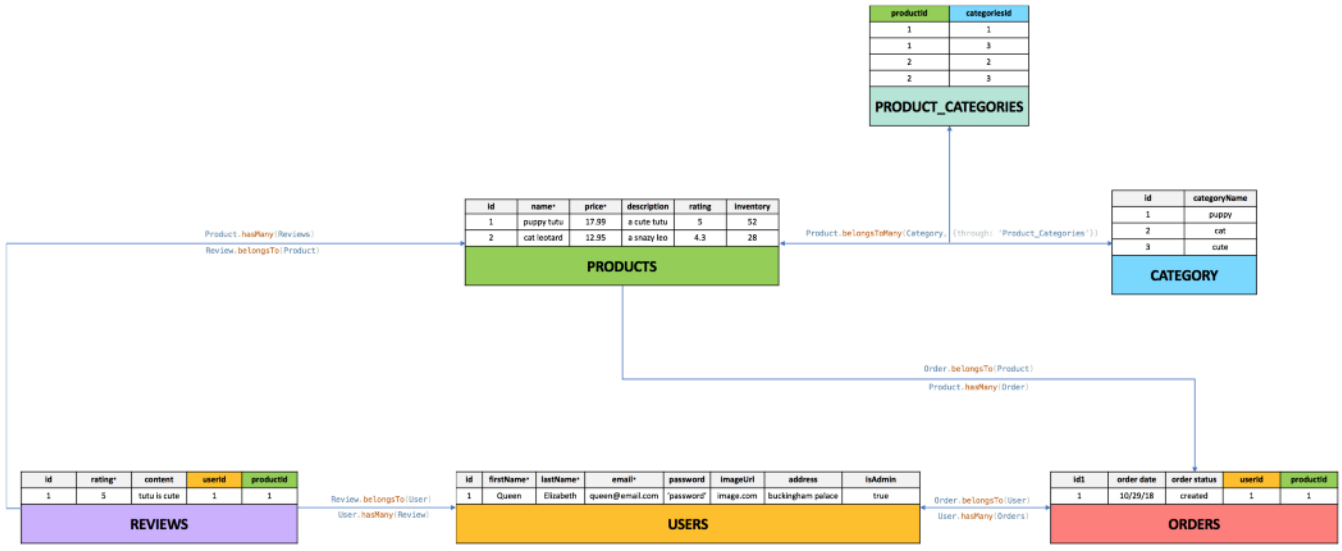
Julianne Crawford
53 Followers · About Follow

Sequelize Associations & Magic Methods

Julianne Crawford Dec 1, 2018 · 3 min read



Objective

The objective of this article is to provide more understandable (visual) documentation on Sequelize associations and magic methods. If you're new to Sequelize and/or relational databases, take a look at these articles (here and here) for some background information.

Motivation

pages of deterring and dense documentation. When I was first learning Sequelize, I struggled to understand model associations and their respective “magic methods”.

Associations & Magic Methods

An association in Sequelize is a relationship between two models (a source and target model). Creating an association generally involves adding a foreign key to one model which creates a reference to the other. There are several Sequelize associations, including *has one*, *belongsTo*, *hasMany*, and *belongsToMany*. In terms of implementation, the key difference between these associations is in which model (the source or the target) the foreign key resides. I’ve made a series of visualization tools (see below) to help differentiate between these associations.

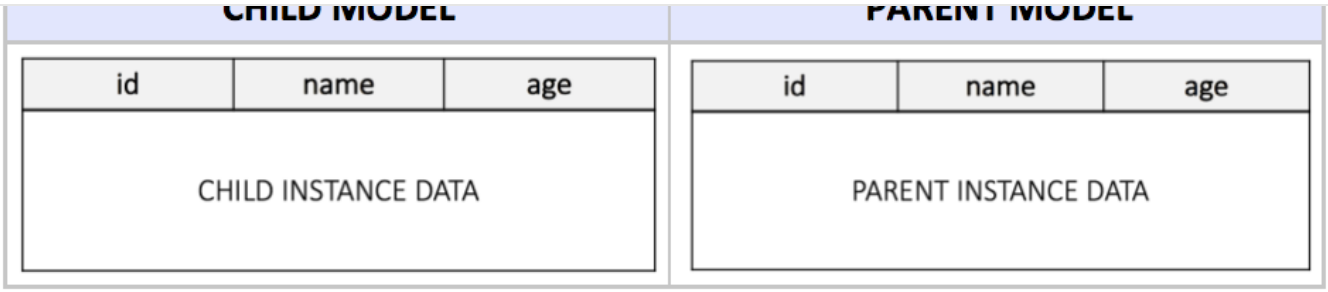
“Magic methods” are simply methods that are generated by defining Sequelize associations. As their name implies, the capabilities they invoke are quite magical. The visualizations below also outline which magic methods a given association will create. Alternatively, another way to easily see what magic methods a model has access to, is to use:

```
Object.keys(ModelName.prototype)
```

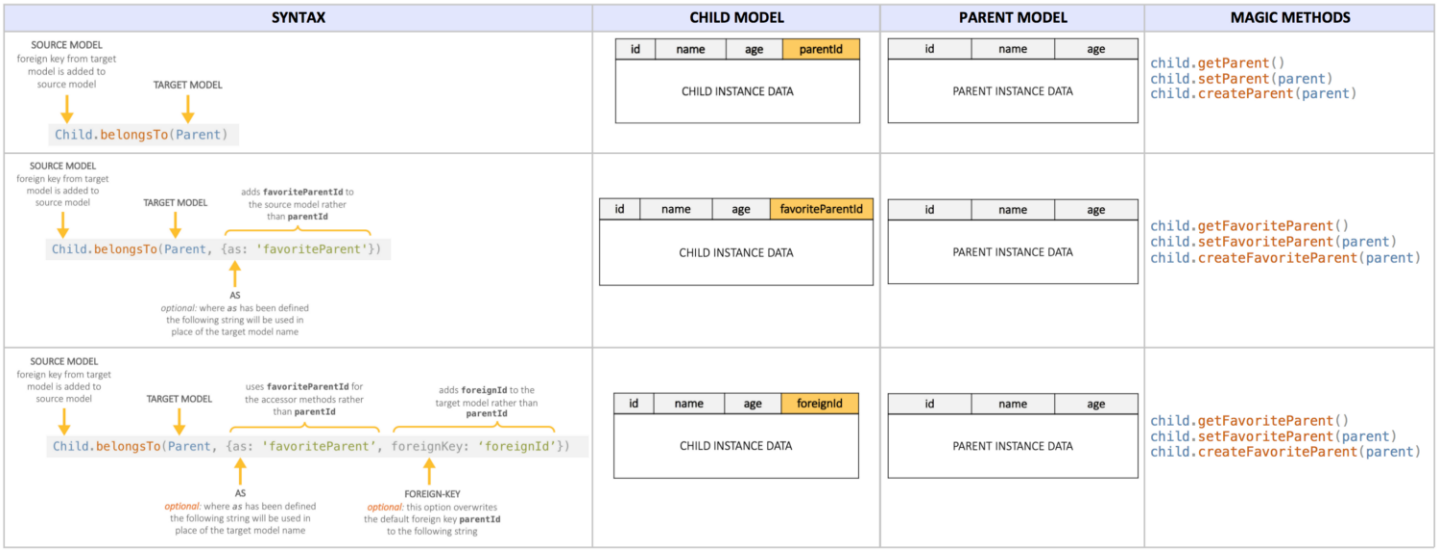
Starting Point

Base Models

For the purpose of this demonstration, I will begin with two base models — a parent and a child model. The tables below show how the 4 Sequelize associations mentioned above affect these base models, and what magic methods are created as a result.



One-To-One



One-to-One Association: Belongs To

Has One

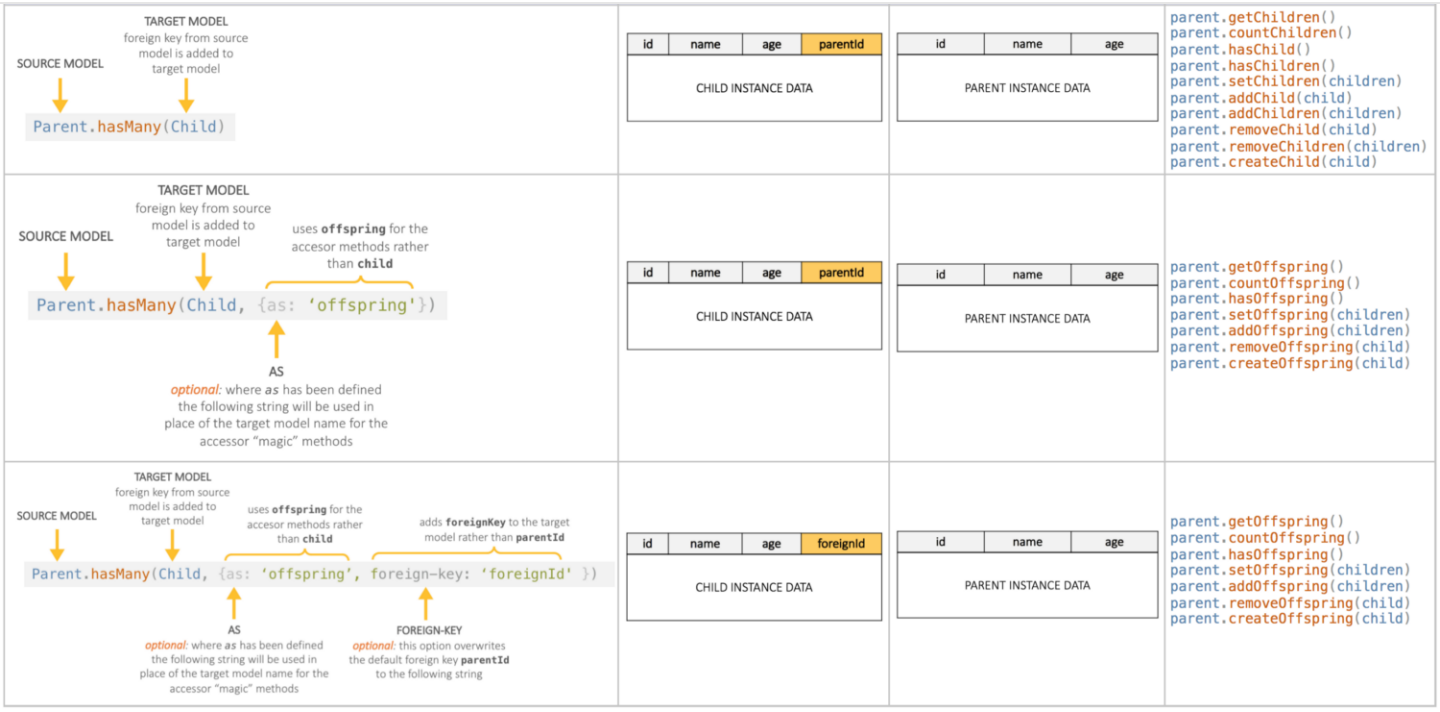
Associations where the foreign key exists on the **target model**

One-to-One Association: Has One

One-To-Many

One-to-Many associations connect one source with multiple targets. The targets however, are connected to exactly one specific source.

Has Many



One-to-Many Association — Has M

Many-To-Many

Many-to-Many associations connect one source with many targets. The targets can also be connected to many sources.

Belongs To Many

Unlike the 3 previous associations, belongsToMany associations create a new joint table with the foreign keys of the target and source models.

Conclusion

differences, Sequelize is extremely powerful in establishing relational databases and querying data.

Sequelize Relational Databases Association Magic JavaScript