

Template Engine In ExpressJS

Topics Covered:-

- What is template engine?
- How to use Template Engine?
- Sending variables to EJS and show on frontend

What is template engine?

The Template Engine is used to merge HTML page with the *data* from your program. The engine is simple and powerful.

At runtime, it replaces variables in a template file with actual values and transforms the template into an HTML file sent to the client.

For Eg:-

```
Welcome <%= username %>
```

[Copy](#)

Here username is a variable. It will replace the username with the value received from the backend. Say username variable contains "Rahul" value then the output will be:-

```
Welcome Rahul
```

[Copy](#)

There are number of template engines available some of them are given below:-

- EJS
- Jade(pug)
- Vash
- Mustache
- Dust.js
- Nunjucks
- Handlebars
- ATPL
- HAML

In this Tutorial, we are going to learn about EJS Template Engine. EJS Stands for Embedded JavaScript.

HOW TO USE TEMPLATE ENGINE

First Install EJS into your project using NPM as below.

```
npm install --save ejs
```

[Copy](#)

- Create a folder named “Views” in project’s root directory Because by default Express.js searches all the views(templates) in the views folder under the root folder. If you don’t want your folder to be named as views you can change name using views property in express.
eg : the name of your folder is *myviews* then in our main file in this case app.js we write `app.set('views', './myviews);`
- All the templates file are saved with template engine’s extension and in views folder. Here we are using EJS so all templates are saved with .ejs extension in views folder.

USING TEMPLATE ENGINE

EJS files are simple to write because in this file we write simple HTML and write logic, backend variables between these tags:-

- `<% %>`(to write logics)
- `<%= %>`(when we want to display content use this)
- `<%- %>`

Here in this tutorial, we have two files one is main JS file i.e. `app.js` , which creates a server and one template file(`display.ejs`).

1. In `app.js` first create a basic server like this:

```
var express = require("express"); //requiring express module
var app = express(); //creating express instance

app.get("/", function(req, res) {
  res.send("hello");
});
app.listen(3000, function() {
  console.log("Node server is running..");
});
```

Copy

2. Now to send templates to frontend, first we set the view engine as follows:-

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

Copy

Add the following line to `app.js`

3. After setting the view engine we create a simple `display.ejs` file in view folder with a simple ‘*hello user*’ message in it.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>templates</title>
</head>
<body>
  <p>Hello User</p>
</body>
</html>
Copy.

```

4. Now to send the file to frontend we have a function called i.e. `res.render()` , it renders a view and sends the rendered HTML string to the client.

To render our file in previous code we replace `res.send("hello");` with `res.render('display');`

```

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

app.set("view engine", "ejs")

app.get('/', function (req, res) {
  //rendering our ejs template
  res.render('demo');
});
var server = app.listen(3000, function () {
  console.log('Node server is running..');
});
Copy.

```

Run the following code as “node app.js” in terminal and press enter:-

```

c:\Express-project\project>node app.js
Copy.

```

The server is now started you can see the message in terminal

```

c:\Express-project\project>node app.js
Node server is running..
Copy.

```

point your browser to `http://localhost:3000` and you will see the message “Hello User”.

Sending variables to EJS and show on frontend

If we want to send data from backend and use it in EJS file we can do this in the following manner:-

Here instead of user in our display.ejs file I want to specify the name of the user. We can send data where we are rendering the file.

Eg:- `res.render('display',{variable_name:'value'});`

Say the username is ‘Rahul’ then our code will be:-

```
res.render('display', {user: 'Rahul'});
```

Copy

And to display this on front end we write this user variable inside `<%= %>` like this:-

```
<p>Hello <%= user%></p>
```

Copy

Now restart the server and run the code again and on refreshing the page you will see *Hello Rahul* instead of *Hello User*.

Sending Arrays or Objects(JSON data)

-We can send any sort of data to the frontend. Suppose if I have a JSON data having an array named `authors` of author object having name and article, and I want to display it on frontend via a list, I can send an array to EJS and then loop through it and display the data. For this follow the given example:-

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

app.set("view engine", "ejs");
var authors = [
  {
    name: "Andrew",
    article: "node.js"
  },
  {
    name: "Brad",
    article: "express.js"
  },
  {
    name: "John",
    article: "mongoose"
  }
];
app.get("/", function(req, res) {
  res.render("display", { authors });
});
var server = app.listen(3000, function() {
  console.log("Node server is running..");
});
```

Copy

-Copy and paste this code in your `app.js` file here I have created an array named *“authors”* and I have rendered it with the `ejs` file. Note:- you can send the data by assigning it to a variable `{values: data}` or directly `{data}` in direct case you can access it with `data` and by assigning it you can access it with whatever variable you have assigned in given example by values.

Now to access the author array in EJS follow the below EJS code:-

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>

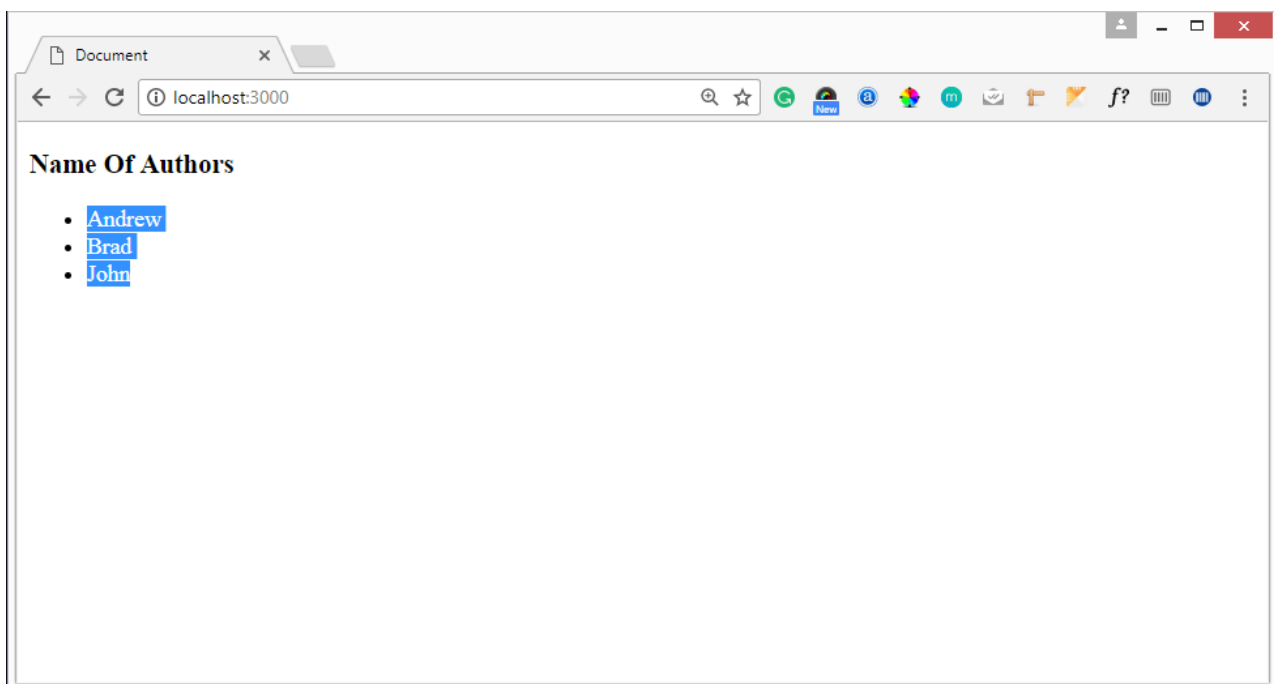
<body>
  <h3>Name Of Authors</h3>
  <ul>
    <!-- looping through authors array -->
    <% authors.forEach(function(author){ %>
      <li>
        <%= author.name %>
      </li>
    <% }) %>
  </ul>
</body>
</html>
Copy

```

Here In the above EJS code, we have used a `forEach` loop to iterate through “authors” array and to access data of *authors* array one by one.

Save *app.js* and *display.ejs* with the following code and then run the code, now as you refresh the web page(<http://localhost:3000>) you will see the following output.

Final Output:-



Learn More-

- [Update Document Database](#)
- [Introduction To Nodejs](#)

- Node Js Buffer