



- C Model
- C QueryInterface
- C Sequelize
- C Transaction
- V DataTypes
- V Deferrable
- V Op
- V QueryTypes
- V TableHints

associations

- C Association
- C BelongsToMany
- C BelongsTo
- C HasMany
- C HasOne

errors

- C AccessDeniedError
- C AssociationError
- C BaseError
- C BulkRecordError
- C ConnectionError
- C ConnectionRefusedError
- C ConnectionTimedOutError
- C DatabaseError
- C EagerLoadingError
- C EmptyResultError
- C ExclusionConstraintError
- C ForeignKeyConstraintError
- C HostNotFoundError
- C HostNotReachableError
- C InstanceError
- C InvalidConnectionError
- C OptimisticLockError
- C QueryError
- C SequelizeScopeError
- C TimeoutError
- C UniqueConstraintError
- C UnknownConstraintError
- C ValidationError
- C ValidationErrorItem

public class |

HasMany

Extends:[Association](#) → HasMany

One-to-many association

In the API reference below, add the name of the association to the method, e.g. for `User.hasMany(Project)` the getter will be `user.getProjects()` . If the association is aliased, use the alias instead, e.g. `User.hasMany(Project, { as: 'jobs' })` will be `user.getJobs()` .

See:[Model.hasMany](#)

Method Summary

Public Methods

public	add (newAssociation(s): Model [] Model string[] string number[] number, options: Object): Promise Associate one or more target rows with <code>this</code> .
public	count (options: Object): Promise<Integer> Count everything currently associated with this, using an optional where clause.
public	create (values: Object, options: Object): Promise Create a new instance of the associated model and associate it with this.
public	get (options: Object): Promise<Array< Model >> Get everything currently associated with this, using an optional where clause.
public	has (instance(s): Model [] Model string[] String number[] Number, options: Object): Promise Check if one or more rows are associated with <code>this</code> .
public	remove (oldAssociatedInstance(s): Model [] Model String[] string Number[] number, options: Object): Promise Un-associate one or several target rows.
public	set (newAssociations: Array< Model String Number>, options: Object): Promise Set the associated models by passing an array of persisted 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 or more target rows with `this` . This method accepts a `Model` / `string` / `number` to associate a single row, or a mixed array of `Model` / `string` / `numbers` to associate multiple rows.

Params:

Name	Type	Attribute	Description
newAssociation(s)	Model[] Model string[] string number[] number	optional	
options	Object	optional	Options passed to <code>target.update</code> .

Return:

Promise

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 <code>target.create</code> .

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
```

Check if one or more rows are associated with `this`.

Params:

Name	Type	Attribute	Description
instance(s)	Model [] Model string[] String number[] Number	optional	
options	Object	optional	Options passed to <code>getAssociations</code>

Return:

Promise

```
public remove(oldAssociatedInstance(s): Model[] | Model | String[] | string | Number[] | number, options: Object): Promise
```

Un-associate one or several target rows.

Params:

Name	Type	Attribute	Description
oldAssociatedInstance(s)	Model [] Model String[] string Number[] number	optional	

options

Object

optional

Options
passed to
`target.update`**Return:**

Promise

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

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

Params:

Name	Type	Attribute	Description
newAssociations	Array<Model String Number>	optional	An array of persisted instances or primary key of instances to associate with this. Pass <code>null</code> or <code>undefined</code> to remove all associations.
options	Object	optional	Options passed to <code>target.findAll</code> and <code>update</code> .
options.validate	Object	optional	Run validation for the join model

Return:

Promise