

public class |

# BelongsToMany

## Extends:

[Association](#) → BelongsToMany

Many-to-many association with a join table.

When the join table has additional attributes, these can be passed in the options object:

```
UserProject = sequelize.define('user_project', {
  role: Sequelize.STRING
});
User.belongsToMany(Project, { through: UserProject });
Project.belongsToMany(User, { through: UserProject });
// through is required!

user.addProject(project, { through: { role: 'manager' }});
```

All methods allow you to pass either a persisted instance, its primary key, or a mixture:

```
Project.create({ id: 11 }).then(function (project) {
  user.addProjects([project, 12]);
});
```

If you want to set several target instances, but with different attributes you have to set the attributes on the instance, using a property with the name of the through model:

```
p1.UserProjects = {
  started: true
}
user.setProjects([p1, p2], { through: { started: false }}) //
The default value is false, but p1 overrides that.
```

Similarly, when fetching through a join table with custom attributes, these attributes will be available as an object with the name of the through model.

```
user.getProjects().then(function (projects) {
  let p1 = projects[0]
  p1.UserProjects.started // Is this project started yet?
})
```

In the API reference below, add the name of the association to the method, e.g. for `User.belongsToMany(Project)` the getter will be `user.getProjects()`.

## See:

[Model.belongsToMany](#)

## Method Summary

Public Methods	
public	<a href="#">add</a> (newAssociation(s): <a href="#">Model</a> []   <a href="#">Model</a>   string[]   string   number[]   Number, options: Object): Promise Associate one ore several rows with <code>this</code> .

public	<code>count(options: Object): Promise&lt;Integer&gt;</code> Count everything currently associated with this, using an optional where clause.
public	<code>create(values: Object, options: Object): Promise</code> Create a new instance of the associated model and associate it with this.
public	<code>get(options: Object): Promise&lt;Array&lt;Model&gt;&gt;</code> Get everything currently associated with this, using an optional where clause.
public	<code>has(instance(s): Model[]   Model   string[]   String   number[]   Number, options: Object): Promise&lt;boolean&gt;</code> Check if one or more instance(s) are associated with this.
public	<code>remove(oldAssociated: Model   String   Number, options: Object): Promise</code> Un-associate one or more instance(s).
public	<code>set(newAssociations: Array&lt;Model String Number&gt;, options: Object): Promise</code> Set the associated models by passing an array of instances or their primary keys.

## Inherited Summary

► From class **Association**

## Public Methods

`public add(newAssociation(s): Model[] | Model | string[] | string | number[] | Number, options: Object): Promise`

Associate one ore several rows with `this` .

### Params:

Name	Type	Attribute	Description
<code>newAssociation(s)</code>	<code>Model[]   Model   string[]   string   number[]   Number</code>	optional	A single instance or primary key, or a mixed array of persisted instances or primary keys
<code>options</code>	<code>Object</code>	optional	Options passed to <code>through.findAll</code> , <code>bulkCreate</code> and <code>update</code>
<code>options.validate</code>	<code>Object</code>	optional	Run validation for the join model.
<code>options.through</code>	<code>Object</code>	optional	Additional attributes for the join table.

`public count(options: Object): Promise<Integer>`

Count everything currently associated with this, using an optional where clause.

**Params:**

Name	Type	Attribute	Description
options	Object	optional	
options.where	Object	optional	An optional where clause to limit the associated models
options.scope	String   Boolean	optional	Apply a scope on the related model, or remove its default scope by passing false

**Return:**

Promise<Integer>

`public create(values: Object, options: Object): Promise`

Create a new instance of the associated model and associate it with this.

**Params:**

Name	Type	Attribute	Description
values	Object	optional	
options	Object	optional	Options passed to create and add
options.through	Object	optional	Additional attributes for the join table

**Return:**

Promise

`public get(options: Object): Promise<Array<Model>>`

Get everything currently associated with this, using an optional where clause.

**Params:**

Name	Type	Attribute	Description
options	Object	optional	
options.where	Object	optional	An optional where clause to limit the associated models
options.scope	String   Boolean	optional	Apply a scope on the related model, or remove its default scope by passing false

options.schema	String	optional	Apply a schema on the related model
----------------	--------	----------	-------------------------------------

**Return:**

Promise<Array<[Model](#)>>

**See:**

[Model.findAll](#) for a full explanation of options

public has(instance(s): [Model](#)[] | [Model](#) | string[] | String | number[] | Number, options: Object): Promise<boolean>

Check if one or more instance(s) are associated with this. If a list of instances is passed, the function returns true if *all* instances are associated

**Params:**

Name	Type	Attribute	Description
instance(s)	<a href="#">Model</a> []   <a href="#">Model</a>   string[]   String   number[]   Number	optional	Can be an array of instances or their primary keys
options	Object	optional	Options passed to <code>getAssociations</code>

**Return:**

Promise<boolean>

public remove(oldAssociated: [Model](#) | String | Number, options: Object): Promise

Un-associate one or more instance(s).

**Params:**

Name	Type	Attribute	Description
oldAssociated	<a href="#">Model</a>   String   Number	optional	Can be an Instance or its primary key, or a mixed array of instances and primary keys
options	Object	optional	Options passed to <code>through.destroy</code>

**Return:**

Promise

public set(newAssociations: Array<[Model](#)|String|Number>, options: Object): Promise

Set the associated models by passing an array of instances or their primary keys. Everything that it not in the passed array will be un-associated.

**Params:**

Name	Type	Attribute	Description
------	------	-----------	-------------

Home

Reference

Join us on Slack

Deterrable

Op

QueryTypes

TableHints

associations

Association

BelongsToMany

BelongsTo

HasMany

HasOne

errors

AccessDeniedError

AssociationError

BaseError

BulkRecordError

ConnectionError

ConnectionRefusedError

ConnectionTimedOutError

DatabaseError

EagerLoadingError

EmptyResultError

ExclusionConstraintError

ForeignKeyConstraintError

HostNotFoundError

HostNotReachableError

InstanceError

InvalidConnectionError

OptimisticLockError

QueryError

SequelizeScopeError

TimeoutError

UniqueConstraintError

UnknownConstraintError

ValidationError

ValidationErrorItem

newAssociations

Array<[Model](#)|String|Number>

Search...

optional

An array of persisted instances or primary key of instances to associate with this. Pass `null` or `undefined` to remove all associations.

options

Object

optional

Options passed to `through.findAll`, `bulkCreate`, `update` and `destroy`

options.validate

Object

optional

Run validation for the join model

options.through

Object

optional

Additional attributes for the join table.

Return:

Promise

Generated by [ESDoc\(0.5.2\)](#)

