

Unit 15.2 Highlights

MVC - (Model-View-Controller) Design pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application.

ORM - Object Relational Mapping - Programming technique for converting data between incompatible type systems using object-oriented programming languages. This creates, in effect, a "virtual object database" that can be used from within the programming language.

Sequelize (sequelizejs.com) - Promise-based Node v4+ ORM which supports dialects of PostgreSQL, MySQL, SQLite & MSSQL

(Source: <http://docs.sequelizejs.com/manual/tutorial/models-usage.html>)

CRUD Sequelize queries and methods

Model.create()
Model.findAll() or Model.findOne()
Model.destroy()
Model.update()

Ex. 1

```
//===== app/routes/api-routes.js (entry point)=====//  
  
var Car = require("../models/car.js");  
  
module.exports = function(app) {  
  
  // Get all Cars  
  app.get("/api/cars/all", function(req, res) {  
  
    // Finding all Cars, and then returning them to the user as JSON.  
    // Sequelize queries are asynchronous, which helps with perceived speed.  
    // If we want something to be guaranteed to happen after the query, we'll use  
    // the .then function  
    Car.findAll({}).then(function(results) {  
      // results are available to us inside the .then  
      res.json(results);  
    });  
  
  });  
  
  // Add a Car  
  app.post("/api/cars/new", function(req, res) {  
  
    Car.create({  
      make: req.body.make,  
      model: req.body.model,  
      year: req.body.year,  
      color: req.body.color  
    }).then(function(results) {  
      // `results` here would be the newly created car  
      res.end();  
    });  
  
  });  
  
  // Delete a Car  
  app.delete("/api/cars/:carid", function(req, res) {  
  
    Car.destroy({  
      where: {  
        id: req.params.carid  
      }  
    }).then(function(deletedCar) {  
      res.json(deletedCar);  
    });  
  
  });  
  
});
```

```
// Update a Car
app.put("/api/cars/:carid", function(req, res) {

  Car.update({
    make: req.body.make,
    model: req.body.model,
    year: req.body.year,
    color: req.body.color
  }, {
    where: {
      id: req.params.carid
    }
  }).then(function(deletedCar) {
    res.json(deletedCar);
  });
});

});

};
```