Waterline

Sails.js Cheat Sheet

Ordem	Finalidade	Comando	Observação
As configurações de armazenamento estão localizadas no arquivo 'config/datastores.js' e podem ser acessadas pelo código como sails.config.datastores.<nome dicionário>.			

1.	Localiza lista de registros que atendam ao critério.	`await Modelo.find(critério);`
		
		ou
		
		`await Modelo.find({`
		`  where: {id: 42},`
		`  select: ['name','email']`
		`});`
			O critério deve ser um objeto contendo um campo válido ou expressão. Ex.: {id: 42}.
			
			As respostas sempre serão JavaScript Promisses, portanto, pode-se utilizar tanto o await quanto o then().catch()

2.	Localiza um único registro que atenda ao critério.	`await Modelo.findOne({});`	Retorna *undefined* se nenhum registro correspondente for encontrado.

3.	Tenta localizar, mas cria um novo registro se não existir.	`await Modelo`
		`.findOrCreate(`
		`  critério,`
		`  valoresIniciais`
		`);`
		
		Ou
		
		`Modelo`
		`.findOrCreate(`
		`  critério,`
		`  valoresIniciais)`
		`  .exec(`
		`async((err, dado, criado)=>`
		`{ if(criado)console.log('OK');`
		`}`
		`);`
			A função **exec()** retorna três valores para o *call-back*: erro (se houver ou undefined); o dado retornado (se for erro, retorna *undefined*) e um flag booleano informando se foi necessário criar o registro ou não.

4.	Localiza e retorna um conjunto de dados que atendam ao critério, em formato *Stream*	`await Modelo.stream(criteria)`
		`.eachRecord(async`
		`(records)=>...`
		`Ou`
		`.eachBatch(async`
		`(records)=>...`
			Isso é útil para trabalhar com conjuntos de resultados muito grandes, do tipo que pode transbordar a RAM disponível do servidor se você tentar manter todo o conjunto na memória ao mesmo tempo.

5.	Faz a validação de um valor para um determinado campo/ atributo	Modelo.validate(atributo, valor);	É um método assíncrono, portanto não necessita de await ou then(). Vide regras de validação mais abaixo

6.	Conta quantos registros atendem ao critério	await Modelo.count(criterio);	
7.	Cálculo de Valor Médio de valores nos registros que atendem ao critério	await Modelo.avg(atributo, criterio);	O atributo deve ser do tipo numérico
8.	Cálculo de Valor Somado de valores os registros que atendem ao critério	await Modelo.sum(atributo, criterio);	O atributo deve ser do tipo numérico
9.	Cria um ou mais registros no banco de dados	await Modelo.create(valoresIniciais); await Modelo .createEach([valoresIniciais])	**valoresIniciais** corresponde a um objeto contendo os pares atributo: valor do JavaScript.
10.	Remove um ou mais registros do banco de dados	await Modelo.destroyOne(criterio) await Modelo.destroy(criterio)	**destroyOne()** retornará erro se mais de um objeto atender ao critério;
11.	Atualiza um ou mais registros do banco de dados	await Modelo .updateOne(criterio, valores) await Modelo.update(criterio, valores)	**updateOne()** retornará erro se mais de um objeto atender ao critério; **valores** corresponde ao objeto contendo pares atributo:valor;

Dicas de Uso
Use: Modelo.find({campo: criterio}).skip(numero).limit(numero) para incorporar paginação
Use: Modelo.[**update() | create() | createEach() | destroy()**].fetch() para recuperar o dado imediatamente.
Use: Modelo.[método()].intercept(filtro, callback) para capturar e manipular erros de banco de dados
Use: Modelo.[método()].tolerate(filtro ou callback) para tolerar um erro e retornar um novo valor.
Use: Modelo.find({}).decrypt() para decriptografar dados armazenados em formato encriptado
Use: Modelo.find({}).populate('nome_do_atributo_que_representa_a_colecao') para popular coleções. Ex. Proprietario.find().populate('carros')

Regras de Validação de Campos de Tabela / Atributos de Objeto Model	
Nome	Função:
type	Define o tipo da variável armazenada. Pode ser: string, number, Boolean, json e ref
required	Define se o campo é obrigatório (true) ou não (false). Pode existir com valor nulo
allowNull	Define se o campo obrigatório aceitará valores nulos. O padrão é false
defaultsTo	Valor atribuído como padrão, se omitido na criação
isIn	Verifica se o valor informado está contido na listagem.ex ['boring','happy','sad']
unique	Define que o valor atribuído deve ser único (true) na tabela.
columnName	Define o nome do campo da tabela que corresponde ao atributo do objeto
columnType	Define o tipo específico atributo do Modelo para o campo na tabela. Ex. 'CHAR(5)'
autoIncrement	Quando um registro é criado, o valor numérico do campo é incrementado de 1.
encrypt	Define se o campo deve ser Encriptado antes de ser armazenado no banco
tableName	Define o nome da tabela do banco de dados que corresponde ao objeto
Ex.:	nome: { type: 'string', required: true, unique: true, columnName: 'NM_USER'}, pais: { type: 'string', required: false, defaultsTo: 'brasileiro', encrypt: true }, nascimento: { type: 'number', columnType: 'date', columnName: 'DT_NASC' }