Unit 14.2 Highlights

bodyParser.urlencoded({extended: true}) - extended option allows to choice between the libraries which parse the URL-encoded data (querystring when false; qs when true)
response.redirect() - Redirects to a URL
response.status() - Sets the HTTP status for the response (Common status codes: 200, 500, 404)
response.end() - Ends the response process

Ex. 1

```
//Load package dependencies
var express = require("express");
var exphbs = require("express-handlebars");
var bodyParser = require("body-parser");
var mysql = require("mysql");

//============MySQL config, connection statements==============//
var connection = mysql.createConnection({
  host: "localhost",
  port: 8889,
  user: "root",
  password: "root",
  database: "task_saver_db"
});

connection.connect(function(err) {
  if (err) {
    console.error("error connecting: " + err.stack);
    return;
  }
  console.log("connected as id " + connection.threadId);
});

//============Express Server Set Up ==============//

var app = express(); //create express instance

//set the port; ensures the proper port is used whether code is run locally or deployed to cloud
var PORT = process.env.PORT || 3000;

// Sets up the Express app middleware to handle data parsing
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());
app.engine("handlebars", exphbs({ defaultLayout: "main" }));
app.set("view engine", "handlebars");

//============Express Server Routes ==============//

app.get("/", function(req, res) {
   connection.query("SELECT * FROM teams;", function(err, data) {
    if (err) throw err;
    res.render("index", { teams: data });
});

app.post("/", function(req, res) {
   var teamPassedInAsJSONInPostRequest = req.body.team;
   connection.query("INSERT INTO teams (team) VALUES (?)",[teamPassedInAsJSONInPostRequest], function(err, data){
   if (err) throw err;
   res.redirect("/"); //send the user to the route which will display the entire list with the new team
});

app.put("/team/:id", function(req, res) {
  connection.query("UPDATE teams SET team = ? WHERE id = ?", [req.body.team, req.params.id], function(err,
result) {
   if (err) return res.status(500).end(); // If an error occurred, send a generic server failure
   else if (result.changedRows === 0)
       return res.status(404).end(); // If no rows were changed, then the ID must not exist, so 404
   res.status(200).end();
  });
});
app.delete("/team/:id", function(req, res) {
  connection.query("DELETE FROM teams WHERE id = ?", [req.params.id], function(err, result) {
   if (err) return res.status(500).end(); // If an error occurred, send a generic server failure
   else if (result.affectedRows === 0)
       return res.status(404).end(); // If no rows were changed, then the ID must not exist, so 404
   res.status(200).end();
  });
});

//============Express Server Start ==============//

app.listen(PORT, function() {
  console.log("App listening on PORT " + PORT);
});
```