Heroku Deployment Process

For MySQL Projects

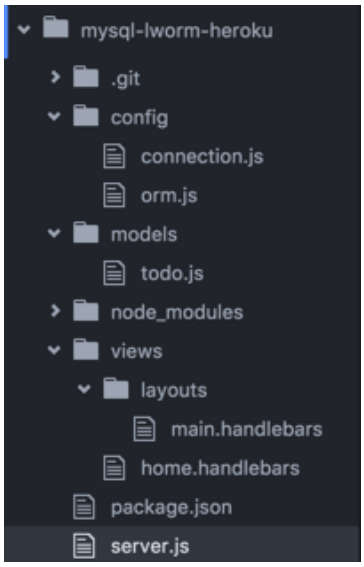
Before you begin, make sure you've installed
the MySQL NPM package.

server.js | package.json | main.handlebars

```
{
  "name": "mysql-sequelize-heroku",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Albert B",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.15.0",
    "express": "^4.13.4",
    "express-handlebars": "^3.0.0",
    "mysql": "^2.10.2"
  }
}
```

Part One: Creating a JawsDB Remote Database

1. On your local machine, navigate to your project folder. At this point, we'll assume you've been pushing/pulling your code with GitHub but have yet to deploy it to Heroku.



IMPORTANT
Make sure you have all the files you need
to interact with a basic MySQL database.

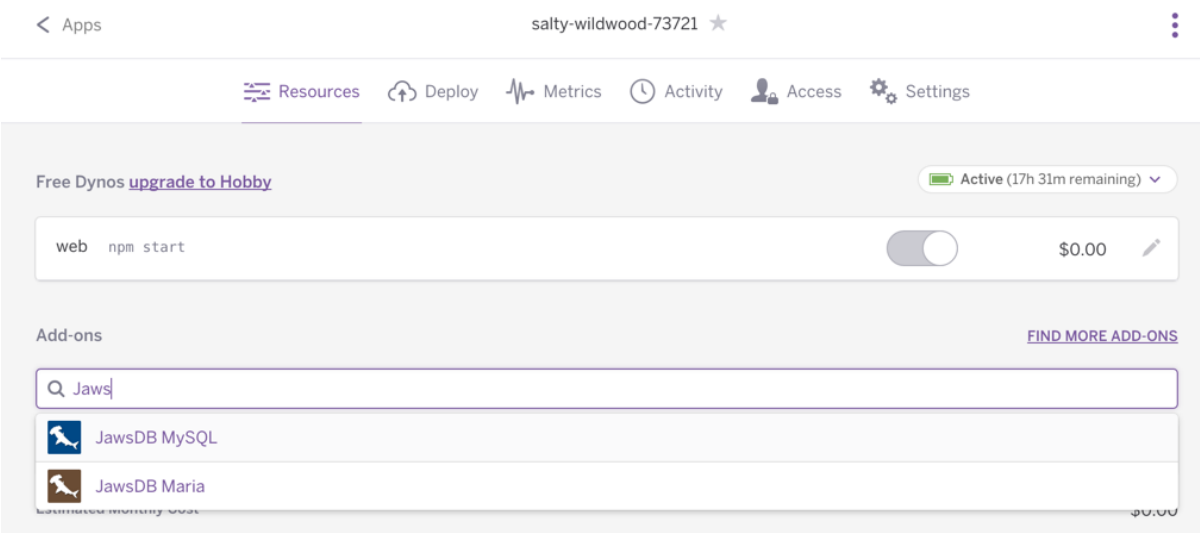
2. Run `heroku create` to connect your repo with Heroku.

```
▶ heroku create
Enter your Heroku credentials.
Email: albert.bahia1@gmail.com
Password (typing will be hidden):
Logged in as albert.bahia1@gmail.com
Creating app... ⬢ salty-wildwood-73721
https://salty-wildwood-73721.herokuapp.com/ | https://git.heroku.com/salty-wildwood-73721.git
```

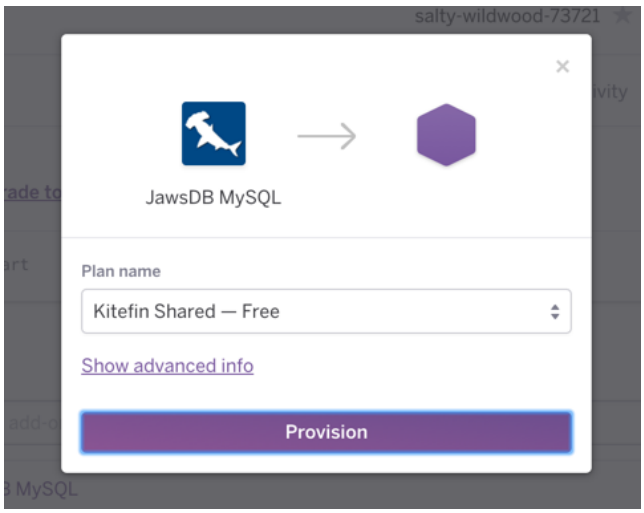
3. Navigate to heroku.com and login with your credentials.

4. Find your Heroku app's name in the dashboard. Click on it.

5. Look for the Add-Ons section in your app's dashboard and type **JawsDB** in the input field. That should bring up the **JawsDB MySQL** add-on.

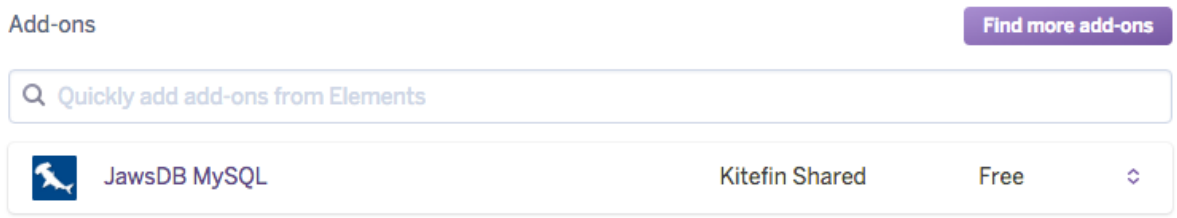


6. Click on **JawsDB MySQL** and that should should bring up a modal asking you to provision a specific tier plan.

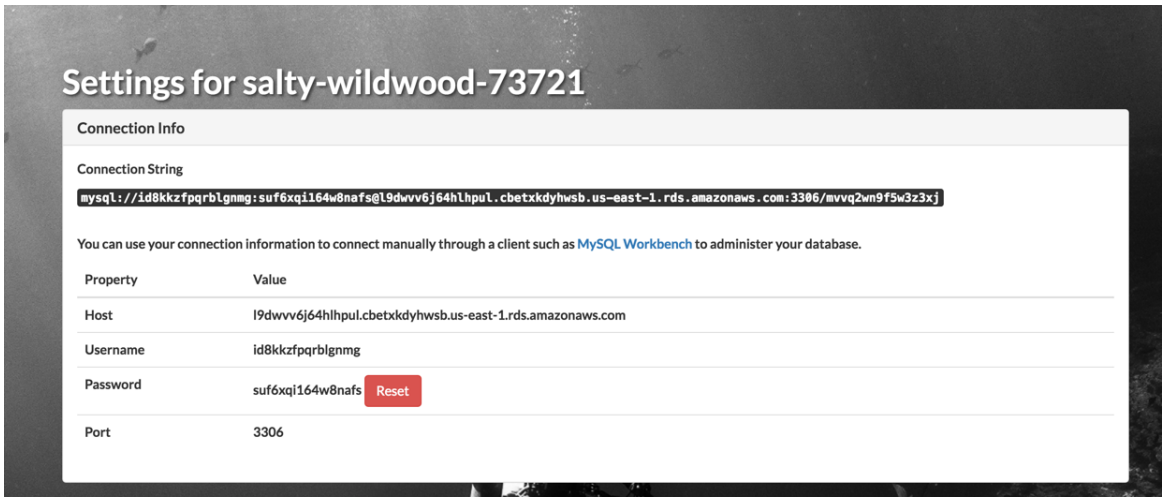


7. Make sure you select the free option, then click the **Provision** button.

8. You'll know that Heroku set up your database when a JawsDB entry shows up in under the Add-ons section.

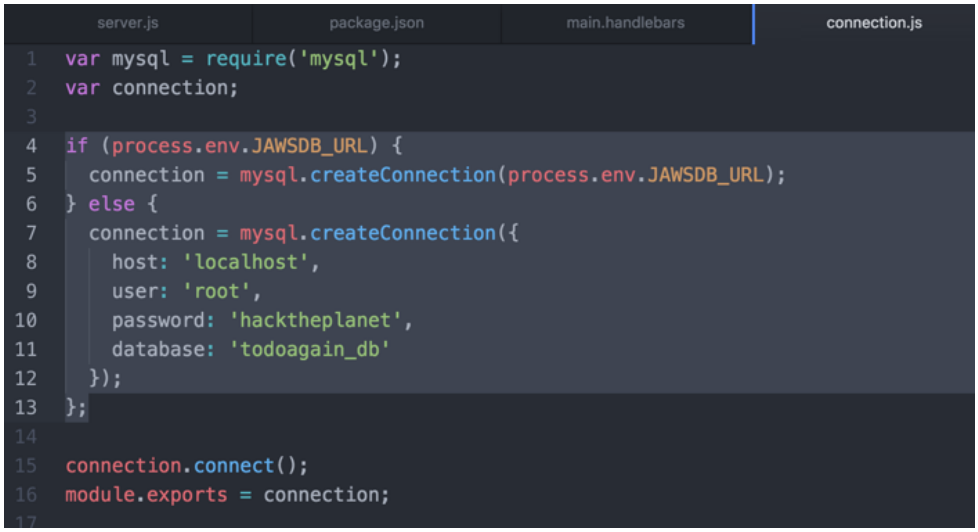


9. Click **JawsDB MySQL** to bring up the settings to your remote database. You'll need this information later.



Part Two: Hooking Your Project with JawsDB

10. In your ***connection.js*** (or whichever file you created your MySQL connection), add the code shown in the image below (additions highlighted):



11. Notice how **process.env.JAWSDB_URL** lets us plug in your connection details with just one object property. When you set up the JawsDB provision, Heroku saved the connection info in an environmental variable. Your Heroku app can reference this variable, hence the if-else statement:

    a. If the server contains the **JAWSDB_URL** environmental variable, it connects to the JawsDB database.

    b. If the server lacks the variable, it falls back on an explicitly defined local database.

    c. You can upload this file to GitHub without worrying about a user finding your remote connection credentials since that info is hidden in the environmental variable.

12. After adding the code above and pushing those changes to your GitHub repository's **master** branch, deploy your app to Heroku: **`git push heroku master`**.

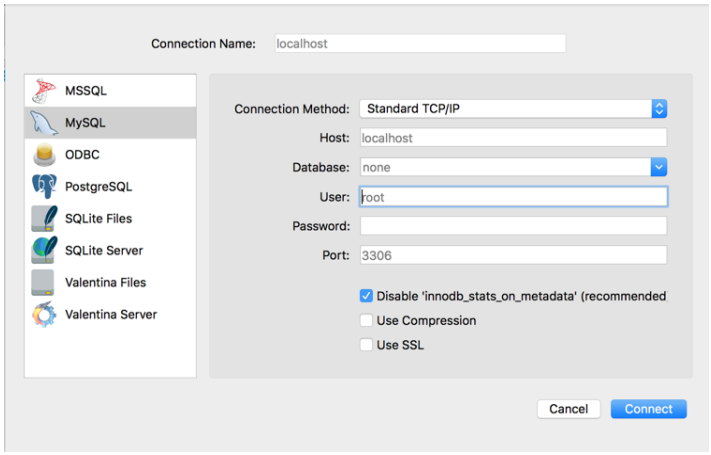
Part Three: Adding Your Project's Tables to JawsDB

13. Now we have to manually create the tables in our JawsDB instance so we can properly connect to it. JawsDB includes certain fields that we must employ in our project. Otherwise, we won't be able to use the remote database.

    a. Open the Graphical User Interface (GUI) software of your choice: **Sequel Pro, MySQL Workbench, Valentina Studio or HeidiSQL**.

        i. To do this, navigate back to the browser and to your JawsDB settings page (as shown in **Step 8**).

    b. Create a connection to the database:



    c. In the **Host** input field, grab the host value from your JawsDB settings page:

Property	Value
Host	l9dwvv6j64hlhpul.cbetxkdyhwsb.us-east-1.rds.amazonaws.com

    d. Use the page's username and password for your connection, as well.

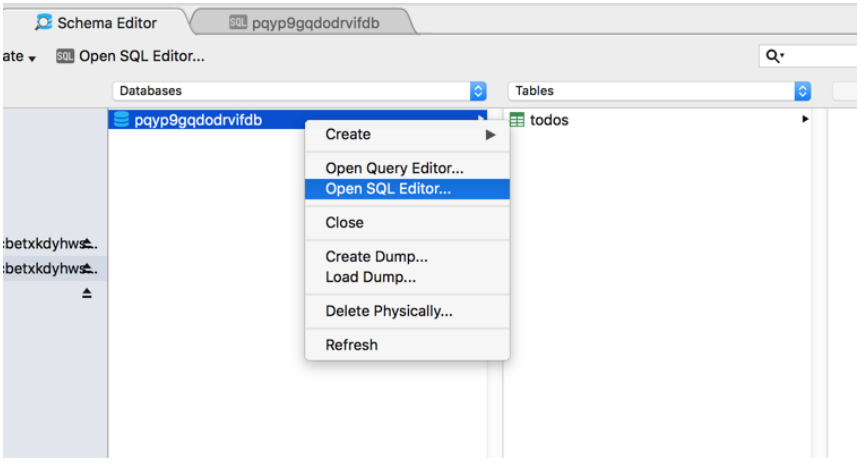
Username	id8kkzfpqrblgnmg
Password	suf6xqi164w8nafs Reset

e. It's easy to accidentally copy an extra space from the settings page. Make sure there are no empty spaces at the end of your **Host, User,** or **Password** values.

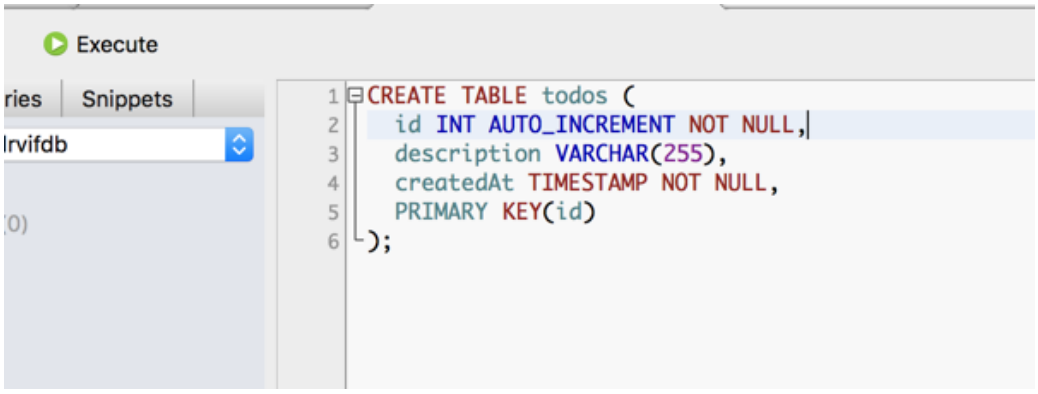
f. After you enter in all the database credentials, establish the connection.

g. Once you're connected to JawsDB, go into its sole database (should be a jumble of characters). You're going to store the tables you need for your project into this database.

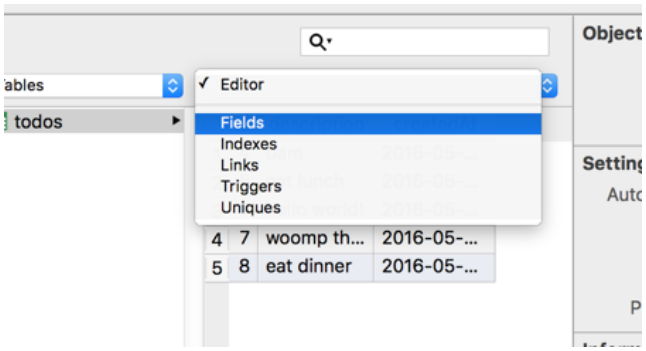
    i. Don't worry about naming this database the same as your local one. Just accept JawsDB's gibberish.



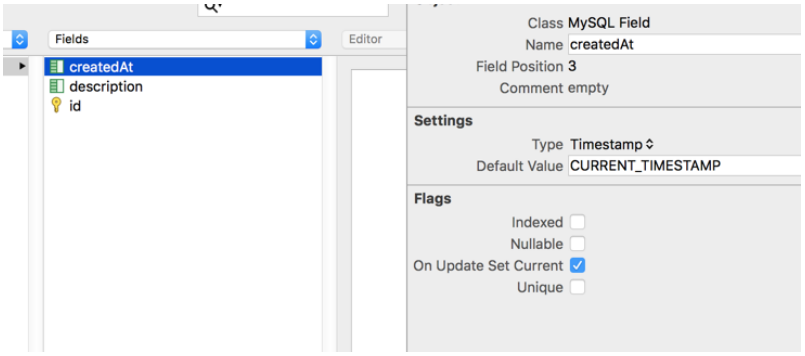
h. Use the schema below as a guide for adding your project's MySQL tables to you're the database. Be sure to include the fields written on **lines 2, 4 and 5** just as they're written.



i. When you create your tables with the prior schema, you also need to add a default value for your `createdAt` field.



i. In your GUI, select the `createdAt` field and change the default value to `CURRENT_TIMESTAMP`. Each GUI has a different process for this, so you may need to do a Google search for those directions (or speak to a TA, who will gladly guide you).



14. After you configure your JawsDB database, run the `heroku open` command in the root of your project directory to open your app in the browser.

Troubleshooting:

If you run into any errors, try going over the steps again to ensure you haven't missed anything. You can avoid errors by checking the following:

A. You set up your **connection.js** folder as we outlined in steps 9 and 10.

B. You copied your JawsDB server details exactly as they appeared in step 8. Be sure that you didn't copy any empty spaces.

C. You included the default fields as outlined in Step 12, Sections g - i.