Exercise (Instructions): Introduction to Express Part 2

Objectives and Outcomes

In this exercise, you will develop a web server that exports a REST API. You will use the Express framework, and the Express router to implement the server. At the end of this exercise, you will be able to:

- Use application routes in the Express framework to support REST API
- Use the Express Router in Express framework to support REST API

Setting up a REST API

- Create a new file named *server-3.js* and add the following code to it:

```
var express = require('express');
var morgan = require('morgan');
var bodyParser = require('body-parser');

var hostname = 'localhost';
var port = 3000;

var app = express();

app.use(morgan('dev'));
app.use(bodyParser.json());

app.all('/dishes', function(req,res,next) {
      res.writeHead(200, { 'Content-Type': 'text/plain' });
      next();
});

app.get('/dishes', function(req,res,next){
        res.end('Will send all the dishes to you!');
});

app.post('/dishes', function(req, res, next){
     res.end('Will add the dish: ' + req.body.name + ' with details: ' + req
       .body.description);
});

app.delete('/dishes', function(req, res, next){
        res.end('Deleting all dishes');
});

app.get('/dishes/:dishId', function(req,res,next){
        res.end('Will send details of the dish: ' + req.params.dishId +' to
          you!');
});

app.put('/dishes/:dishId', function(req, res, next){
    res.write('Updating the dish: ' + req.params.dishId + '\n');
    res.end('Will update the dish: ' + req.body.name +
            ' with details: ' + req.body.description);
});

app.delete('/dishes/:dishId', function(req, res, next){
        res.end('Deleting dish: ' + req.params.dishId);
});

app.use(express.static(__dirname + '/public'));

app.listen(port, hostname, function(){
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

- Install body-parser by typing the following at the command prompt:

```
npm install body-parser --save
```

- Start the server and interact with it from the browser/postman.

Using Express Router

- Create a new file named *server-4.js* and add the following code to it:

```
var express = require('express');
var morgan = require('morgan');
var bodyParser = require('body-parser');

var hostname = 'localhost';
var port = 3000;

var app = express();

app.use(morgan('dev'));

var dishRouter = express.Router();

dishRouter.use(bodyParser.json());

dishRouter.route('/')
.all(function(req,res,next) {
      res.writeHead(200, { 'Content-Type': 'text/plain' });
      next();
})

.get(function(req,res,next){
        res.end('Will send all the dishes to you!');
})

.post(function(req, res, next){
    res.end('Will add the dish: ' + req.body.name + ' with details: ' + req.body
      .description);
})

.delete(function(req, res, next){
        res.end('Deleting all dishes');
});

dishRouter.route('/:dishId')
.all(function(req,res,next) {
      res.writeHead(200, { 'Content-Type': 'text/plain' });
      next();
})

.get(function(req,res,next){
        res.end('Will send details of the dish: ' + req.params.dishId +' to
          you!');
})

.put(function(req, res, next){
        res.write('Updating the dish: ' + req.params.dishId + '\n');
    res.end('Will update the dish: ' + req.body.name +
             ' with details: ' + req.body.description);
})

.delete(function(req, res, next){
        res.end('Deleting dish: ' + req.params.dishId);
});

app.use('/dishes',dishRouter);

app.use(express.static(__dirname + '/public'));

app.listen(port, hostname, function(){
  console.log(`Server running at http://${hostname}:${port}/`);
});

```

- Start the server and interact with it and see the result.

Conclusions

In this exercise, you used the Express framework and Express router to build a server supporting a REST API.

✓ Complete