### **Express Router**

#### 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

* You will continue in the node-express folder and modify the server in this exercise.
* Install body-parser by typing the following at the command prompt:

     npm install body-parser

* Update index.js as shown below:
* . . .
* const bodyParser = require('body-parser');
* . . .
* app.use(bodyParser.json());
* app.all('/dishes', (req,res,next) => {
* res.statusCode = 200;
* res.setHeader('Content-Type', 'text/plain');
* next();
* });
* app.get('/dishes', (req,res,next) => {
* res.end('Will send all the dishes to you!');
* });
* app.post('/dishes', (req, res, next) => {
* res.end('Will add the dish: ' + req.body.name + ' with details: ' + req.body.description);
* });
* app.put('/dishes', (req, res, next) => {
* res.statusCode = 403;
* res.end('PUT operation not supported on /dishes');
* });
* app.delete('/dishes', (req, res, next) => {
* res.end('Deleting all dishes');
* });
* app.get('/dishes/:dishId', (req,res,next) => {
* res.end('Will send details of the dish: ' + req.params.dishId +' to you!');
* });
* app.post('/dishes/:dishId', (req, res, next) => {
* res.statusCode = 403;
* res.end('POST operation not supported on /dishes/'+ req.params.dishId);
* });
* app.put('/dishes/:dishId', (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', (req, res, next) => {
* res.end('Deleting dish: ' + req.params.dishId);
* });
* . . .
* Start the server and interact with it from the browser/postman.
* Do a Git commit with the message "Express Simple REST".

#### Using Express Router

* Create a new folder named routes in the node-express folder.
* Create a new file named dishRouter.js in the routes folder and add the following code to it:
* const express = require('express');
* const bodyParser = require('body-parser');
* const dishRouter = express.Router();
* dishRouter.use(bodyParser.json());
* dishRouter.route('/')
* .all((req,res,next) => {
* res.statusCode = 200;
* res.setHeader('Content-Type', 'text/plain');
* next();
* })
* .get((req,res,next) => {
* res.end('Will send all the dishes to you!');
* })
* .post((req, res, next) => {
* res.end('Will add the dish: ' + req.body.name + ' with details: ' + req.body.description);
* })
* .put((req, res, next) => {
* res.statusCode = 403;
* res.end('PUT operation not supported on /dishes');
* })
* .delete((req, res, next) => {
* res.end('Deleting all dishes');
* });
* module.exports = dishRouter;
* Update index.js as follows:
* . . .
* const dishRouter = require('./routes/dishRouter');
* app.use('/dishes', dishRouter);
* . . .
* Start the server and interact with it and see the result.
* Do a Git commit with the message "Express Router".

#### Conclusions

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