MySQL

The Sequel

Objectives

- To introduce Sequelize as a complex tool that simplifies MySQL
- To understand the makeup of a Sequelize directory, how to make one, and how to make queries using Sequelize
- To set up a project to work with Sequelize
- To create models using Sequelize
- To use Sequelize CRUD methods for updating and deleting database entries

#parkinglot

Demo: Chirpy

http://localhost:8080/

Activity: Chirpy (20 min)

See 01-Chirpy-mySQL for files and instructions

Review: Chirpy

- We created the database and the table
- In config/connections.js, we use the mysql package to connect to the db
- In routes/api-routes.js, we handle requests and create mySQL queries

Quiz O’Clock

What is full stack web development?

What is programming?

Is it hard?


such sigh
phantompiper.tumblr.com

Sequelize

http://docs.sequelizejs.com/

Sequelize is a premade ORM that simplifies database queries in Node applications, allowing us to do complex data management with simple JavaScript methods

Activity: Sequelize Quiz (15 min)

With a partner, answer the questions in 02-SequelizeQuestions

Use the Sequelize Docs and Google Fu

What is Sequelize?

A promise-based ORM for Node.js

Promise?

An object representing the eventual completion or failure of an asynchronous operation

.then()

RTFM https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise

What advantages does Sequelize offer?

- Easy to test
- Gives you support for syncing databases
- Validation (restricts to specific type of data)
- Complex SQL queries in simple syntax

How do I add Sequelize to my app?

```
npm install --save sequelize
```

What is a Sequelize model?

A representation of table data

How would I model this table in Sequelize?

How would I query for all the records where the Independence Year was less than 50 years ago?

How would I query the table, order it by descending Independence Years, and limit the results to just show 2 of the records. Skipping the first two? (i.e. Results: Zambia, Afghanistan

Bonus:
How do I use sequelize to make changes to an existing table with data in it?

Use Sequelize migrations from the command line

RTFM http://docs.sequelizejs.com/en/latest/docs/migrations/

Demo: Chirpy, the Sequel

- config/connections.js
- models/chirpy.js
- routes/api-routes.js

config/connections.js

- In the connection.js file we require the Sequelize package, and use it to create a connection to our database. This is very similar to how we use the MySQL package.

Sequelize model === MySQL table

models/chirp.js

- We first require the Sequelize library (big ‘S”), and then the db connection (little "s")
- We use .define() to create a table “chirps” and its columns and datatypes
- Then we .sync() our model with the database
    - Sequelize will create a table in our database for each model if it does not already exist.
- We don't need to define an id column in our Sequelize models. By default, we're given a NOT NULL, auto-incrementing id of type INTEGER to serve as the primary key.

routes/api-routes.js

- We first require the Chirp model, then we use sequelize query methods to query our database.
  - In our example we are using Model.findAll() and Model.create() but there are many more!
    - http://docs.sequelizejs.com/manual/tutorial/models-usage.html#-findall-search-for-multiple-elements-in-the-database
    -
- Instead of writing out an entire MySQL query string, we can use predefined query methods defined by Sequelize. This allows us to greatly simplify how we retrieve data from our database!

Activity: Star Wars (20 min)

See 04-StarWars/server.js for instructions

Review: Star Wars

Demo: Sequelize Library

Add a few books…

Activity: Sequelize Library (25 min)

See 05-SequelizeLibrary

This app will require you to research more advanced sequelize querying.

Use the Sequelize documentation to find the answers

Review: Sequelize Library

Lunch

Activity: Sequelize CLI

See Slack and/or class repo for Sequelize Quick Start Guide

NOTE: The Sequelize CLI has a lot of functionality, including creating models via the CLI. We will only be using be using the CLI to initialize an index.js file, and a config.json file.

Do not create models with the CLI at this point.

It will create far more trouble than it's worth later in the week when our models become more complex if you don’t have experience writing them by hand.

Activity: ORM to Sequelize (30 min)

See 08-ORM-to-Sequelize for files and instructions

Review: ORM to Sequelize

- Note difference between connections.js and config.json files
- See also todo.js
- We sync() our models in the server.js file before we start our server.
  - This is what ensures our tables are created and ready when we start Express
  - The Todos table is created when we run our server and sync
- Note: our models are required inside of the api-routes.js file
  - We will use them in the next activity

Activity: findAll & create (15 min)

See 09-Sequelize-Create-Read-Unsolved

Review: findAll & create

Note how in the api-routes.js file, the db.Todo.create method takes in an object with the values we want our new Todo to have as an argument.


WOW

Demo: Update & Delete

- $ node 10-Sequelize-Update-Delete/server.js
- Note: we can delete todo items by clicking the x button on the todo. Refresh!
- Note: when clicking a todo item, you can update the todos text. After editing, hit "Enter" to finish editing, or click anywhere else on the page to cancel editing.
- Also note: how clicking the check mark toggles a todo's complete property.

Activity: Update & Delete (20 min)

See 10-Sequelize-Update-Delete

Review: Update & Delete

Note: .destroy() and .update() in api-routes.js