



## TUTORIAL

# How To Use EJS to Template Your Node Application

Node.js JavaScript

By [Chris Sev](#)

Last Validated on May 5, 2021 • Originally Published on September 19, 2020 93.9k

English

## Introduction

When quickly creating Node applications, a fast way to template your application is sometimes necessary.

Jade comes as the default template engine for Express but Jade syntax can be overly complex for many use cases.

Embedded JavaScript templates (EJS) can be used as an alternative template engine.

In this article, you will learn how to apply EJS to an Express application, include repeatable parts of your site, and pass data to the views.

## Prerequisites

If you would like to follow along with this article, you will need:

- A local development environment for Node.js. Follow How to Install Node.js and Create a

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics.

[Sign Up](#)

## Step 1 — Setting Up the Project

First, open your terminal window and create a new project directory:

```
$ mkdir ejs-demo
```

Then, navigate to the newly created directory:

```
$ cd ejs-demo
```

At this point, you can initialize a new npm project:

```
$ npm init -y
```

Next, you will need to install the `express` package:

```
$ npm install express@4.17.1
```

Then install the `ejs` package:

```
$ npm install ejs@3.1.6
```


At this point, you have a new project ready to use Express and EJS.

## Step 1 — Configuring with `server.js`

With all of the dependencies installed, let's configure the application to use EJS and set up the routes for the Index page and the About page.

Create a new `server.js` file and open it with your code editor and add the following lines of code:

---

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics. 

**Sign Up**

```
// use res.render to load up an ejs view file

// index page
app.get('/', function(req, res) {
  res.render('pages/index');
});

// about page
app.get('/about', function(req, res) {
  res.render('pages/about');
});

app.listen(8080);
console.log('Server is listening on port 8080');
```

This code defines the application and listens on port 8080 .

This code also sets EJS as the view engine for the Express application using:

```
`app.set('view engine', 'ejs');`
```

Notice how the code sends a view to the user by using `res.render()` . It is important to note that `res.render()` will look in a `views` folder for the view. So you only have to define `pages/index` since the full path is `views/pages/index` .

Next, you will create the views using EJS.


## Step 2 — Creating the EJS Partial

Like a lot of the applications you build, there will be a lot of code that is reused. These are considered *partials*. In this example, there will be three partials that will be reused on the Index page and About page: `head.ejs` , `header.ejs` , and `footer.ejs` . Let's make those files now.

Create a new `views` directory:

```
$ mkdir views
```

---

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics. 

**Sign Up**

In this directory, create a new `head.ejs` file and open it with your code editor. Add the following lines of code:

views/partials/head.ejs

```
<meta charset="UTF-8">
<title>EJS Is Fun</title>

<!-- CSS (load bootstrap from a CDN) -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-boots
<style>
  body { padding-top:50px; }
</style>
```

This code contains metadata for the head for an HTML document. It also includes Bootstrap styles.

Next, create a new `header.ejs` file and open it with your code editor. Add the following lines of code:


views/partials/header.ejs

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="/">EJS Is Fun</a>
  <ul class="navbar-nav mr-auto">
    <li class="nav-item">
      <a class="nav-link" href="/">Home</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="/about">About</a>
    </li>
  </ul>
</nav>
```

This code contains navigation for an HTML document and uses several classes from Bootstrap for styling.

Next, create a new `footer.ejs` file and open it with your code editor. Add the following

---

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics. 

Enter your email address

**Sign Up**

This code contains copyright information and uses several classes from Bootstrap for styling.

Next, you will use these partials in `index.ejs` and `about.ejs`.

## Step 3 — Adding the EJS Partial to Views

You have three partials defined. Now you can `include` them in your views.

Use `<%- include( 'RELATIVE/PATH/TO/FILE' ) %>` to embed an EJS partial in another file.

- The hyphen `<%-` instead of just `<%` to tell EJS to render raw HTML.
- The path to the partial is relative to the current file.

Then, create a new `pages` subdirectory:

```
$ mkdir views/pages
```

In this directory, create a new `index.ejs` file and open it with your code editor. Add the following lines of code:

views/pages/index.ejs

```
<!DOCTYPE html>
<html lang="en">
<head>
  <%- include( '../partials/head' ); %>
</head>
<body class="container">

<header>
  <%- include( '../partials/header' ); %>
</header>

<main>
```

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics.



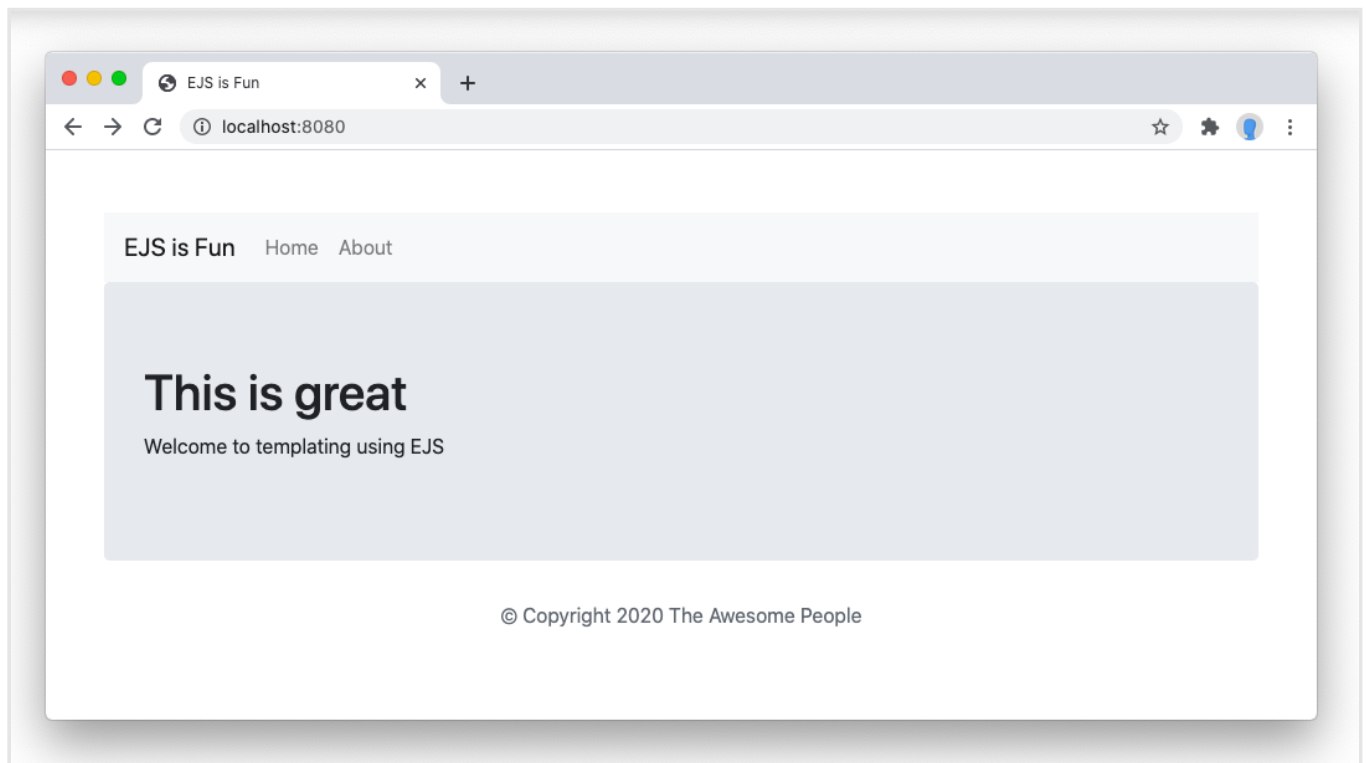
Sign Up

```
</body>  
</html>
```

Save the changes to this file and then run the application:

```
$ node server.js
```

If you visit `http://localhost:8080/` in a web browser, you can observe the Index page:



Next, create a new `about.ejs` file and open it with your code editor. Add the following lines of code:

`views/pages/about.ejs`

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <%- include('../partials/head'); %>
```

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics.



**Sign Up**

```
<div class="col-sm-8">
  <div class="jumbotron">
    <h1>This is great</h1>
    <p>Welcome to templating using EJS</p>
  </div>
</div>

<div class="col-sm-4">
  <div class="well">
    <h3>Look I'm A Sidebar!</h3>
  </div>
</div>
</div>
</main>

<footer>
  <%- include('../partials/footer'); %>
</footer>

</body>
</html>
```


This code adds a Bootstrap sidebar to demonstrate how partials can be structured to reuse across different templates and pages.

Save the changes to this file and then run the application:

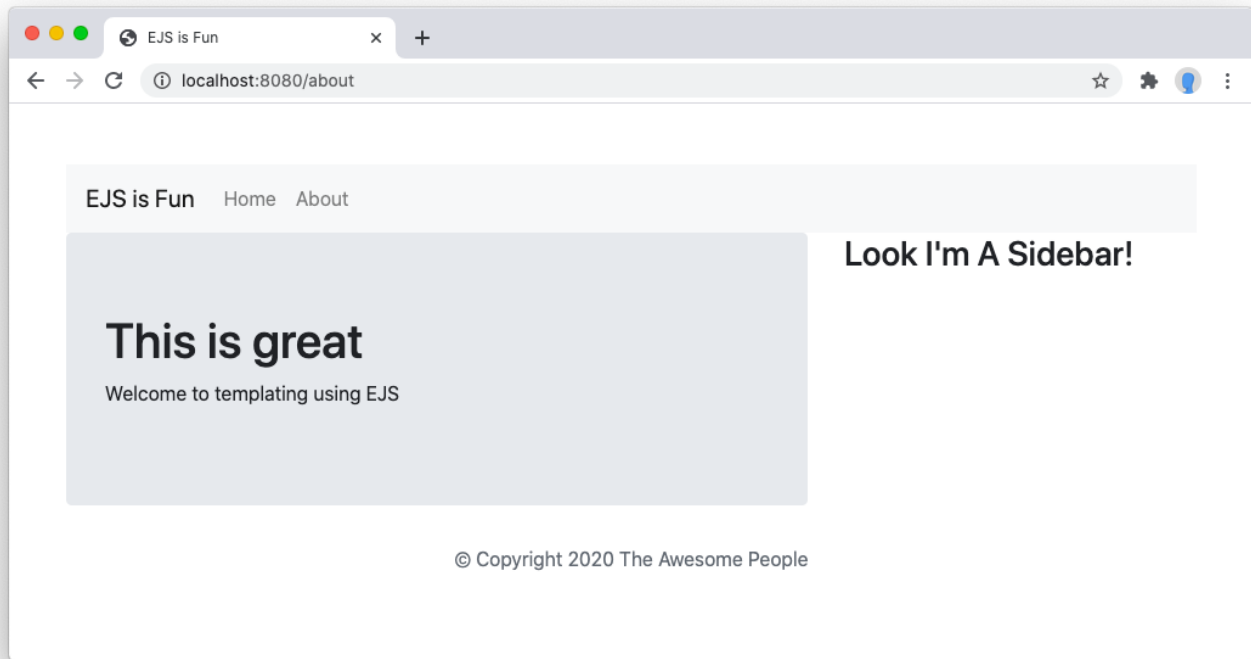
```
$ node server.js
```

If you visit `http://localhost:8080/about` in a web browser, you can observe the About page with a sidebar:

---

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics. 

**Sign Up**



Now you can start using EJS for passing data from the Node application to the views.

## Step 4 — Passing Data to Views and Partial

Let's define some basic variables and a list to pass to the Index page.

Revisit `server.js` in your code editor and add the following lines of code inside the `app.get('/')` route:

`server.js`

```
var express = require('express');
var app = express();

// set the view engine to ejs
app.set('view engine', 'ejs');

// use res.render to load up an ejs view file

// index page
app.get('/', function(req, res) {
```

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics.



Sign Up



```
    mascots: mascots,
    tagline: tagline
  });
});

// about page
app.get('/about', function(req, res) {
  res.render('pages/about');
});

app.listen(8080);
console.log('Server is listening on port 8080');
```

This code defines an array called `mascots` and a string called `tagline`. Next, let's use them in `index.ejs`.

## Rendering a Single Variable in EJS

To echo a single variable, you can use `<%= tagline %>`.

Revisit `index.ejs` in your code editor and add the following lines of code:

views/pages/index.ejs

```
<!DOCTYPE html>
<html lang="en">
<head>
  <%- include('../partials/head'); %>
</head>
<body class="container">

<header>
  <%- include('../partials/header'); %>
</header>

<main>
  <div class="jumbotron">
    <h1>This is great</h1>
    <p>Welcome to templating using EJS</p>

    <h2>Variable</h2>
```

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics.



Sign Up

```
</body>
</html>
```

This code will display the `tagline` value on the Index page.

## Looping Over Data in EJS

To loop over data, you can use `.forEach`.

Revisit `index.ejs` in your code editor and add the following lines of code:

views/pages/index.ejs

```
<!DOCTYPE html>
<html lang="en">
<head>
  <%- include('../partials/head'); %>
</head>
<body class="container">

<header>
  <%- include('../partials/header'); %>
</header>

<main>
  <div class="jumbotron">
    <h1>This is great</h1>
    <p>Welcome to templating using EJS</p>

    <h2>Variable</h2>
    <p><%= tagline %></p>

    <ul>
      <% mascots.forEach(function(mascot) { %>
        <li>
          <strong><%= mascot.name %></strong>
          representing <%= mascot.organization %>,
          born <%= mascot.birth_year %>
        </li>
      <% }); %>
    </ul>
```

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics.

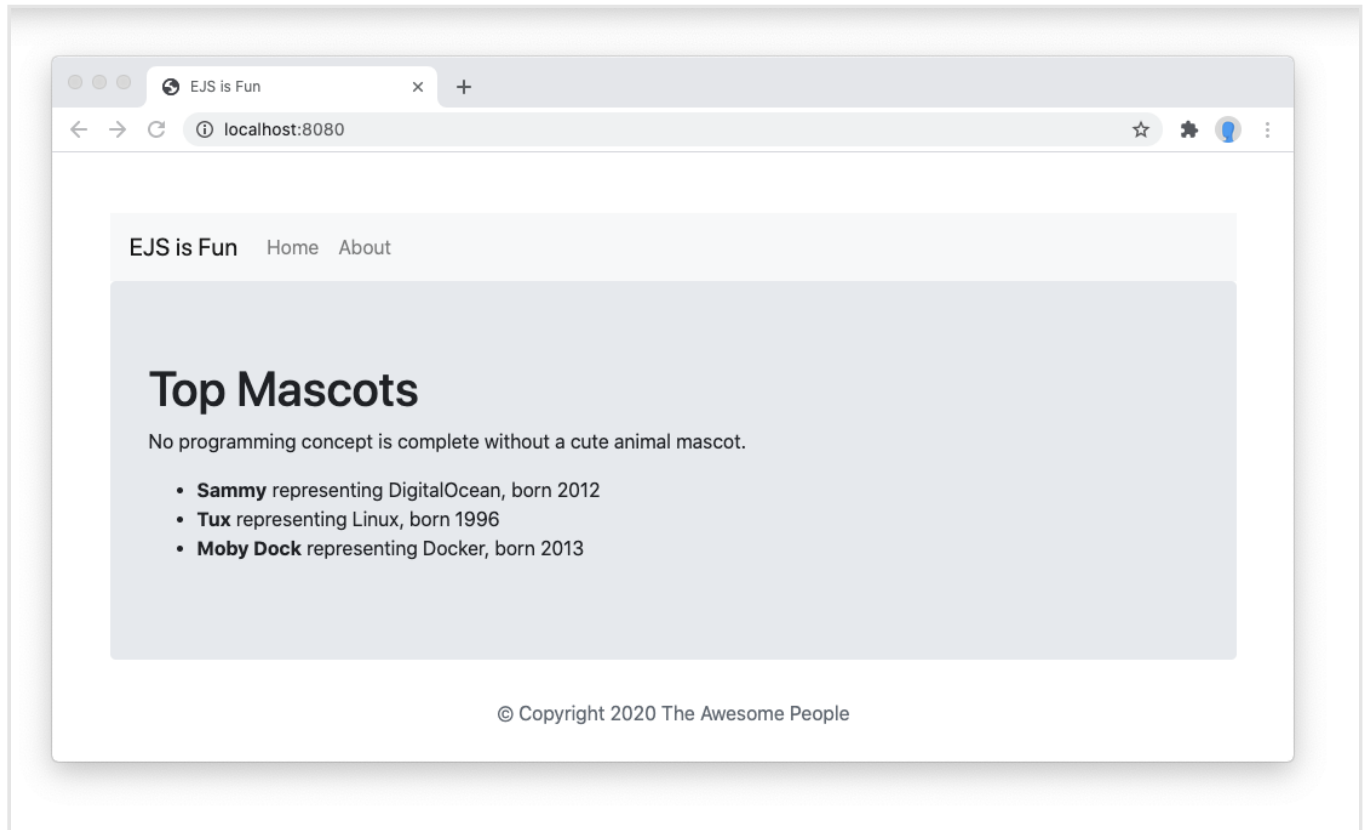


Sign Up

Save the changes to this file and then run the application:

```
$ node server.js
```

If you visit `http://localhost:8080/` in a web browser, you can observe the Index page with the mascots:



## Passing Data to a Partial in EJS

The EJS partial has access to all the same data as the parent view. But be careful. If you are referencing a variable in a partial, it needs to be defined in every view that uses the partial or it will throw an error.

You can also define and pass variables to an EJS partial in the include syntax like this:

```
views/pages/about.ejs
```

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics. ✕

Sign Up

But you need to again be careful about assuming a variable has been defined.

If you want to reference a variable in a partial that may not always be defined, and give it a default value, you can do so like this:

views/partials/header.ejs

```
...  
<em>Variant: <%= typeof variant != 'undefined' ? variant : 'default' %></em>  
...
```

In the line above, the EJS code is rendering the value of `variant` if it's defined, and `default` if not.

## Conclusion

In this article, you learned how to apply EJS to an Express application, include repeatable parts of your site, and pass data to the views.

EJS lets you build applications when you do not require additional complexity. By using partials and having the ability to easily pass variables to your views, you can build some great applications quickly.

Consult [the EJS documentation](#) for additional information on features and syntax. Consult [Comparing JavaScript Templating Engines: Jade, Mustache, Dust and More](#) for understanding the pros and cons of different view engines.

Was this helpful?

Yes

No



[Report an issue](#)

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics.



Enter your email address

Sign Up

Founder of scotch.io. Slapping the keyboard until good things happen.

## Still looking for an answer?



Ask a question



Search for more help

### RELATED

#### Join the DigitalOcean Community



##### Join 1M+ other developers and:

- Get help and share knowledge in Q&A
- Subscribe to topics of interest
- Get courses & tools that help you grow as a developer or small business owner

Join Now

How To Install the Deno JavaScript Runtime on Ubuntu 20.04

 [Tutorial](#)

Understanding Closures in JavaScript

 [Tutorial](#)

### Comments

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics.



Enter your email address

Sign Up

Leave a comment...

Sign In to Comment

^ [stadibenker](#) September 29, 2020

0 Great article! Thank you so much! :)

[Reply](#) [Report](#)

^ [CarsonBurke](#) May 11, 2021

0 Thank you! Very helpful!

[Reply](#) [Report](#)



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

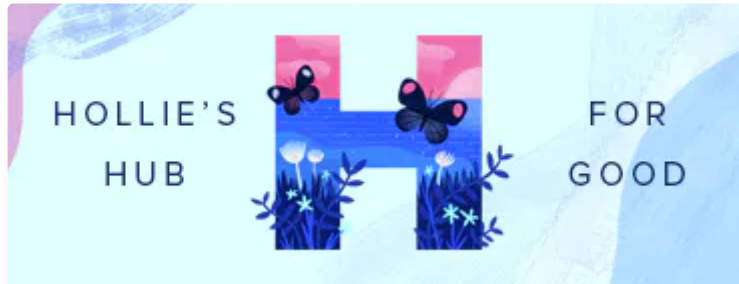


**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics.



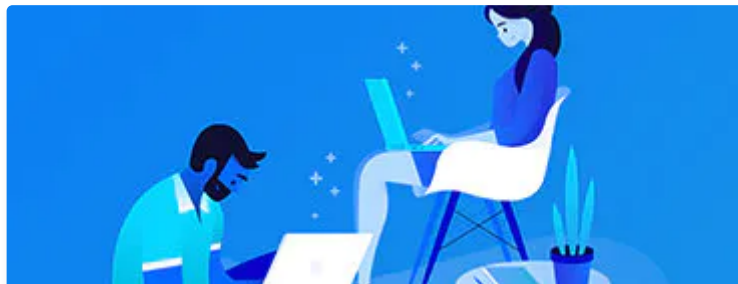
Enter your email address

Sign Up



#### HOLLIE'S HUB FOR GOOD

Working on improving health and education, reducing inequality, and spurring economic growth? We'd like to help.



#### BECOME A CONTRIBUTOR

You get paid; we donate to tech nonprofits.

Featured on [Community](#) [Kubernetes Course](#) [Learn Python 3](#) [Machine Learning in Python](#)  
[Getting started with Go](#) [Intro to Kubernetes](#)

DigitalOcean Products [Virtual Machines](#) [Managed Databases](#) [Managed Kubernetes](#) [Block Storage](#)

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics.



**Sign Up**

DigitalOcean makes it simple to launch in the cloud and scale up as you grow – whether you're running one virtual machine or ten thousand.

[Learn More](#)



© 2021 DigitalOcean, LLC. All rights reserved.

#### Company

[About](#)  
[Leadership](#)  
[Blog](#)  
[Careers](#)  
[Partners](#)  
[Referral Program](#)  
[Press](#)  
[Legal](#)  
[Security & Trust Center](#)

#### Products

[Pricing](#)  
[Products Overview](#)  
[Droplets](#)  
[Kubernetes](#)

#### Community

[Tutorials](#)  
[Q&A](#)  
[Tools and Integrations](#)  
[Tags](#)

#### Contact

[Get Support](#)  
[Trouble Signing In?](#)  
[Sales](#)  
[Report Abuse](#)

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics.



**Sign Up**



---

**Sign up for our newsletter** Get the latest tutorials on SysAdmin and open source topics.



**Sign Up**