

ORM

Object relational maps

Syntactic sugar for using a database

`Model.find(name: 'rob')` rather than `SELECT * from models WHERE name='rob';`

Define relations between objects and not worry about the SQL

Example using Bookshelf:

```
var User = bookshelf.Model.extend({  
  tableName: 'users',  
  messages: function() {  
    return this.hasMany(Posts);  
  }  
});
```

this way, we can find a User and just ask for its posts.

We can also define lots of other relations

heres how you define a hasOne using node-orm2

```
Car.hasOne('engine', Engine).
```

Now we're given a lot methods on our car models

```
Car.getEngine(function..) // Gets owner
Car.setEngine(person, function..) // Sets owner_id
Car.hasEngine(function..) // Checks if owner exists
Car.removeEngine() // erases owner
```

Different ORMS

- Bookshelf.js - based on the knex SQL query builder no type definition, use `checkit`
- Sequelize - gives you migrations!
- WaterlineORM - works with everything, postgres, mysql, mongoDB - no transactions
- node-orm2

Using ORMS

- usually need to install a specific database driver
- specify a connection string or a connection object that you pass to the ORM
- connect
- sync/migrate