

Before we get started, we'll need to install the Mongoose module:

```
npm install --save mongoose
```

Then we'll add our database credentials to the *credentials.js* file:

```
mongo: {
  development: {
    connectionString: 'your_dev_connection_string',
  },
  production: {
    connectionString: 'your_production_connection_string',
  },
},
```

You'll find your connection string on the database page in MongoLab: from your home screen, click the appropriate database. You'll see a box with your MongoDB connection URI (it starts with *mongodb://*). You'll also need a user for your database. To create one, click Users, then "Add database user."

Notice that we store two sets of credentials: one for development and one for production. You can go ahead and set up two databases now, or just point both to the same database (when it's time to go live, you can switch to using two separate databases).

## Database Connections with Mongoose

We'll start by creating a connection to our database:

```
var mongoose = require('mongoose');
var opts = {
  server: {
    socketOptions: { keepAlive: 1 }
  }
};
switch(app.get('env')){
  case 'development':
    mongoose.connect(credentials.mongo.development.connectionString, opts);
    break;
  case 'production':
    mongoose.connect(credentials.mongo.production.connectionString, opts);
    break;
  default:
    throw new Error('Unknown execution environment: ' + app.get('env'));
}
```

The options object is optional, but we want to specify the *keepAlive* option, which will prevent database connection errors for long-running applications (like a website).