Heroku Deployment with MySQL

A. **WITH** the Sequelize ORM

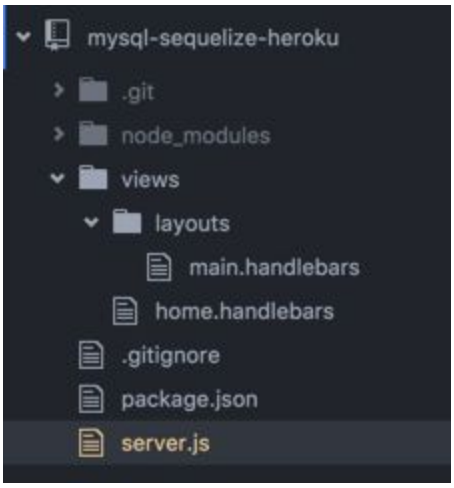
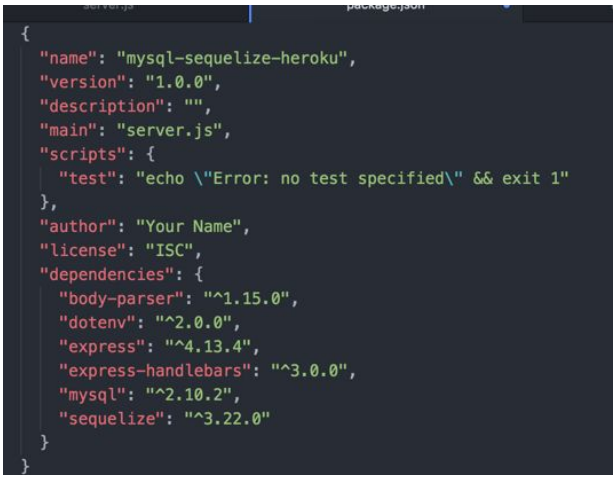


Image Above: File Structure for MySQL Heroku Deployment Instructions with Sequelize



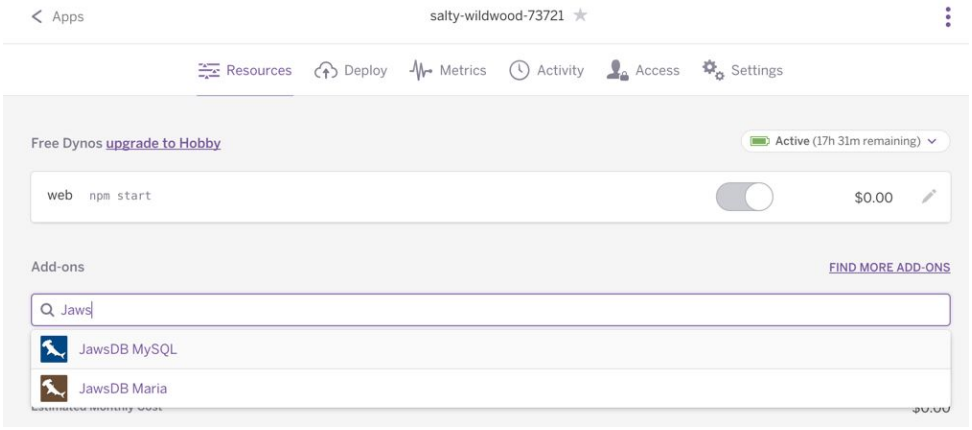
```
{
  "name": "mysql-sequelize-heroku",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Your Name",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.15.0",
    "dotenv": "^2.0.0",
    "express": "^4.13.4",
    "express-handlebars": "^3.0.0",
    "mysql": "^2.10.2",
    "sequelize": "^3.22.0"
  }
}
```

Image Above: *package.json* for MySQL Heroku Deployment Instructions

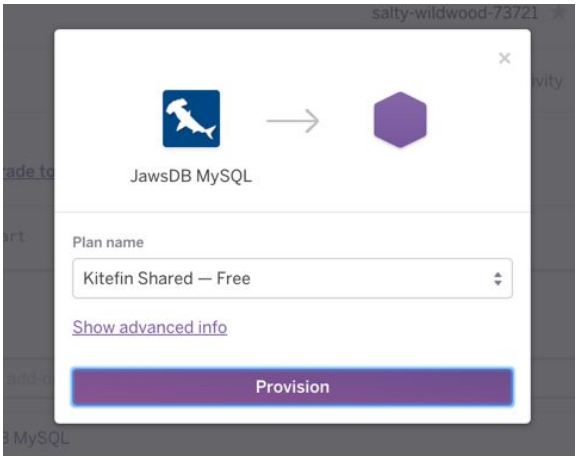
1. On your local machine, navigate to your folder (your local github repository) where you have your code. At this point, you've been pushing/pulling your code to/from Github and not connected to heroku yet.
2. Type in heroku create (enter your Heroku credentials if prompted) and that should create and connect to your heroku app, as shown in the image below.

```
▶ 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
```

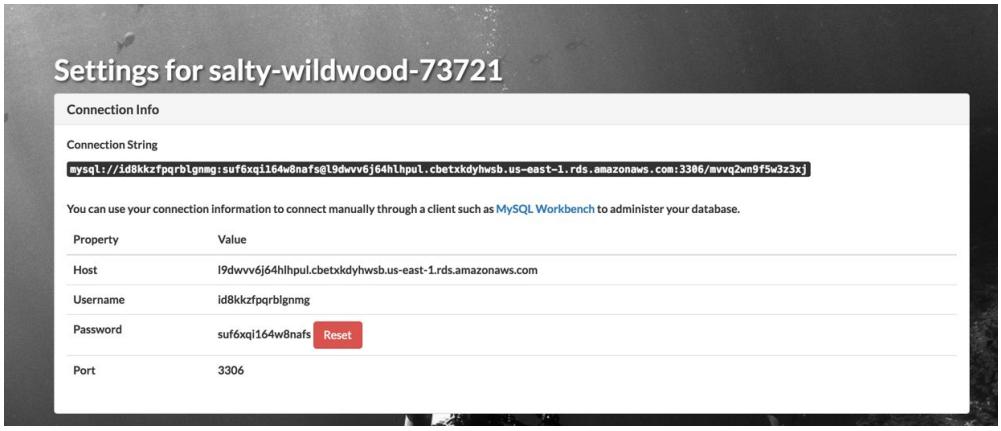
3. Navigate to heroku.com and login with your credentials.
4. Find your heroku app name and click on it.
5. Look for the Add Ons section in your app's dashboard and type in ***JawsDB*** in the input field. That should bring up the *JawsDB MySQL* add on service, as seen in the image below:



6. Click on ***JawsDB MySQL*** and that should should bring up a modal asking you to provision a specific tier plan as seen in the image below:



7. In the modal, the Free tier plan will be selected automatically so just click on the *Provision* button.
8. After clicking the *Provision* button, it will redirect you to your JawsDB settings page like in the image shown below:



9. Back in your code editor, navigate to the ***server.js*** file and create the connection to the MySQL database, as seen in the image below (code highlighted):

```
server.js | home.handlebars | main.handlebars

11
12  // Database setup
13  var Sequelize = require('sequelize'),
14      connection;
15  if (process.env.JAWSDB_URL) {
16    connection = new Sequelize(process.env.JAWSDB_URL);
17  } else {
18    connection = new Sequelize('todo_db', 'root', 'password', {
19      host: 'localhost',
20      dialect: 'mysql',
21      port: '3306'
22    })
23  }
24
25  // Create the model and define the schema using Sequelize
26  var Todo = connection.define('todo', {
27    description: {
28      type: Sequelize.STRING,
29      field: 'description',
30      allowNull: false
31    }
32  });
```

10. Depending on where you place your connection to the MySQL database, the code in the image above takes into account the heroku environment and your local environment, so no matter where your code is running, either on heroku or on your local machine, it will be connecting to mysql.
11. After adding the code above and pushing those changes to your github repository's ***master*** branch, deploy your app to heroku through the command ***git push heroku master***.
12. After the successful deployment to heroku, type in ***heroku open*** on the command line to open your heroku app in the browser (*hopefully Google Chrome :)).*

B. **WITHOUT** the Sequelize ORM (Lightweight ORM)

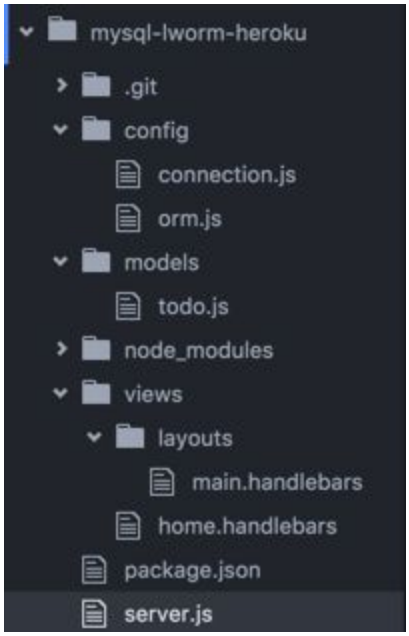


Image Above: File Structure for MySQL Heroku Deployment Instructions using a Lightweight ORM

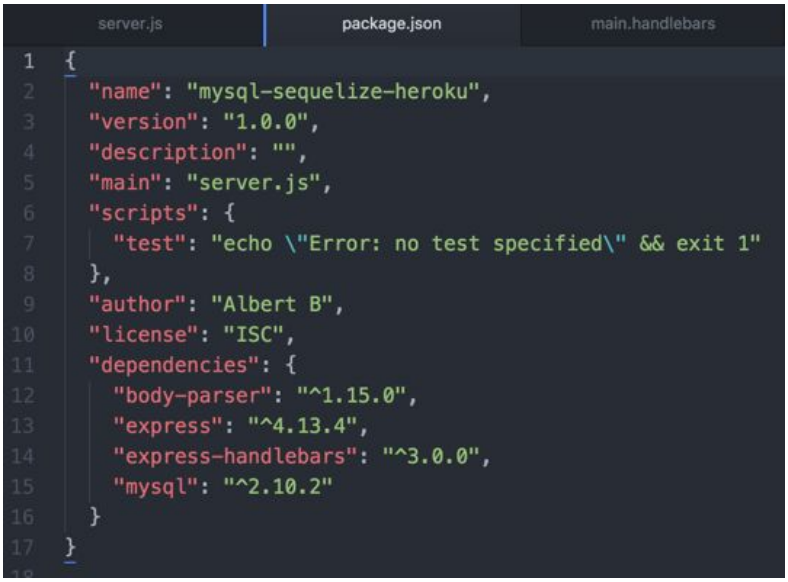


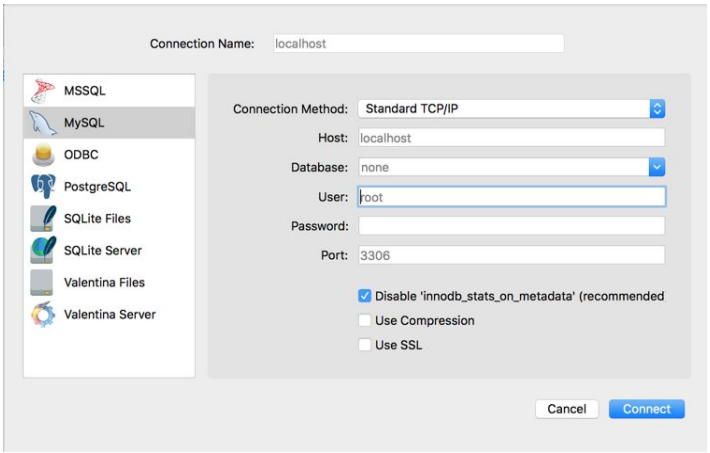
Image Above: *package.json* for MySQL Heroku Deployment Instructions with the Lightweight ORM

1. Repeat steps from **A.1 to A.8**
2. In your ***connection.js*** or where your create the connection to your MySQL database, add the code as seen in the image below (code highlighted):

server.js | package.json | main.handlebars | **connection.js**

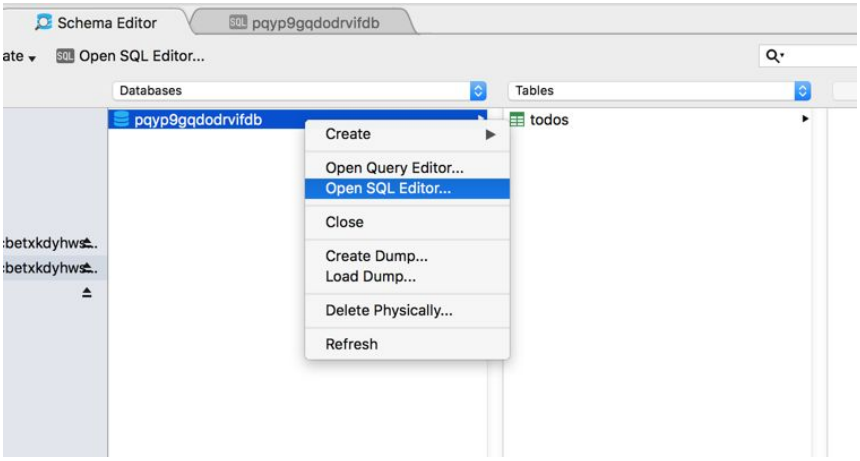
```
var mysql = require('mysql');
var connection;

if (process.env.JAWSDB_URL) {
  connection = mysql.createConnection(process.env.JAWSDB_URL);
} else {
  connection = mysql.createConnection({
    host: 'localhost',
    user: 'root',
    password: 'hacktheplanet',
    database: 'todoagain_db'
  });
};

connection.connect();
module.exports = connection;
```

3. Depending on where you place your connection to the MySQL database, the code in the image above takes into account the heroku environment and your local environment, so no matter where your code is running, either on heroku or on your local machine, it will be connecting to mysql.
4. After adding the code above and pushing those changes to your github repository's ***master*** branch, deploy your app to heroku through the command ***git push heroku master***.
5. Now we have to manually create the tables in our JawsDB instance so we can properly connect to it.
    a. Using Graphical User Interface (GUI) software, either **Valentina Studio** / **HeidiSQL**, connect to your JawsDB database.
        i. To do this, navigate back to the browser and to your JawsDB settings page as outlined in step **A.8**.
    b. As an example using **Valentina Studio**, start by creating the connection to the database:



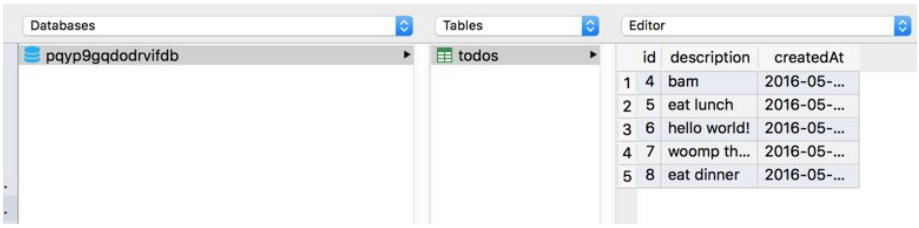
c. Select the **MySQL** connection in the left menu.
d. In the **Host** input field, grab the host value from your JawsDB settings page:

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

e. For the **User** input field, grab the Username value from your JawsDB settings page and likewise with the **Password** input field from your JawsDB settings.
f. Click the **Connect** button or the equivalent in **HeidiSQL**.
g. Make sure there are no spaces in your **Host, User,** or **Password** values.
h. Once you're connected, right-click on your JawsDB database and select the ***Open SQL Editor*** option.
i. 

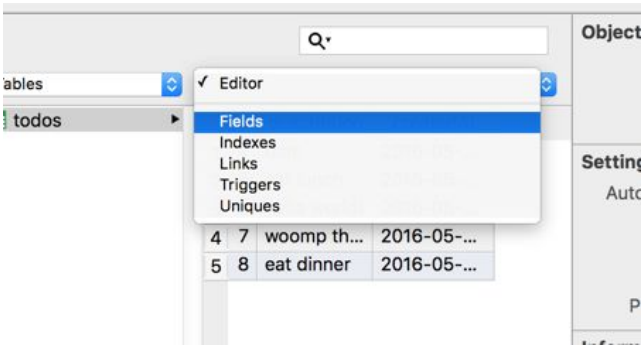
j. That should bring up the SQL editor for your specific JawsDB database as shown in the image below:
k. 

l. Using the image above as a guideline, create your MySQL table. Lines 2, 4, and 5 are what you should have in your schema.
m. After writing your SQL syntax to define your schema, click on the **Execute** button as shown in image above and in your **SQL Editor**. This will create the table for us as shown in the image below.

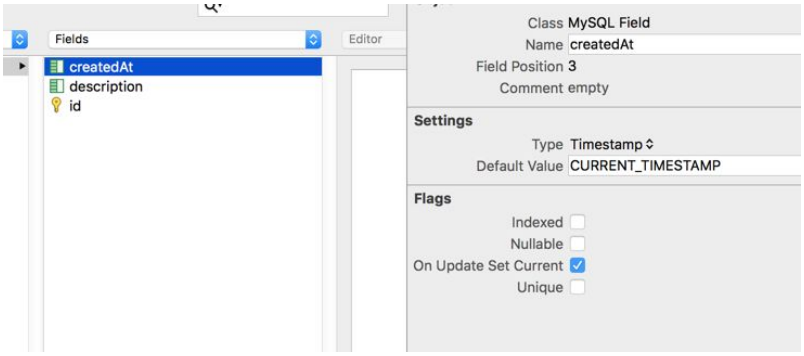
n. 

o. Let's modify the Default Value for our **Date** field.

i. Click on the drop down menu where you see **Editor** and select the **Fields** option:

ii. 

iii. After selecting the **Fields** option, add the **Default Value** of ***CURRENT_TIMESTAMP*** and check the **On Update Set Current** flag in the **Flags** section.

iv. 

v. That should properly configure the ***createdAt*** field for our table.

6. Now type in ***heroku open*** on the command line in your local git repository (your folder where github and heroku are linked to) to open your heroku app in the browser (*hopefully Google Chrome :))* and your app should load up (may take a few seconds since we have to wake our app up on heroku).