# Graduate Credit Assignment

Lester D’Souza

## Introduction

In week6 of the course we learnt about view templates in a Node+Express server setting. In that module “Pug” template engine was used. Pug is a good template engine, but it is too sensitive to spaces. Also the syntax is quite different from the regular html. It will be better to use a template engine that is closer to regular html. I thought of trying handlebars templates instead.  It turns out it is quite easy to set it up if you use express generator.

$ express --view=hbs handlebars-project

The above command took care of most of the setup.  The localhost:8080  works out of the box and shows output from the index.hbs (handlebar template)

The only difference between the pug setup and handlebars setup is that the views folder contains the .hbs files instead of .pug files.

Handlebar templates are closer to real html and maybe easier to deal with in larger projects. This tutorial shows how to setup your Node+Express server to serve handlebar templates.

## Prerequisites

This tutorial will best work at the end of the week6 module.

Week 6: A Deeper Look at Template Engines and Pug

<https://canvas.harvard.edu/courses/54354/pages/week-6-a-deeper-look-at-template-engines-and-pug?module_item_id=453532>

# Week 6: Express Generator

<https://canvas.harvard.edu/courses/54354/pages/week-6-express-generator?module_item_id=453533>

At the end of week6 module if the students are frustrated using Pug templates and are looking to migrate to the handlebar templates then this tutorial is for them.

## Handlebar Templates

This tutorial is limited to introducing handlebar setup with node and express. Only basic handlebar templates are explained. Detail handlebar tutorial is beyond the scope of this tutorial. Here is the link to the documentation for handlebar templates.

<https://handlebarsjs.com/>

The documentation is quite intuitive and easy to follow and you will be able to create your own templates once you reach the end of the tutorial.

## Node Express Generator

Please review how to prepare your node environment to generate a node express project from Week 6: Express Generator videos and lessons.

## Create and Initialize a Node Express Project

Type the following commands to create a node express project using the express generator.

$ express --view=hbs assignment-graduate-zzr994

This creates a new project folder

$ cd assignment-graduate-zzr994

Under the folder “views” there are three files with .hbs extension. These files are the handlebar files. In your “Week 6: A Deeper Look at Template Engines and Pug” video lesson these same files were created with .pug extension. This is the main distinction between the two options in the express generator.

The other difference is in the app.js where the template engine is setup for handlebars instead of the pug template engine.



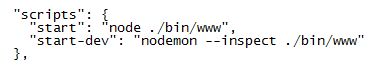
$ npm init

Edit the file “bin/www” and change the port number to your liking. This tutorial assumes it is set at 8080

var port = normalizePort(process.env.PORT || '8080');

## Start the Node Server

Add additional “start-dev” script in the package.json for aiding faster development.



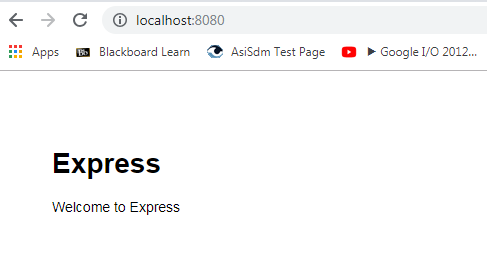
Type the following commands to start the node server.

$ npm run start ;for starting the server

$ npm run start-dev ; for starting the server in debug mode

Open your Chrome browser (you can use any browser of your choosing). This tutorial assumes it is Chrome.

Enter URL: <http://localhost:8080> You should see the following screen shot.



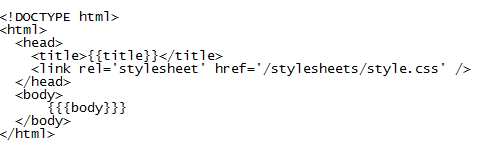
If you do not see this then you should go back to Week 6: Express Generator videos and lessons which has all the details that you need to get to this point.

## Default Index Route

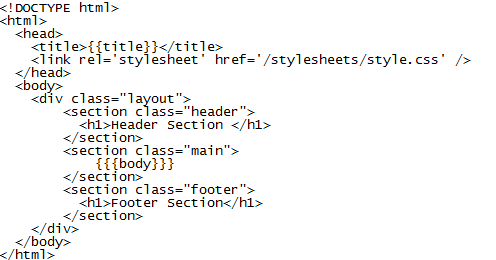
The default route is the “index” route. If you type in only <http://localhost:8080> then the default “index” route will be served by the node express server. In the “routes/index.js” we can setup the template that will be displayed for this route. We have set it up to serve “views/index.hbs” and also we can pass an object to the rendering template. In this case we are sending just the title information.

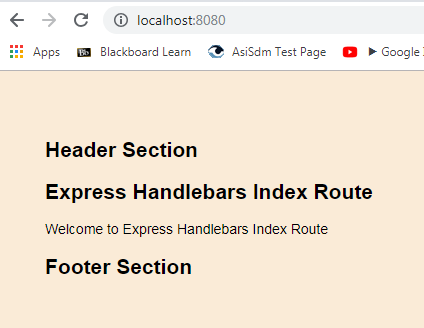
## Setting up Default Template

The “views/layout.hbs” is the main template. This is included in all the templates. The {{{body}}} expression is where all other sub templates will be placed. This template provides the opportunity to include styling sheets such as bootstrap etc. In this tutorial all styling will be in “public/stylesheets/style.css” file.

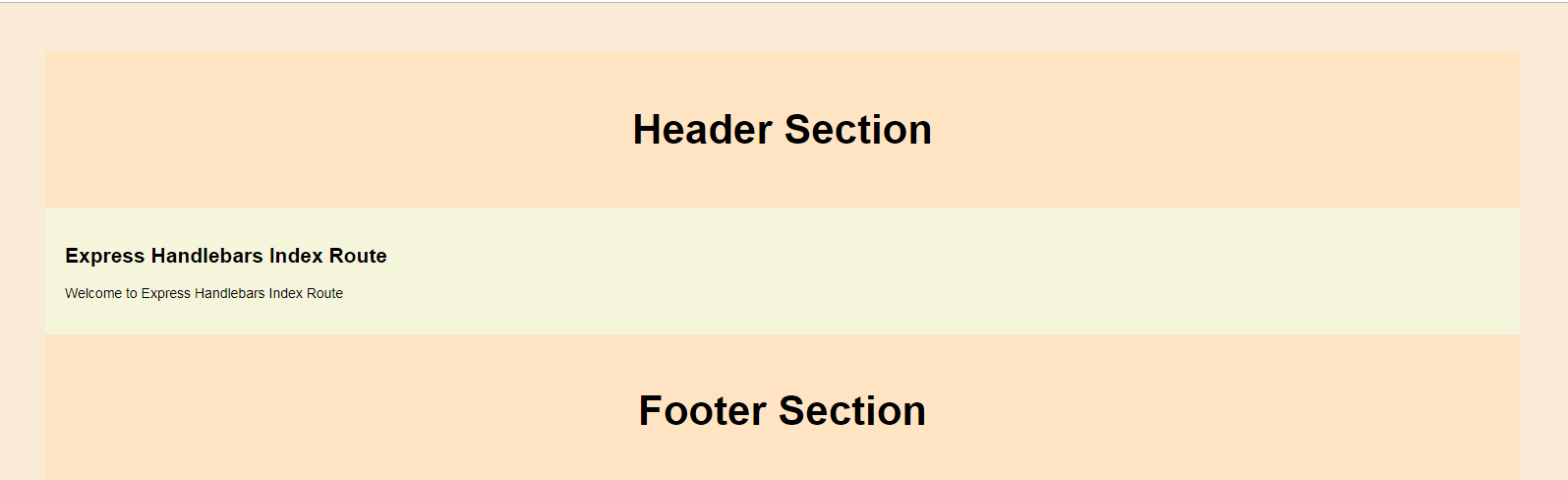


Modify the body part of the “layout.hbs” file to provide header and footer sections to all views.



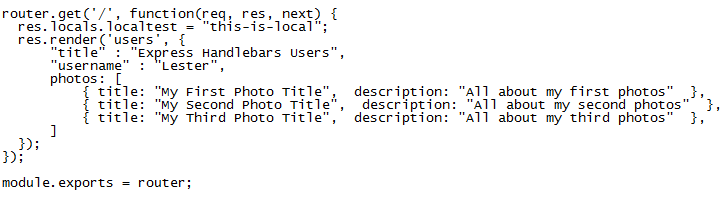


You can apply individual styles and contents to each of the sections. Modify the “routes/index.js” to send a different title string to be displayed in the “views/index.hbs”.

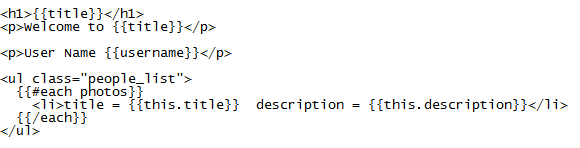


## Setting up Users Template

Another route that we have is the “users” route. This route will render a “views/users.hbs” template. We will pass an array list and iterate over this array to display the contents of the array in the template. This will provide an example of how to iterate over an array in the handlebars template.

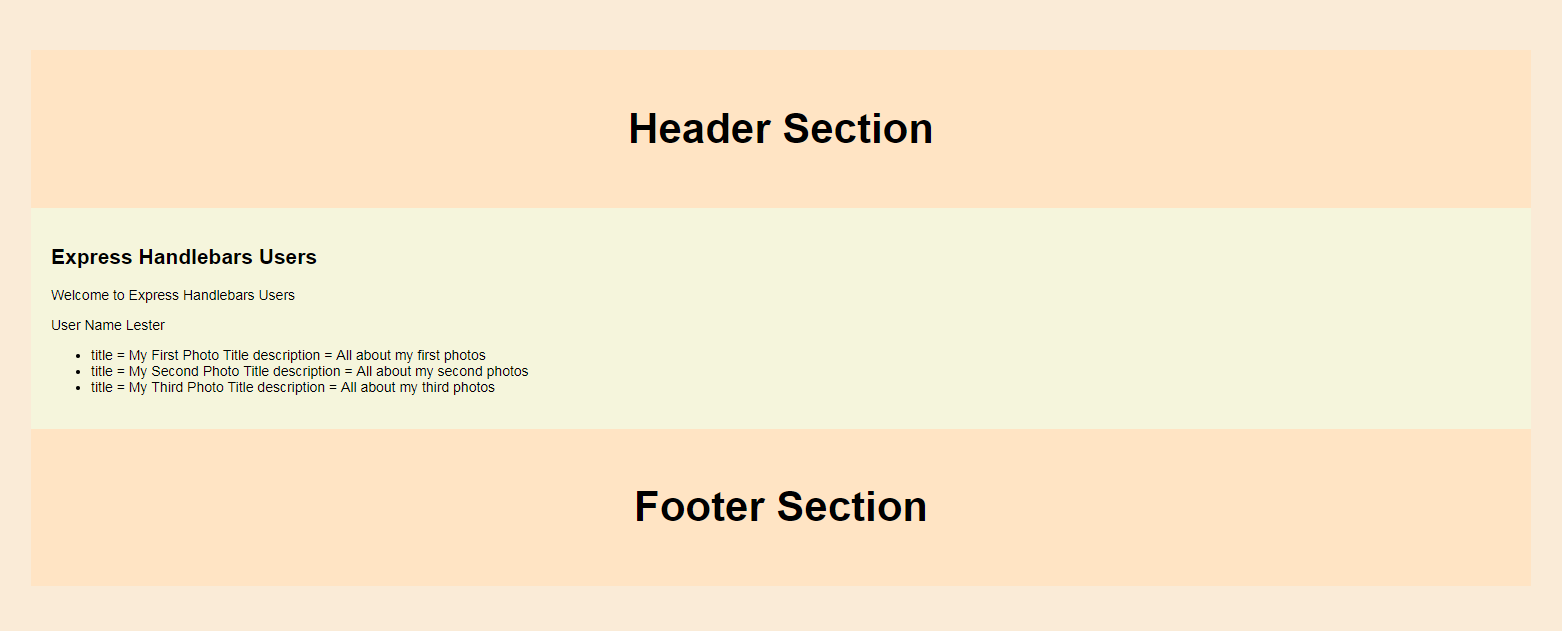


Modify the “routes/users.js” file to send an object with an array. The “photos” object is an array in this case.



Modify the “views/users.hbs” template with the contents of the above. Notice how “#each” operator is used to iterate over the “photos” array.

Use <http://localhost:8080/users> to display the users list.



## Conclusion

Hope it is clear from the above setup to see how easy it is to setup a node+express server to use handlebar templates. We have not even scratched the surface for using handlebars. But now you have a complete setup in which you can experiment and get better at using handlebar templates. Visit the handlebars documentation for learning more and to see how powerful is handlebar templates.