Có 2 cách để defined FK

1. Cung cấp trực tiếp (auto, manually)
2. Bằng alias

Cách 1 : Mặc định

Ship.belongsTo(Captain); // This creates the `captainId` foreign key in Ship.

Ship.belongsTo(Captain, { foreignKey: 'bossId' });

console.log((await Ship.findAll({ include: Captain })).toJSON());

console.log((await Ship.findAll({ include: 'captain' })).toJSON());

const ship = Ship.findOne();

console.log((await ship.getCaptain()).toJSON());

Cách 2 : Alias

Sử dụng alias khá hữu ích trong 1 số trường hợp khi model name không lột tả hết

VD : Mail & Person (models)

* mail.getPerson() sẽ hơi khó hiểu.
* mail.getSender() and mail.getReceiver() sẽ dễ hiểu hơn rất nhiều

Ship.belongsTo(Captain, { as: 'leader' });

// This creates the `leaderId` foreign key in Ship.

Ship.belongsTo(Captain, { as: 'leader', foreignKey: 'bossId' });

// This creates the `bossId` foreign key in Ship.

// Not work 🡪 Throws an error

console.log((await Ship.findAll({ include: Captain })).toJSON());

// Work

console.log((await Ship.findAll({ include: 'leader' })).toJSON());

console.log((await Ship.findAll({

include: {

model: Captain,

as: 'leader'

}

})).toJSON());

const ship = Ship.findOne();

console.log((await ship.getLeader()).toJSON());

\*Nếu quan hệ 1 – 1 🡪 ‘leader’ còn nếu 1 – N 🡪 ‘leaders’

## Special Methods

Task.hasOne(User, { as: 'Author' });

* taskInstance.getAuthor()
* taskInstance.setAuthor()
* taskInstance.createAuthor()

Gồm :

Foo.hasOne(Bar) = Foo.belongsTo(Bar)

* fooInstance.getBar()
* fooInstance.setBar()
* fooInstance.createBar()

Foo.hasMany(Bar) = Foo.belongsToMany(Bar, { through: Baz })

* fooInstance.getBars()
* fooInstance.countBars()
* fooInstance.hasBar()
* fooInstance.hasBars()
* fooInstance.setBars()
* fooInstance.addBar()
* fooInstance.addBars()
* fooInstance.removeBar()
* fooInstance.removeBars()
* fooInstance.createBar()

## Tạo FK cho Non-PK field

Lưu ý : field này phải có unique constraint trên nó

### belongsTo

Ship.belongsTo(Captain, { targetKey: 'name', foreignKey: 'captainName' });

Sẽ references captainName(of Ship - FK) 🡪 name(of Captain - field)

Giờ thì

await Captain.create({ name: "Jack Sparrow" });

const ship = await Ship.create({ name: "Black Pearl", captainName: "Jack Sparrow" });

console.log((await ship.getCaptain()).name); // "Jack Sparrow"

hasOne & hasMany

Foo.hasOne(Bar, { sourceKey: 'name', foreignKey: 'fooName' });

Bar.hasMany(Baz, { sourceKey: 'title', foreignKey: 'barTitle' });

belongsToMany

4 trường hợp :

1. PrimaryKey\_REF cho cả 2 models

Foo.belongsToMany(Bar, { through: 'foo\_bar' });

1. Non-PK field cho MODEL\_B(target), not for A(source)

Foo.belongsToMany(Bar, { through: 'foo\_bar', targetKey: 'title' });

1. Non-PK field cho MODEL\_A(source), not for B(target)

Foo.belongsToMany(Bar, { through: 'foo\_bar', sourceKey: 'name' });

1. Non\_PK\_REF cho cả 2 models

Foo.belongsToMany(Bar, { through: 'foo\_bar', sourceKey: 'name', targetKey: 'title' });