

The ORM

Activity: Quotes App (15 min)

Review: Quotes App

ORM

OMG, NAA





Any questions?

Object Relational Mapping

So much to code, so little time

What is DRY?

What is DRY?

What is ORM?

A system which allows programmers to easily collect and manipulate data from databases using reusable methods

Demo: ORM

- `$ atom 16-OrmExample/server.js`
- `$ atom 16-OrmExample/config/connection.js`
- `$ atom 16-OrmExample/config/orm.js`
- What's new here?

Review: ORM

- All of our database-connection code is contained within `connection.js`, which is then required in `orm.js`, which is then required in `server.js`.
- `orm.js` contains ALL of our MySQL queries inside of it as methods within an object referred to as "orm."
- These methods take in variables which are then used to alter the properties of our queries. In other words, we can now make similar queries to different MySQL tables, columns, and rows without having to write out entirely new MySQL commands every time. Instead, all we have to do is change around the variables we pass into the method we are calling upon.

Activity: ORM (10 min)

Discuss 16-OrmExample with your friends and neighbors

- What are the pros of ORM?
- What are situations in which an ORM would be considered valuable?

Activity: Party Database App (20 min)

See 17-PartyDatabase for instructions

Review: Party Database App

Split-Up

```
$ atom 17-PartyDatabase/Solved/server.js
```

```
$ atom 17-PartyDatabase/Solved/config/orm.js
```

```
orm.leftJoin("", "clients", "parties", "id", "client_id");
```

```
...
```

```
leftJoin: function(whatToSelect, tableOne, tableTwo, onTableOneCol,  
onTableTwoCol) {
```

```
  var queryString = "SELECT ?? FROM ?? AS tOne";
```

```
  queryString += " LEFT JOIN ?? AS tTwo";
```

```
  queryString += " ON tOne.?? = tTwo.??";
```

```
  console.log(queryString);
```

```
  connection.query(queryString, [whatToSelect, tableOne, tableTwo,  
onTableOneCol, onTableTwoCol], function(err, result) {
```

```
    console.log(result);
```

```
  });
```

```
}
```

Select everything from the clients table and call it 'tOne', then left join the parties table and call it 'tTwo'.

Finally, find the matches in tOne's id column and tTwo's client_id column.

Demo: The Asynchronous Problem

Open up 18-TheAsynchProblem/server.js and
18-TheAsynchProblem/config/orm.js alongside one another

```
$ node server.js
```

Why is data undefined?

Our query is asynchronous to the rest of
our JavaScript code.

Our server is not waiting for a response
from the database before running our
console.log command.

Big problem

Simple solution

Activity: The Asynchronous Problem (10 min)

With a partner

- explain the asynchronous nature of this example
- find a solution

Give the ORM a callback

Demo: The Asynchronous Solution

Side-by-side:

- 20-TheAsynchSolution/server.js
- 20-TheAsynchSolution/config/orm.js

Demo: The Asynchronous Solution

- `orm.js`:
 - we added in a `cb` argument to the `selectWhere` function's arguments list.
 - We also executed `cb` (because we're expecting it to be a function) and passed it the data returned from the MySQL query.
- `server.js`:
 - The 4th argument of `orm.selectWhere` is an anonymous function with `res` passed into it as an argument.
 - That function gets sent to `orm.selectWhere` along with `parties`, `party_type`, and `grown-up`

Demo: The Asynchronous Solution

There are a number of important things occurring in the ORM object:

- The `selectWhere` method inside `orm.js` builds the query which is then stored within the `queryStringvariable`.
- `queryString` gets passed to `connection.query` as well as `[valOfCol]`, which gets placed inside the question mark part of the `queryString`.
- After we get the data back from `connection.query`, we pass result to `cb` and execute it as `cb(result)`.
- The variable `res` now equals `result` and `res` gets console logged within `server.js`.

Activity: The Asynchronous Solution

With a partner, discuss 20-TheAsynchSolution

Activity: Lunch

Read `cooking.md` for a refresher on callbacks

What is an anonymous function?

A function without a name

Functions stored in variables do not need function names. They are always invoked (called) using the variable name.

What is a callback?

A callback function is a function passed into another function as an argument, which is then invoked inside the outer function to complete some kind of routine or action

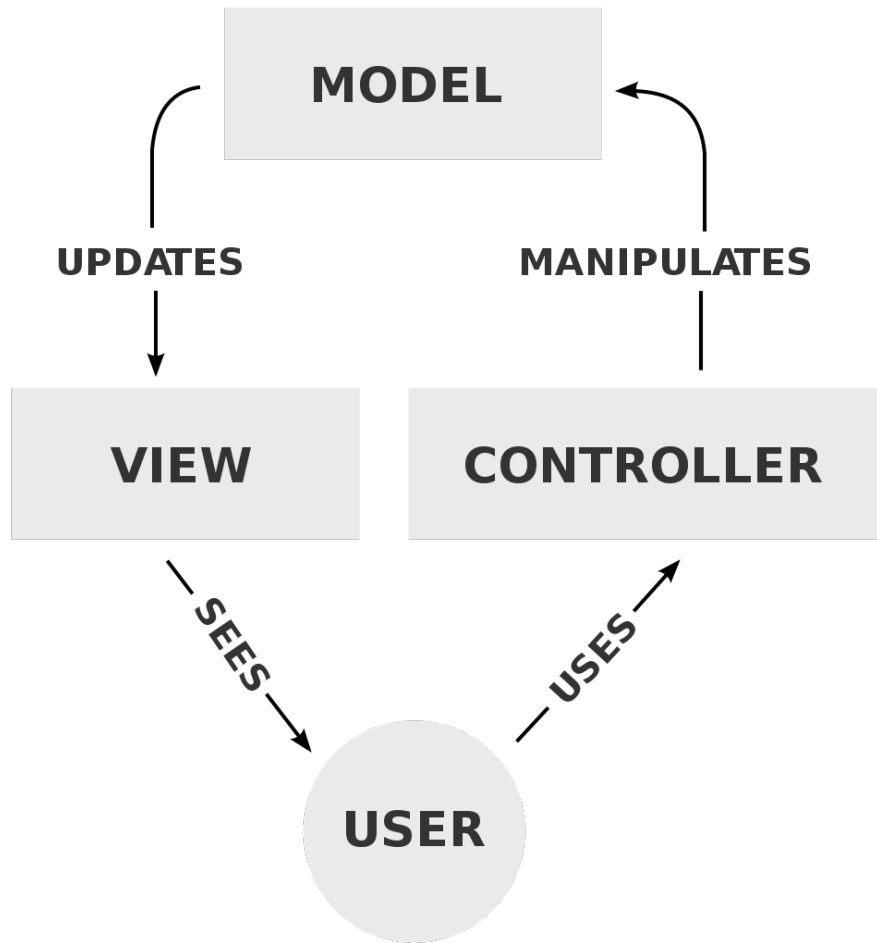
Demo/Review/Activity: 19-CallbackReview

MVC



Model View Controller

Creates a folder/file-system that is both
easy to navigate and easy to
understand.



RTFM <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

Demo: Model View Controller

Open 21-MvcExample

<http://localhost:3000/>

MVC is a framework for building web applications:

- The Model represents the application core (for instance a list of database records).
- The View displays the data (the database records).
- The Controller handles the input (to the database records).

Activity: MVC

With a partner, discuss MVC and find examples in other programming languages

Handlebars {{#if}}

The {{#if}} helper does just what you would expect it to do. It allows you to implement an if block into your Handlebars code.

- The {{#if}} helper outputs the block that it contains if the value given to it is truthy.
- The tricky bit here is that you cannot use conditionals within these statements. In other words, you are checking to see whether or not a specific variable exists or whether it contains a value inside of it.

Handlebars {{#unless}}

The {{#unless}} statement is pretty much the exact opposite of the {{#if}} statement since it will run the block of code contained inside of it if the given expression is false.

RTFM http://handlebarsjs.com/builtin_helpers.html

Activity: Cats App Problem (20 min)

See 22-CatsAppProblem for instructions

Review: Cats App Problem

Homework

Heroku + MySQL

Setting up a remote database for the first time will take about an hour.

See **MySQLHerokuDeploymentProcess.pdf** for instructions