KNEX.JS

Objectives

- Understand what Knex.js is
- Explain why we use Knex.js
- Represent rows of a database in JavaScript
- Create, read, update, and delete rows from a database within a server-side application
- Respond, retrieve, and format rows from multiple tables

What is Knex?

- It is a 3rd Party JavaScript library
- Builds SQL Commands (Database Queries)
- Sends database queries to PostgreSQL Database
- Catches Errors
- prevents against SQL injection attacks
- Hosted on GitHub

First some structure

```
knex('movies')
.then((result) => {
  console.log(result);
  knex.destroy();
})
.catch((err) => {
  console.error(err);
  knex.destroy();
  process.exit(1);
});
```

Get your pens out, we are making a checklist!

1. Install with NPM
   - `npm install --save knex`
2. Install Node Module to use PostgreSQL database
   - `npm install pg --save`
3. Touch knexfile.js
   -
```
module.exports = {
development : {
  client: 'pg',
  connection: 'postgres://localhost/movie_junkie_dev'
}
  };
```

Get your pens out, we are making a checklist!

1. Touch knex.js
   - define environment variable
   - require knexfile.js and connect environment variable
   - require knex.js

```
'use strict';

const env = 'development';
const config = require('./knexfile.js')[env];
const knex = require('knex')(config);

module.exports = knex;
```

Get your pens out, we are making a checklist!

1. Require Knex in your Route files

- To close connection:
    + knex.destroy()
    + res.send(results)
- To catch errors:
    + .catch(err) => {console.log('error');
    + next(err) => {console.log('error');

Checklist!

1. Add a Route
2. Insert table name
3. Build out your code!

```
router.get ('/' , (req, res, next) => {

  knex('tableName')
  //query syntax will go here!

    .then((results) => {
      res.send(results);
      })

    .catch((err) => {
      next(err);
  });
});
```

So what's with the SQL

- Chain in query language inside of your code
  - .select
  - .insert
  - .update
  - .delete
- What does this remind you of?!?!

JIGSAW ACTIVITY

- Break into groups of 4
- Each person will take one of these topics
  - select
  - insert
  - update
  - delete
- Objectives
  - functionality
  - Arguments
  - return anything special
  - does '*' apply?
  - Whiteboard example

Joins?


THAT IS BRAND NEW INFORMATION!
reservationatdorsie.tumblr.com

Joins?

```
const express = require('express');
const router = express.Router();
const knex = require('./knex');

router.get('/', (req,res,next)=>{
  knex('table')
  .select('*')
  .join('users', 'table.users_id', 'users.id' )
  .then(result=>{
    console.log(result)
  })
})
```

```
=# SELECT *
-# FROM table
-# INNER JOIN users ON table.users_id = users.id;
```

Objectives

- Understand what Knex.js is
- Explain why we use Knex.js
- Represent rows of a database in JavaScript
- Create, read, update, and delete rows from a database within a server-side application
- Respond, retrieve, and format rows from multiple tables

Links/Resources

Knex Query Lab

Knex.js Docs

Please go check out how to Insert, Update, and Delete data from multiple tables based on user input

As well as configure your environment for production