



Para saber mais: belongsTo para quê?

Se você pesquisou tutoriais sobre como fazer associações com Sequelize, pode ter visto que em alguns deles a associação 1:n (um para muitos) é feita utilizando somente o método `hasMany()`, sem acrescentar o `belongsTo()`. Vai funcionar? Então por que a documentação do Sequelize diz para utilizar os dois métodos juntos?

Vamos pegar o seguinte exemplo:

```
Equipe.hasMany(Atleta);
Atleta.belongsTo(Equipe);
```

Ou seja, uma equipe tem vários (`hasMany`) atletas, mas atletas pertencem à (`belongsTo`) somente uma equipe cada.

Bom, mas e aí?

Quando utilizamos o método `Atleta.belongsTo(Equipe)` o Sequelize cria, “por baixo dos panos” alguns métodos “getters” e “setters”, como por exemplo `atleta.getEquipe()`.

O método `Equipe.hasMany(Atleta)` faz a associação na outra ponta, permitindo a criação do método `equipe.getAtletas()`. A criação destes métodos é um comportamento padrão do Sequelize, mesmo que não tenham sido usados no projeto que fizemos no curso.

Se utilizarmos somente um dos métodos - por exemplo, somente o `hasMany` em um dos lados da relação - seria possível usar o método para `get` (trazer) todas as atletas de uma equipe, mas não a equipe a que pertence uma atleta.