How to implement many to many association in sequelize

Ask Question

I have two tables: Books and Articles with a many-to-many relationship between them. Joining table is BookArticles.

models/books.js

```
module.exports = function(sequelize, DataTypes) {
  return Food = sequelize.define("Book", {
    id: {
      type: DataTypes.INTEGER,
      primaryKey: true,
      allowNull: false,
      autoIncrement: true,
      unique: true
    }
  });
}
```

models/articles.js

```
module.exports = function(sequelize, DataTypes) {
  return Food = sequelize.define("Article", {
    id: {
      type: DataTypes.INTEGER,
      primaryKey: true,
      allowNull: false,
      autoIncrement: true,
      unique: true
    }
  });
}
```

models/bookArticles.js

```
module.exports = function(sequelize, DataTypes) {
  return Food = sequelize.define("BookArticles", {
    id: {
      type: DataTypes.INTEGER,
      primaryKey: true,
      allowNull: false,
      autoIncrement: true,
      unique: true
    },
   bookId: {
      type: DataTypes.INTEGER,
      references: 'Book',
      referencesKey: 'id',
      allowNull: false
    },
    ArticleId: {
      type: DataTypes.INTEGER,
      references: 'Article',
      referencesKey: 'id',
      allowNull: false
    },
  });
}
```

And models/index.js

```
m.BookArticles.belongsTo(m.Book);
m.Book.hasMany(m.Article, {through: m.BookArticles});


m.BookArticles.belongsTo(m.Article);
m.Article.hasMany(m.Books, {through: m.BookArticles});
```

but I could not get book articles

How can I get it ??

node.js express sequelize.js

asked Apr 9 '14 at 9:44

SpunkyLive
639 2 10 21

ht ch_qa&u

the documentation for this senario may help: docs.sequelizejs.com/class/lib/associations/… – Ulad Kasach Mar 29 at 5:22

delete BookArticles model and update relation to:

```
m.Book.hasMany(m.Article, {through: 'book_articles'});
m.Article.hasMany(m.Books, {through: 'book_articles'});
```

answered Apr 9 '14 at 10:29

ahiipsa
1,362 13 20

19 depreciated. see answers below – Damon Yuan Nov 27 '15 at 11:03

`Update 17 Feb 15` : 1. The new v2, uses 2x `.belongsToMany()` for N:M.

There have been many problems in understanding all these associations.

Generally I think we are confused as to what are the tables created, and what methods are gained by associations.

The below text is something I wrote to standardise how I want my team to handle all these. As for the naming conventions, you may ignore it if you just let Sequelize default everything.

However it is recommended to explicitly name your conventions for many reasons.

Brief:

O:O, set up a `Parent.hasOne(Child)` AND `Child.belongsTo(Parent)` .

O:M, set up `Parent.hasMany(Child)` AND `Child.belongsTo(Parent)` .

N:M*, set up `Parent.belongsToMany(Child, {through: 'Parent_Child', foreignKey: 'Parent_rowId'})` and `Child.belongsToMany(Parent, {through: 'Parent_Child', foreignKey: 'Child_rowId'})` .

Methods gained by hasOne(), hasMany() and belongsTo()/belongsToMany()

To understand why we do the above associations we start off by knowing what are the methods we gain for each model.

hasOne():

In setting a `Parent.hasOne(Child)` , methods available to `parent` DAO instance:

```
parent.getChild,
parent.setChild,
parent.addChild,
parent.createChild,
parent.removeChild,
parent.hasChild
```

hasMany():

In setting a `Parent.hasMany(Child)` , methods available to `parent` DAO instance:

```
parent.getChildren,
parent.setChildren,
parent.addChild,
parent.createChild,
parent.removeChild,
parent.hasChild,
parent.hasChildren
```

belongsTo()/belongsToMany:

In setting a `Child.belongsTo(Parent)` , methods available to `child` DAO instance:

```
child.getParent,
child.setParent,
child.createParent

//belongsToMany
child.getParents,
child.setParents,
child.createParents
```

The syntax for setting up relationships. And our conventions

For O:O, and O:M:

```
Parent.hasOne(Child, {foreignKey: 'Parent_childID'});
Child.belongsTo(Parent, {foreignKey: 'Parent_childID'});
```

Note that we explicitly defined our foreignKeys to be Parent_childID. This is because we want this PascalCase_camelCase for TableName_keyName convention.

Many to Many relationship

For a N:M relationship, do this:

```
Parent.belongsToMany( Child, {
    as: [Relationship],
    through: [Parent_Child] //this can be string or a model,
    foreignKey: 'Parent_rowId'
});

Child.belongsToMany(Parent, {
    as: [Relationship2],
    through: [Parent_Child],
    foreignKey: 'Child_rowId'
});
```

`*New in v2` : Using "through" is now a must. As a standard, using the "through" parameter, we explicitly define all the crosstable names for consistency and less gotchas.

The above will create the Parent_Child, with RelationshipId and Relationship2ID.

Sequelize can create foreignKeys automagically, but I usually define my own.

Table and keys naming conventions

TableNames: PascalCase

keys: camelCase

foreignkeys: TableNameInPascalCase_foreignKeyInCamelCase

Example: User_pictureId Meaning: This key of pictureId came from the User table.

edited Aug 11 '15 at 14:34

answered Aug 1 '14 at 3:33

Calvintwr
4,300 2 17 26

1 I have a similar prob, but i cant get it working after trying the solutions mentioned above. when i query, i get an error, but the error object is empty. Below is how my models are associated: `sequelize.define('A', {id, name}, A.belongsToMany(models.B, {through: models.A_B, foreignKey: 'a_id'}));` `sequelize.define('B', {id, name}, B.belongsToMany(models.A, {through: models.A_B, foreignKey: 'b_id'}));` `sequelize.define('A_B', {id, a_id, b_id});` `//mapping table` `A.findAll(query) .then(function (r){ }) .catch (function(e){ });` @Calvintwr @ahiipsa , could you please help ? – optimusPrime Apr 20 '15 at 11:46

I presume that you had `console.log(e)` in your `.catch()` handler. Try losing `a_id` and `b_id` in the `A_B` model. – Calvintwr Apr 20 '15 at 12:15

1 @Calvintwr awesome explanation !!! – Sunil Sharma Jan 27 '16 at 17:47

@Calvintwr thank you for this explanation. Does the N:M relationship always require a third model? What if the relationship is just a technical join like the one between User and Project? The table tracking the relationship could be called, user_projects. Does one have to create a model UserProject and if not, how does one create a migration file for this table and how would you declare the relationships in the models: User and Project? – robertjewell Oct 8 '16 at 18:00

it is best to create the model UserProject so that you can use it as well, various purposes. if you don't create the model, sequelize will automatically create a default cross table, which will still work but you cannot query this table directly, unless using raw sql. migration is never problem whether you create or not create. whether you choose to do so has to do with whether there is a higher level usage for the crosstable, such as querying it, to get useful information, or even store useful data about each cross relation. – Calvintwr Oct 11 '16 at 2:03

This is how i solved the similar problem i had two models a user model

```
var user = sequelize.define('user', {
    name: {
        Sequelize.STRING(255)
    },
    email: {
        type: Sequelize.STRING(255),
        unique: true,
        validate: {
            isEmail: true
        }
    }
});
```

and a roles model

```
var Role = sequelize.define('role', {
    name: {
        Sequelize.ENUM('ER', 'ALL', 'DL')
```

```
    },
    description: {
        type: Sequelize.TEXT
    }
});
```

Then i created the union model UserRole

```
var UserRole = sequelize.define('user_role', {
    id: {
        type: Sequelize.INTEGER,
        primaryKey: true,
        autoIncrement: true
    },
    name: {
        type: Sequelize.ENUM('Admin', 'Staff', 'Customer', 'Owner')
    }
});
```

Note: you have to explicitly define the id for UserRole otherwise sequelize will use the two foreign keys in this case `user_id` and `role_id` as your primary keys.

Then i created the belongs to many relationship as follows

```
User.belongsToMany(Role, { as: 'Roles', through: { model: UserRole, unique: false
}, foreignKey: 'user_id' });
Role.belongsToMany(User, { as: 'Users', through: { model: UserRole, unique: false
}, foreignKey: 'role_id' });
```

edited Jan 3 '17 at 1:30

Amir

4,262 2 24 41

answered Jan 24 '16 at 9:57

Buhiire Keneth

752 8 8

`M:M` relation through table `BookArticles` :

```
m.Book.belongsToMany(m.Article, {through: m.BookArticles});
m.Article.belongsToMany(m.Books, {through: m.BookArticles});
```

answered Jul 25 '16 at 11:43

Tilekbekov Yrysbek

1,184 2 8
