MODEL

Before we start writing routes which manipulate resources, we need to create a model of our resource so that we may ensure some sort of structure for the documents in our database collection.

Step 1: Create a City.js file in our models folder in the backend of our project.

Step 2: At the top of our file import mongoose and set it to a constant.

```
const mongoose = require('mongoose')
```

Step 3: We then need create a new instance of a mongoose.Schema and set it to a constant.

```
const CitySchema = new mongoose.Schema({
```

Step 4: Now we define the properties that our CitySchema will possess. For our purposes we only need three properties.

- name
- country
- img

Step 5: Set the values of each property to an object.

```
 name: {
    type: String,
    required: true
  },
// By setting the value of each property to an object we have a wide range of options when it comes to customizing each property.
// The first property of the object is the “type” which signifies which type of data our server will be expecting to receive when we send our request for the respective property of our schema.
// For instance image our administrator tries to enter a number into the name field for a city. An error will be thrown and the resource will not be saved to our database.
```

```
// The next property of the object is the “required” field which is a Boolean we set to true. This way our admin will also receive an error if he leaves an input on our form blank.
```

Step 6: Repeat Step 5 for the country and img fields.

Step 7: Export model as ‘City’.

```
module.exports = City = mongoose.model('city', CitySchema)
// We set the export to a variable City and make it equal a call to mongoose.model().
// The first argument is a string of our Schema, this will automatically convert to plural once we post the first city to our database.
// The second argument is the Schema that we just defined.
```