

GET: collecting forms data

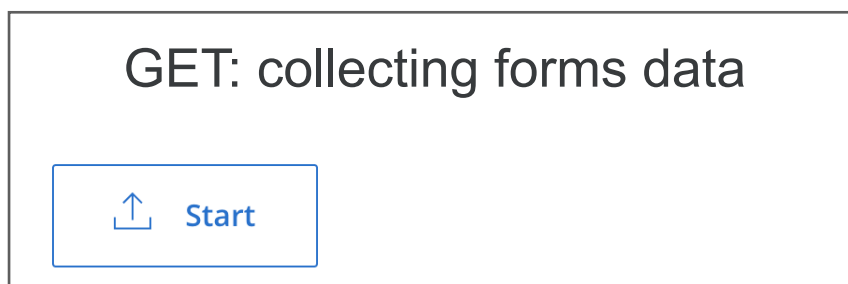
Welcome to this Lab activity

In this lab activity you will be exploring how to collect data from your existing form using the GET request method. For the purpose of this lab activity you will be working on the same structure as your previous lab: *topic4/htmlExpress*.

Start the Lab environment application

It is simple to launch a lab exercise. You only need to click on the button “Start” below the activity title to enter a lab environment.

Let’s explore this lab activity. Go ahead and click on the “Start” button!



Task 1: Collecting form data, when http request method is GET

The folder structure has already been partially constructed for you and organised into different topics. For the purpose of this lab, you will be making changes inside the *topic4* folder structure. You will be working in your previous lab application folder *topic4/htmlExpress*. Let’s get started!

Please do not delete or move any existing folders/files inside the lab environment.

First of all you need to edit your form inside the *topic4/htmlExpress/views/search.html* file. Rename the name attribute “search-keyword” to “keyword” and update the “action” value to “/search-result” as you can see below:

```
<form action="/search-result" method="GET">  
  <input id="search-box" type="text" name="keyword" value="Default">  
  <input type="submit" value="OK" >  
</form>
```

Now edit your “main.js” file inside the *topic4/htmlExpress/routes/* folder and create a new route to collect and display data from your form.

Add the following code to your “main.js” file:

```
app.get("/search-result", function (req, res) {  
  //searching in the database  
  res.send(req.query);  
});
```

As you can see we have added a comment to search in the database. You will explore how to integrate databases with your application later in the course. For the moment you are simply collecting the data passed along with the form. This information is collected inside **req.query**.

Running the index.js file

Run the *index.js* file with the following Terminal command:

- **node index.js**: type this command and press Enter. The node command, followed by the file name, tells Node.js to execute the content of the file.

The above command will start a web server running on port 8086.

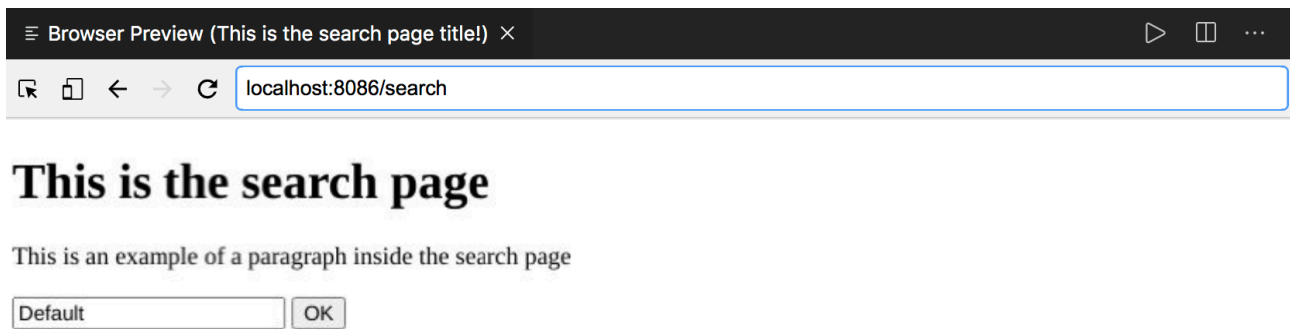
Task 2: Access your server via HTTP

Now that your code is running, serving your application on port 8086, you can access your web page from a browser!

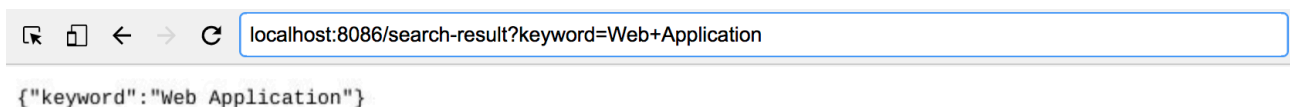
Use the “Browser Preview” plugin to visualise your web application.

If you do not remember how to use the “Browser Preview” plugin, please refer to the “Creating my first Node.js web server” lab instructions.

Type `localhost:8086/search` on the “Browser Preview” tab and press *Enter* on your keyboard to visualise your web application “search” page:



Now enter a keyword in the text field and press the “OK” button to submit your form data. If you have successfully followed all the above steps you should get the following result:



Note: your keyword value will contain the text field value submitted in your form.

Finally replace the “`res.send`” line in your “`main.js`” file with the following:

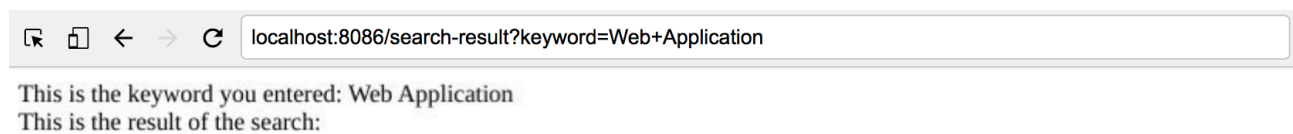
```
app.get("/search-result", function (req, res) {  
  //searching in the database  
  res.send(req.query.keyword);  
});
```

Visit the “search” route one more time and submit the form with another value. You can see that data items can be accessed individually in the query.

What about replacing the “res.send” with the following line:

```
app.get("/search-result", function (req, res) {  
  //searching in the database  
  res.send("This is the keyword you entered: " + req.query.keyword + "<br>" + "This is the  
    result of the search:");  
});
```

If you run your application now, you should see the following result:



End of Section

Congratulations for completing this section. As long as you have saved your work, your files will remain when you close this lab activity so do not worry about losing your data. You have successfully created a web application that serves html files containing forms. Furthermore, you are now able to collect data using the GET request method. In the next lab activity you will explore another important method for data manipulation: the POST method.