

Node.js Demo

1 Objectives

In this lab, you will learn how to set up a Node.js web service that accepts http calls implementing a very basic RESTful interface, queries your local database server, and returns JSON objects.

2 Assumptions

This lab assumes you already have Node.js installed along with Fiddler 4.

Downloads for Node.js can be found [here](#).

Fiddler Classic downloads can be found [here](#).

3 Exercise 1: Using a RESTful Web Service

In this exercise, Node.js code will be used to set up a web service that accepts “GET” requests from clients that are wanting to “consume” information currently contained in a local database.

1. Navigate to the folder on the C: drive containing the provided Node.js code. Open the file index.js using a text editor (i.e., Notepad++).
2. Look through the code and discuss with your group what you think each part does. Notice the names of the properties of the variable named “currentWeather”. These property names will need to be used to POST to this web service.

This code defines our web service API. It currently supports only one type of call; a “GET”. A “GET” returns the current value of the requested data.

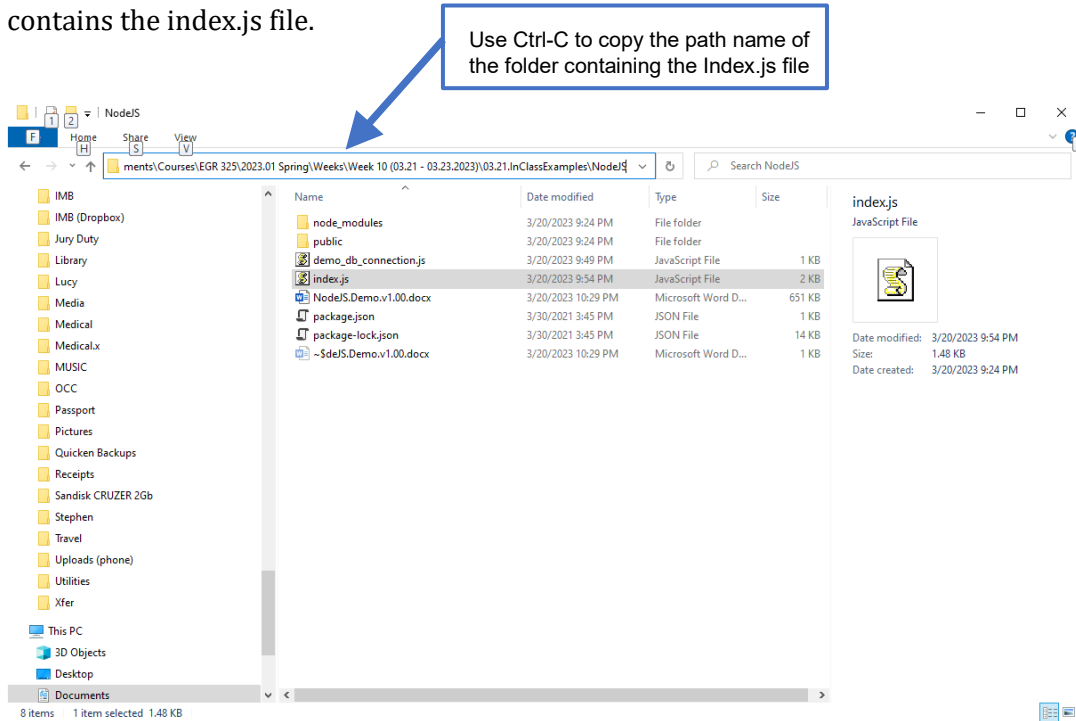
```
...
app.get('/invoice', function(req, res) {
    console.log("Invoice request: ALL invoices" + new Date());

    con.connect(function(err) {
        if (err) throw err;
        console.log("Connected!");

        var sql = "SELECT * FROM invoices;";

        con.query(sql, function (err, result, fields) {
            if (err) throw err;
            res.json(result);
        });
    });
});
```

3. Close the file without making any changes. Use windows explorer to copy the name of the folder that contains the index.js file.



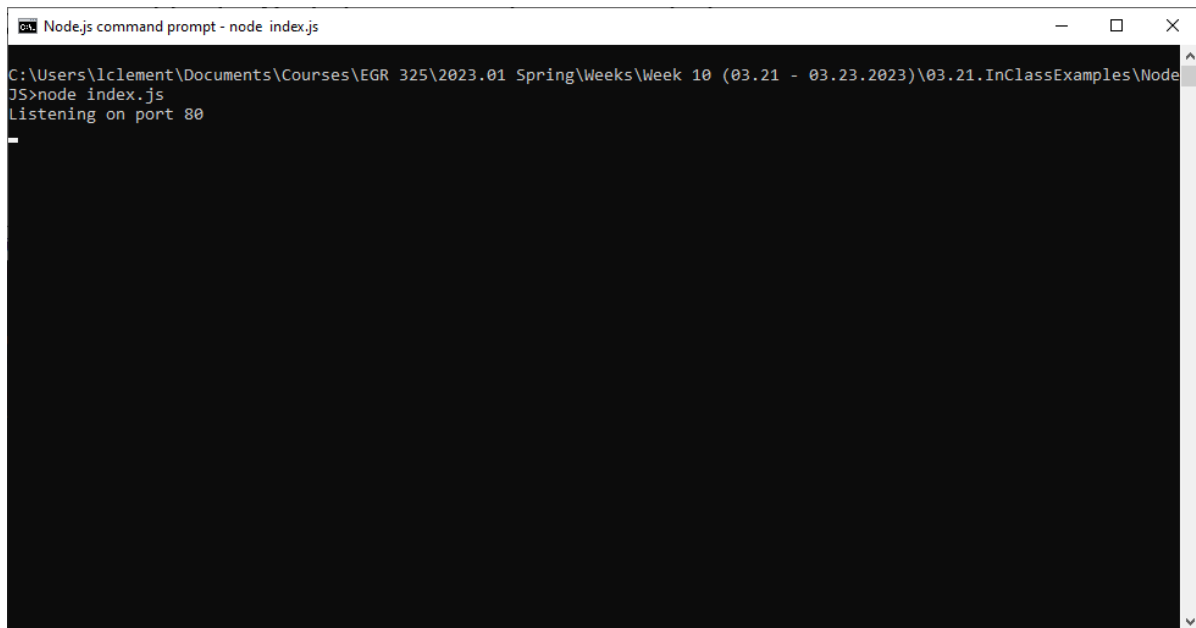
4. Click “Start->All Programs->Node.js->Node.js command prompt” to open a Node.js command prompt window. Type “CD” followed by a space. Then “right click” and PASTE the full path of the folder containing the index.js file to create a single command that changes the “directory” (another name for folder) to the one containing the index.js file.

```
Node.js command prompt
Your environment has been set up for using Node.js 14.16.0 (x64) and npm.

C:\Users\lcllement>CD C:\Users\lcllement\Documents\Courses\EGR 325\2023.01 Spring\Weeks\Week 10 (03.21 - 03.23.2023)\03.21.InClassExamples\NodeJS
```

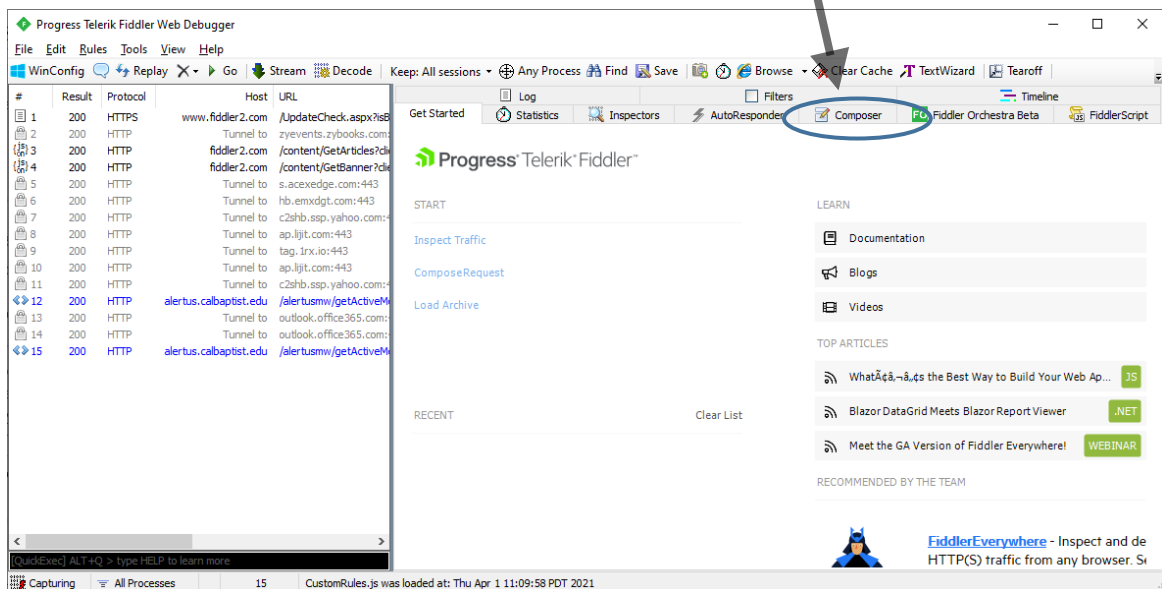
5. Enter the following command in the Node.js command prompt window:
node index.js

You should see a message appear indicating that the service is now “Listening on port 80”.

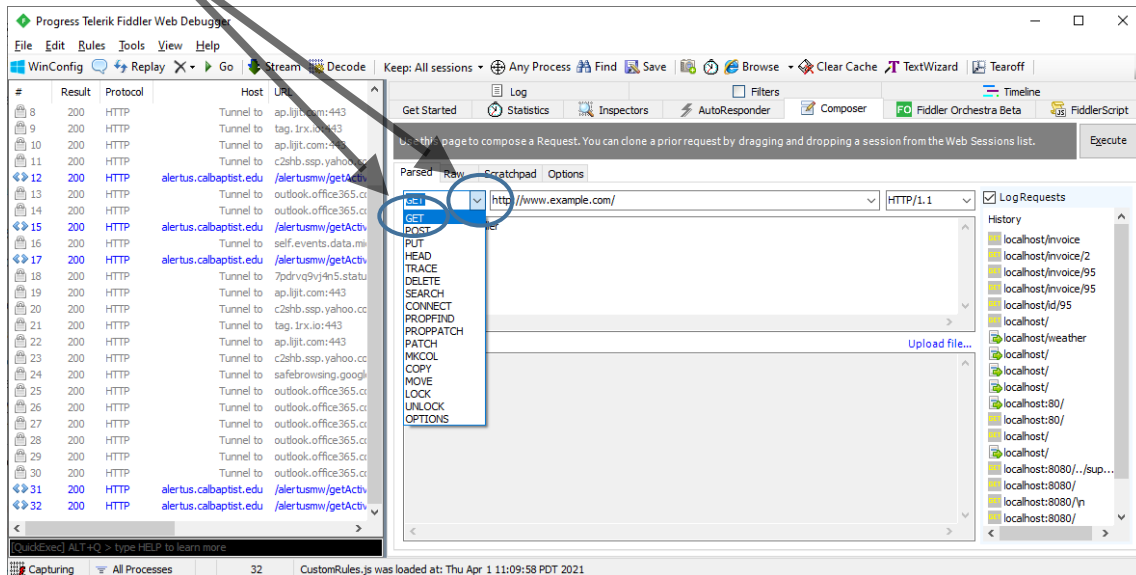


```
Node.js command prompt - node index.js
C:\Users\lclement\Documents\Courses\EGR 325\2023.01 Spring\Weeks\Week 10 (03.21 - 03.23.2023)\03.21.InClassExamples\Node
JS>node index.js
Listening on port 80
```

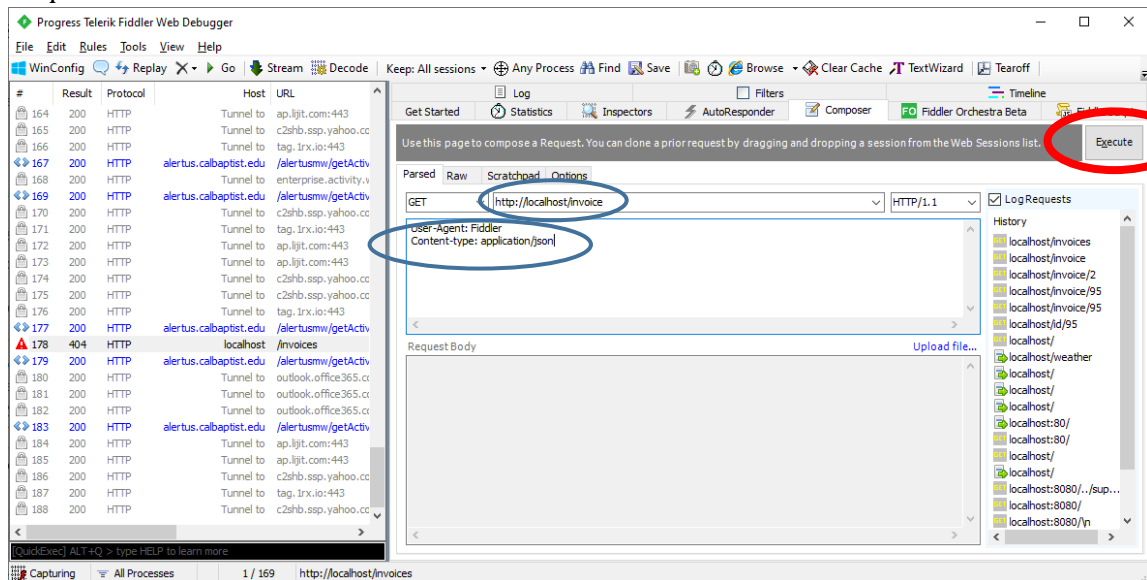
6. Click “Start->All Programs->Fiddler 4” to launch a program named “Fiddler” that will allow us to create and send a GET to the web service. Click on the tab named “Composer”.



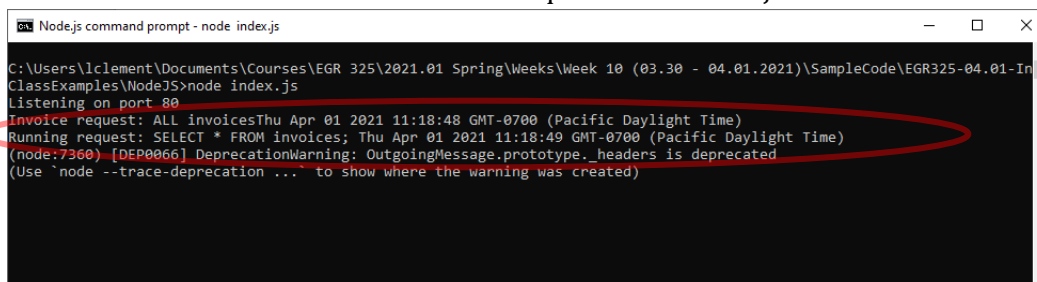
7. Click the down arrow to open a drop down menu and select “POST”.



8. Change the URL to be `http://localhost/invoice`, add a new header “Content-type: application/json” to the text box containing the headers. When everything is entered click the Execute button to submit the GET request.

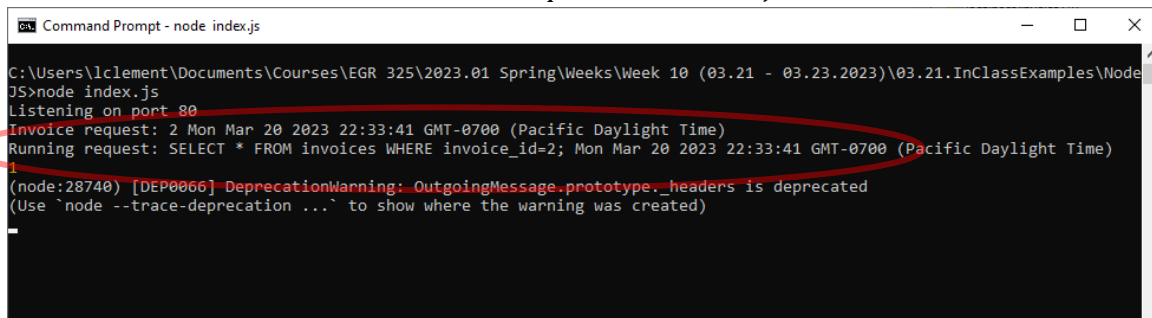


You should then see a record of the GET request in the Node.js command window.



- Return to the Fiddler window and update the URL for the HTTP GET request to be `http://localhost/invoice/2`. Click the Execute button to submit the GET request.

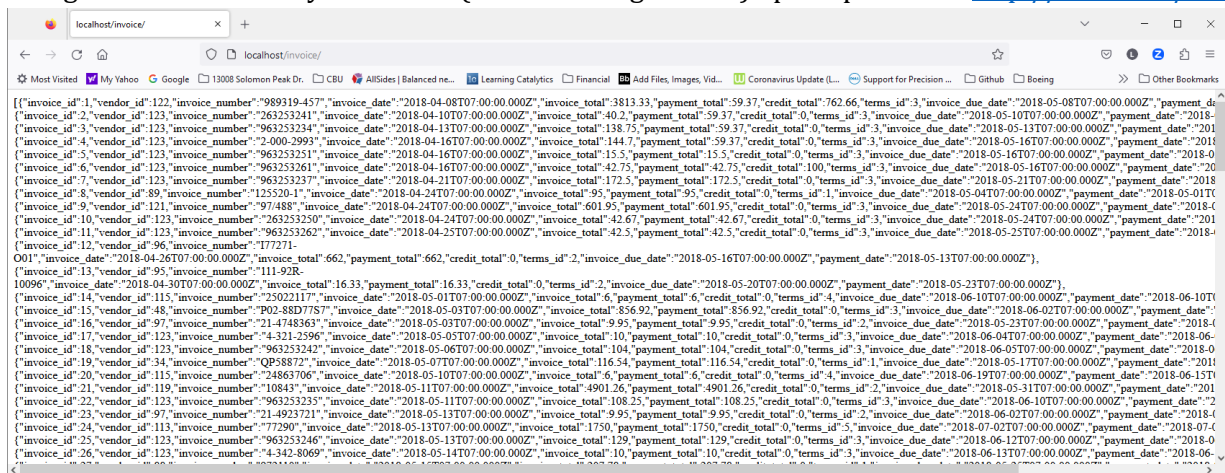
You should then see a record of the GET request in the Node.js command window.



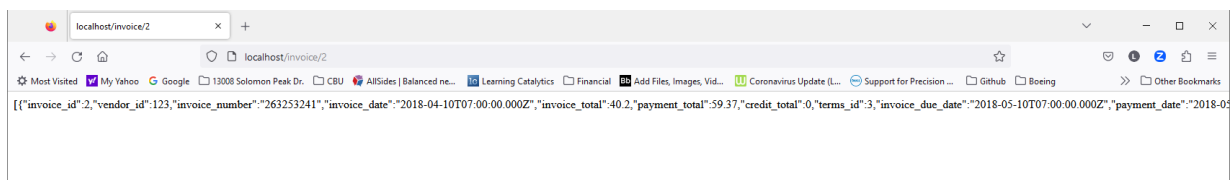
```
Command Prompt - node index.js
C:\Users\lclement\Documents\Courses\EGR 325\2023.01 Spring\Weeks\Week 10 (03.21 - 03.23.2023)\03.21.InClassExamples\Node
JS>node index.js
Listening on port 80
Invoice request: 2 Mon Mar 20 2023 22:33:41 GMT-0700 (Pacific Daylight Time)
Running request: SELECT * FROM invoices WHERE invoice_id=2; Mon Mar 20 2023 22:33:41 GMT-0700 (Pacific Daylight Time)
(node:28740) [DEP0066] DeprecationWarning: OutgoingMessage.prototype.headers is deprecated
(Use `node --trace-deprecation ...` to show where the warning was created)
```

- We can also access the web service using a web browser.

Using a web browser of your choice (I will be using Firefox) open up the URL <http://localhost/invoice>.



If we change the URL to read <http://localhost/invoice/2> we will just get the data for the single invoice having id = 2.



- Node.js can be used to set up a complete API that would provide data to any device capable of making HTTP requests using the RESTful architecture.
- Here is a URL that describes how to get data from Twitter using the publicly available REST API. <https://developer.twitter.com/en/docs/tutorials/step-by-step-guide-to-making-your-first-request-to-the-twitter-api-v2>