongoDB, fazendo referência à coleção autores.

**const** livroSchema = **new** mongoose.**Schema**({

id: { type: mongoose.Schema.Types.ObjectId },

titulo: { type: **String**, required: **true** },

editora: { type: **String** },

preco: { type: **Number** },

paginas: { type: **Number** },

autor: {type: mongoose.Schema.Types.ObjectId, ref: 'autores', required: **true**},

}, { versionKey: **false** });

### No arquivo livroController.js:

Quando usamos references o autor não faz mais parte do objeto livro. Assim, cada livro deve ser “populado” com as referências do autor.

Os métodos livro.find({}).populate("autor").exec(); vão utilizar o ID informado no campo autor do livro para buscar a referência desse ID e “popular” a propriedade.

**static** **async** listarLivros (req, res) {

**try** {

**const** listaLivros = **await** livro.**find**({}).**populate**("autor").**exec**();

res.**status**(200).**json**(listaLivros);

} **catch** (erro) {

res.**status**(500).**json**({ message: `${erro.message} - falha na requisição` });

}

};

Já o método de cadastrar livro fica como estava anteriormente, pois agora o schema Livro apenas recebeu a propriedade autor com o tipo de dado ID.

**static** **async** cadastrarLivro (req, res) {

**try** {

**const** novoLivro = **await** livro.**create**(req.body);

res.**status**(201).**json**({ message: "criado com sucesso", livro: novoLivro });

} **catch** (erro) {

res.**status**(500).**json**({ message: `${erro.message} - falha ao cadastrar livro` });

}

}

### Testes no Postman

Faça um novo teste no Postman com POST em http://localhost:3000/autores adicionando um novo autor:

{

"nome": "Machado de Assis",

"nacionalidade": "Brasil"

}

O objeto adicionado na coleção autores deverá ter a seguinte estrutura:

{

"nome": "Machado de Assis",

"nacionalidade": "Brasil",

"\_id": "64c4303f71627bda06635b6f"

}

Copie o ID do autor recém-criado e utilize este dado na criação de um novo livro no Postman com POST em http://localhost:3000/livros:

{

"titulo": "Dom Casmurro",

"autor": "64c4303f71627bda06635b6f"

}

O retorno deverá ser um novo livro com a seguinte estrutura:

{

"message": "criado com sucesso",

"livro": {

"titulo": "Dom Casmurro",

"autor": "64c4303f71627bda06635b6f",

"\_id": "64c4306d71627bda06635b71"

}

}

Agora, uma requisição do tipo GET em http://localhost:3000/livros deverá retornar o livro já com o autor “referenciado”:

{

"\_id": "64c4306d71627bda06635b71",

"titulo": "Dom Casmurro",

"autor": {

"\_id": "64c4303f71627bda06635b6f",

"nome": "Machado de Assis",

"nacionalidade": "Brasil"

}

}