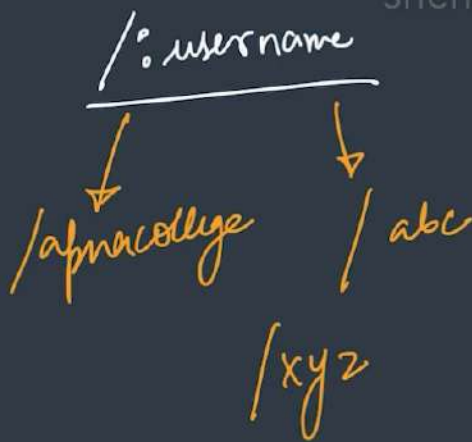


# Templating

**EJS** (Embedded JavaScript templates)

EJS is a simple templating language that lets you generate HTML markup with plain JavaScript.

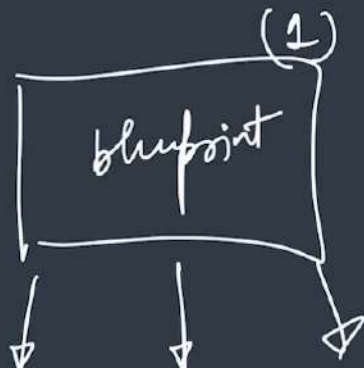
snehagupta7385@gmail.com



# Templating

**EJS** (Embedded JavaScript templates)

EJS is a simple templating language that lets you generate HTML markup with plain JavaScript.



```
shradhakhapra@Shradhas-MacBook-Air  
EJSdir % npm init -y  
Wrote to /Users/shradhakhapra/WebD
```

```
[shradhakhapra@Shradhas-MacBook-Air]  
EJSdir % npm i express
```

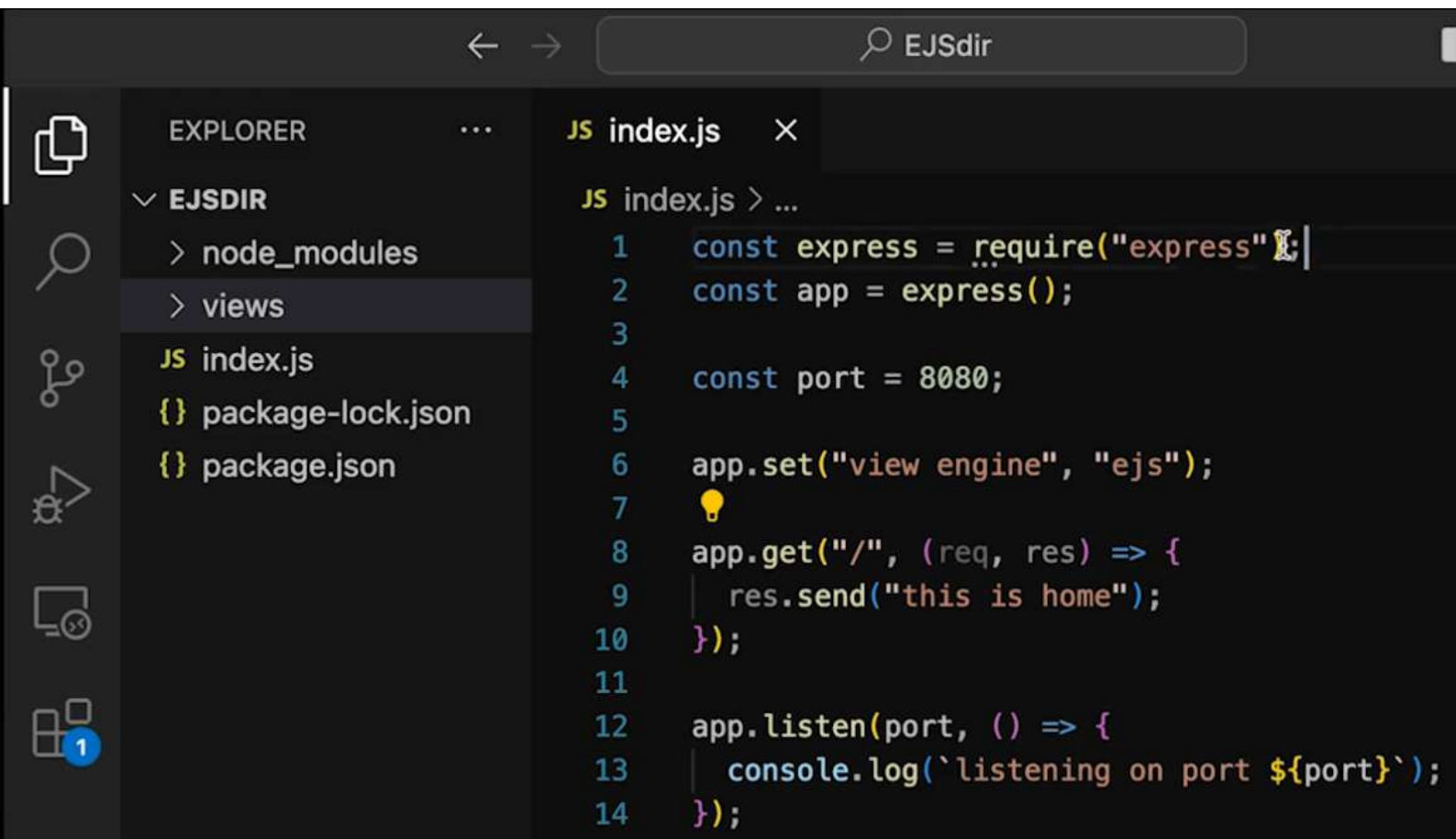
shradnakhapra@Shradha  
EJSdir % npm i ejs

```
ShradhaKhapra@Shradhas-MacB  
EJSdir % touch index.js
```

# Using EJS

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

app.get("/", (req, res) => {
  res.render("home.ejs");
});
```





JS index.js

<> home.ejs X



views > <> home.ejs > html

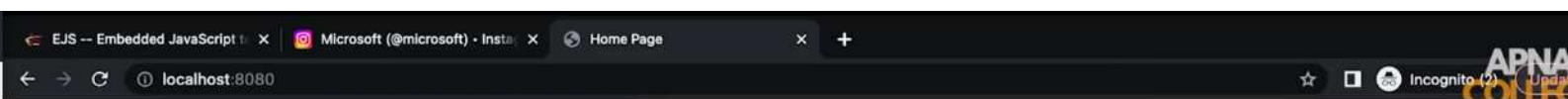
```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Home Page</title>
7   </head>
8   <body>
9     <h1>This is the home page</h1>
10    <p>
11      Lorem ipsum dolor sit, amet consectetur adipisicing elit. Unde,
12      voluptatibus, recusandae ea laudantium accusamus ab enim necessitatibus,
13      odit est nisi ut debitis beatae quasi possimus tempora dicta distinctio,
14      modi quibusdam.
15    </p>
16    <button>click me!</button>
```

JS index.js X

<> home.ejs

JS index.js > ...

```
1  const express = require("express");
2  const app = express();
3
4  const port = 8080;
5
6  app.set("view engine", "ejs");
7
8  app.get("/", (req, res) => {
9    res.render("home.ejs");
10  });
11
12  app.listen(port, () => {
13    console.log(`listening on port ${port}`);
14  });
15
```




## This is the home page

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Unde, voluptatibus, recusandae ea laudantium accusamus ab enim necessitatibus odit est nisi ut debitis beatae quasi possimus tempora dicta distinctio modi quibusdam.

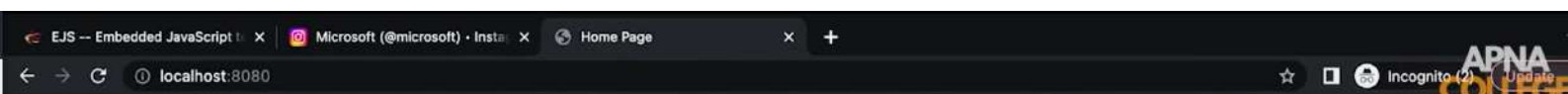
click me!

JS index.js

<> home.ejs

JS index.js >  app.get("/") callback

```
1  const express = require("express");
2  const app = express();
3
4  const port = 8080;
5
6  app.set("view engine", "ejs");
7
8  app.get("/", (req, res) => {
9    res.render("home");
10 });
11
12 app.listen(port, () => {
13   console.log(`listening on port ${port}`);
14 });
15
```



## This is the home page

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Unde, voluptatibus, recusandae ea laudantium accusamus ab enim necessitatibus odit est nisi ut debitis beatae quasi possimus tempora dicta distinctio modi quibusdam.

click me!

<% EJS %>

## EJS language support v1.3.3

DigitalBrainstem | 906,564 | ★★★★★ (2

2019 - EJS language support for Visual Studio Co...

Disable ▾

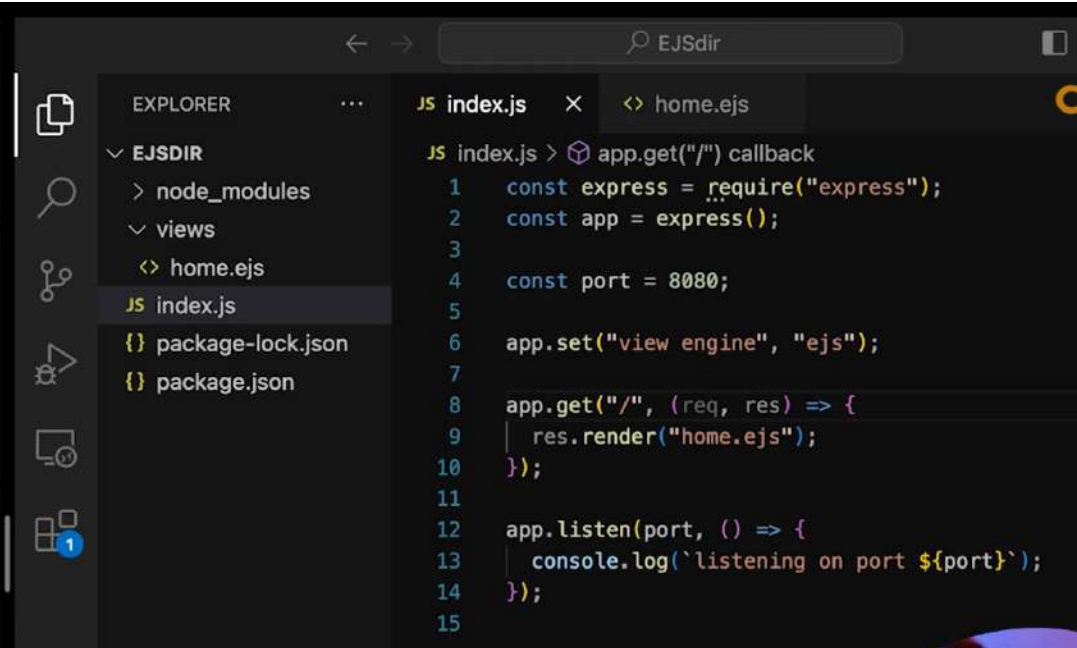
Uninstall ▾



This extension is enabled globally.

```
shradhakhapra@Shradhas-MacBook-Air:~$ cd ..
shradhakhapra@Shradhas-MacBook-Air:~$ Backend % ls
Class1          ExpressDir
EJSdir
shradhakhapra@Shradhas-MacBook-Air:~$ Backend % nodemon EJSdir/index.js

[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: .js, .mjs, .cjs, .json
[nodemon] starting `node EJSdir/index.js`
listening on port 8080
```



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays the project structure for 'EJSdir':

- node\_modules
- views
  - home.ejs
- index.js
- package-lock.json
- package.json

The main editor window shows the content of 'index.js':

```
JS index.js > app.get("/") callback
1  const express = require("express");
2  const app = express();
3
4  const port = 8080;
5
6  app.set("view engine", "ejs");
7
8  app.get("/", (req, res) => {
9    res.render("home.ejs");
10 });
11
12 app.listen(port, () => {
13   console.log(`listening on port ${port}`);
14 });
15
```



JS index.js ×

<> home.ejs

JS index.js > ...

```
1  const express = require("express");
2  const app = express();
3
4  const port = 8080;
5
6  app.set("view engine", "ejs");
7
8  app.get("/", (req, res) => {
9    res.render("home.ejs");
10 });
11
12 app.get("/hello", (req, res) => {
13   res.send("hello");
14 });
15
16 app.listen(port, () => {
17   console.log(`listening on port ${port}`);
18 });
19
```





localhost:8080/hello

hello



JS index.js × <> home.ejs

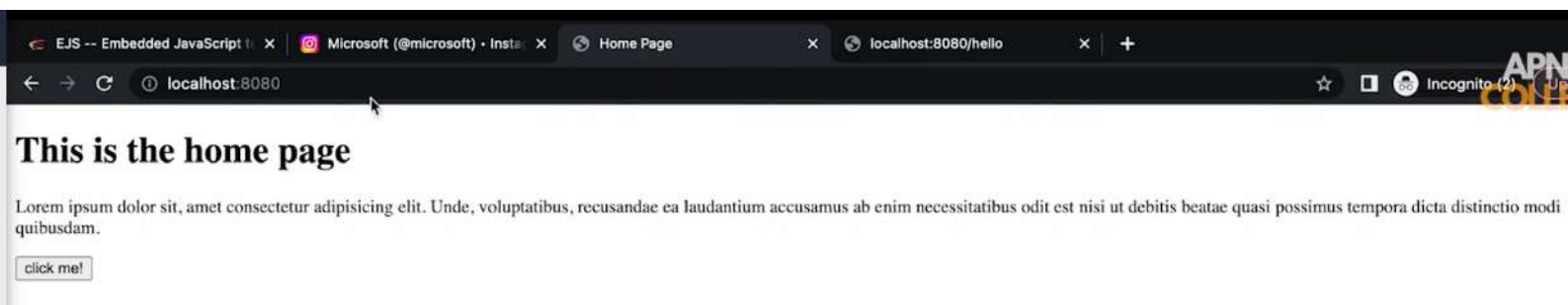


JS index.js > ...

```
1  const express = require("express");
2  const app = express();
3  const path = require("path");
4
5  const port = 8080;
6
7  app.set("view engine", "ejs");
8  app.set("views", path.join(__dirname, "/views"));
9
10 app.get("/", (req, res) => {
11   res.render("home.ejs");
12 });
13
14 app.get("/hello", (req, res) => {
15   res.send("hello");
16 });
17
18 app.listen(port, () => {
19   console.log(`listening on port ${port}`);
20 });
```

# Views Directory

```
const path = require("path");  
app.set("views", path.join(__dirname, "/views"));
```



# Interpolation Syntax

Interpolation refers to **embedding expressions** into marked up text.

## Tags

- `<%` 'Scriptlet' tag, for control-flow, no output
- `<%=` 'Whitespace Slurping' Scriptlet tag, strips all whitespace before it
- `<%=` Outputs the value into the template (HTML escaped)
- `<%=` Outputs the unescaped value into the template
- `<%=` Comment tag, no execution, no output
- `<%=` Outputs a literal '<='
- `%>` Plain ending tag
- `->` Trim-mode ('newline slurp') tag, trims following newline
- `_>` 'Whitespace Slurping' ending tag, removes all whitespace after it



JS index.js

<> home.ejs X



views > <> home.ejs > html > body > h3 > ?

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <title>Home Page</title>
7    </head>
8    <body>
9      <h1>This is the home page 1 + 2</h1>
10     <h3><%= 1 + 2 %></h3>
11     <p>|
12       Lorem ipsum dolor sit, amet consectetur
13       voluptatibus, recusandae ea laudantium
14       odit est nisi ut debitis beatae quasi
15       modi quibusdam.
16     </p>
17     <button>click me!</button>
18   </body>
19 </html>
```

**This is the home page 1 + 2**

**3**

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Unde, voluptatibus, recusandae ea laudantium accusamus ab enim necessitatibus odit est nisi ut debitis beatae quasi possimus tempora dicta distinctio modi quibusdam.

[click me!](#)



JS index.js

<> home.ejs

views > <> home.ejs > html > body > h3 > ?

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-
6      <title>Home Page</title>
7    </head>
8    <body>
9      <h1>This is the home page 1 * 2</h1>
10     <h3><%= "apnacollege".toUpperCase() %></h3>
11     <p>
12       Lorem ipsum dolor sit, amet consectetur ad
13       voluptatibus, recusandae ea laudantium acc
14       odit est nisi ut debitis beatae quasi poss
15       modi quibusdam.
16     </p>
17     <button>click me!</button>
18   </body>
19 </html>
```

# This is the home page 1 \* 2

APNACOLLEGE

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Unde, voluptatibus, recusandae ea laudantium accusamus ab enim necessitatibus odit est nisi ut debitis beatae quasi possimus tempora dicta distinctio modi quibusdam.

[click me!](#)

JS index.js

<> home.ejs X

views > <> home.ejs > html > body > h3 > ?

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width,
6        <title>Home Page</title>
7    </head>
8    <body>
9      <h1>This is the home page 1 * 2</h1>
10     <h3><%= ["hello", "bonjour", "namaste"][[2]] %></h3>
11     <p>
12       Lorem ipsum dolor sit, amet consectetur adipisicing
13       voluptatibus, recusandae ea laudantium accusamus
14       odit est nisi ut debitis beatae quasi possimus te
15       modi quibusdam.
16     </p>
17     <button>click me!</button>
18   </body>
19 </html>
```

# This is the home page 1 \* 2

namaste

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Unde, voluptatibus, recusandae ea laudantium accusamus ab enim necessitatibus odit est nisi ut debitis beatae quasi possimus tempora dicta distinctio modi quibusdam.

click me!

## Passing data to EJS

```
app.get("/rolldice", (req, res) => {  
  let num = Math.floor(Math.random() * 6) + 1;  
  res.render("rollDice.ejs", { diceVal: num });  
});
```

```
<h1>Your dice gave value : <%= diceVal %></h1>
```



EXPLORER

...

✓ EJS DIR    

> node\_modules

✓ views

<> rolldice.ejs

<> home.ejs

JS index.js

{ } package-lock.json

{ } package.json



JS index.js

<> home.ejs

JS index.js > ...

```
1  const express = require('express');
2  const app = express();
3  const path = require('path');
4
5  const port = 8080;
6
7  app.set("view engine", 'ejs');
8  app.set("views", path.resolve(__dirname, 'views'));
9
10 app.get("/", (req, res) => {
11   res.render("home.ejs");
12 });
13
14 app.get("/hello", (req, res) => {
15   res.send("hello");
16 });
17
18 app.get("/rolldice", (req, res) => {
19   res.send("hello");
20 });
21
22 app.listen(port, () => {
23   console.log(`listening on port ${port}`);
```

1

JS index.js

rolldice.ejs

home.ejs

views > <> rolldice.ejs > html > body > h1

1

<!DOCTYPE html>

2

<html lang="en">

3

<head>

4

<meta charset="UTF-8" />

5

<meta name="viewport" content="width=device-width, initial-scale=1.0">

6

<title>Roll Dice</title>

7

</head>

8

<body>

9

<h1>Dice gave value : <%= Math.floor(Math.random() \* 6) + 1 %></h1>

10

</body>


11

</html>

12



JS index.js X

JS index.js >  app.get("/rolldice") callback

```
1  const express = require("express");
2  const app = express();
3  const path = require("path");
4
5  const port = 8080;
6
7  app.set("view engine", "ejs");
8  app.set("views", path.join(__dirname, "/views"));
9
10 app.get("/", (req, res) => {
11   res.render("home.ejs");
12 });
13
14 app.get("/hello", (req, res) => {
15   res.send("hello");
16 });
17
18 app.get("/rolldice", (req, res) => {
19   res.render("rolldice.ejs");
20 });
21
22 app.listen(port, () => {
23   console.log(`listening on port ${port}`);
```



JS index.js × <> rolldice.ejs

JS index.js > ...

```
1  const express = require("express");
2  const app = express();
3  const path = require("path");
4
5  const port = 8080;
6
7  app.set("view engine", "ejs");
8  app.set("views", path.join(__dirname, "/views"));
9
10 app.get("/", (req, res) => {
11   res.render("home.ejs");
12 });
13
14 app.get("/hello", (req, res) => {
15   res.send("hello");
16 });
17 💡
18 app.get("/rolldice", (req, res) => {
19   let diceVal = Math.floor(Math.random() * 6) + 1;
20   res.render("rolldice.ejs", { num: diceVal });
21 });
22
23 app.listen(port, () => {
24   console.log(`listening on port ${port}`);
25 });
```

views > <> rolldice.ejs > html > body > h1 > ?

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Roll Dice</title>
7   </head>
8   <body>
9     <h1>Dice gave value : <%= num %></h1>
10  </body>
11 </html>
12
```

← → ↻ ⓘ localhost:8080/rolldice

**Dice gave value : 2**

JS index.js



&lt;&gt; rolldice.ejs

snehagu

JS index.js &gt; app.get("/rolldice") callback &gt; diceVal

```
1  const express = require("express");
2  const app = express();
3  const path = require("path");
4
5  const port = 8080;
6
7  app.set("view engine", "ejs");
8  app.set("views", path.join(__dirname, "/views"));
9
10 app.get("/", (req, res) => {
11   res.render("home.ejs");
12 });
13
14 app.get("/hello", (req, res) => {
15   res.send("hello");
16 });
17
18 app.get("/rolldice", (req, res) => {
19   let diceVal = Math.floor(Math.random() * 6) + 1;
20   res.render("rolldice.ejs", { diceVal });
21 });
22
23 app.listen(port, () => {
24   console.log(`listening on port ${port}`);
25 });
```

**Dice gave value : 5**

# Instagram EJS

Create a basic template for instagram page based on following route :

**/ig/:username**

*/ig/afna college*  
*/ig/rahul kumar*  
↓





```

6
7 app.set("view engine", "ejs");
8 app.set("views", path.join(__dirname, "/views"));
9
10 app.get("/", (req, res) => {
11   res.render("home.ejs");
12 });
13
14 app.get("/ig/:username", (req, res) => {
15   let { username } = req.params;
16   console.log(username);
17 });
18
19 app.get("/hello", (req, res) => {
20   res.send("hello");
21 });
22
23 app.get("/rolldice", (req, res) => {
24   let diceVal = Math.floor(Math.random() *
25   res.render("rolldice.ejs", { diceVal });
26 });
27

```

JS index.js >  app.get("/ig/:username") callback

```
1  const express = require("express");
2  const app = express();
3  const path = require("path");
4
5  const port = 8080;
6
7  app.set("view engine", "ejs");
8  app.set("views", path.join(__dirname, "/views"));
9
10 app.get("/", (req, res) => {
11   res.render("home.ejs");
12 });
13
14 app.get("/ig/:username", (req, res) => {
15   let { username } = req.params;
16   res.render("instagram.ejs", {username})
17 }
18 );
19
20 app.get("/hello", (req, res) => {
21   res.send("hello");
22 });
23
24 app.get("/rolldice", (req, res) => {
25   let diceVal = Math.floor(Math.random() * 6) + 1;
26   res.render("rolldice.ejs", {diceVal});
```



JS index.js

<> instagram.ejs X

<> rolldice.ejs

views > <> instagram.ejs > html > body > h2 > ?

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Instagram</title>
7    </head>
8    <body>
9      <h2>This page belongs to <%= username %></h2>
10     <button>Follow</button>
11     <button>Message</button>
12   </body>
13 </html>
14
```

snehagu

# Conditional Statements in EJS

## Adding conditions inside EJS

```
<% if(diceVal == 6) { %>  
<h2>Nice! Roll dice again.</h2>  
<% } %>
```

JS index.js

<> rolldice.ejs ×

views > <> rolldice.ejs > html > body > ? > h2

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device
6      <title>Roll Dice</title>
7    </head>
8    <body>
9      <h1>Dice gave value : <%= diceVal %></h1>
10     <% if(diceVal == 6) { %>
11       <h2>Nice! Roll dice again.</h2>
12     <% } %>
13   </body>
14 </html>
```

**Dice gave value : 6**

**Nice! Roll dice again.**

## Loops in EJS

```
<% for(user of followers) { %>  
<li><%= user %></li>  
<% } %>
```

JS index.js



<> rolldice.ejs

<> instagram.ejs

JS index.js > app.get("/ig/:username") callback > followers

```
8   app.set("views", path.join(__dirname, "/views"));
9
10  app.get("/", (req, res) => {
11    res.render("home.ejs");
12  });
13
14  app.get("/ig/:username", (req, res) => {
15    const followers = ["adam", "bob", "steve", "abc"];
16    let { username } = req.params;
17    res.render("instagram.ejs", { username, followers });
18  });
```

JS index.js

<> instagram.ejs X

views > <> instagram.ejs > html > body > ul > ?

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <title>Instagram</title>
7    </head>
8    <body>
9      <h2>This page belongs to @<%= username %>
10     <button>Follow</button>
11     <button>Message</button>
12
13     <h3>Accounts that follow you :</h3>
14     <ul>
15       <% for(let name of followers) { %>
16         <li>name</li>
17       <% } %>
18     </ul>
19   </body>
20 </html>
21
```



**This page belongs to @apnacollege**

Follow

Message

**Accounts that follow you :**

- name
- name
- name
- name

sneh



JS index.js

<> instagram.ejs X

views > <> instagram.ejs > html > body > ul > ? > li

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width">
6      <title>Instagram</title>
7    </head>
8    <body>
9      <h2>This page belongs to @<%= username %></h2>
10     <button>Follow</button>
11     <button>Message</button>
12
13     <h3>Accounts that follow you :</h3>
14     <ul>
15       <% for(let name of followers) { %>
16         <li><%= name %></li>
17       <% } %>
18     </ul>
19   </body>
20 </html>
```

**This page belongs to @apnacollege**

Follow

Message

**Accounts that follow you :**

- adam
- bob
- steve
- abc

Please download data.json file for the next part from this link :  
<https://github.com/apna-college/Delta/blob/main/EJS/data.json>

snehagupta7385@gmail.com

# Instagram page with EJS

```
const instaData = require("./data.json");
```

JS index.js X

JS index.js >  app.get("/ig/:username") callback

```
2   const app = express();
3   const path = require("path");
4
5   const port = 8080;
6
7   app.set("view engine", "ejs");
8   app.set("views", path.join(__dirname, "/views"));
9
10  app.get("/", (req, res) => {
11    res.render("home.ejs");
12  });
13
14  app.get("/ig/:username", (req, res) => {
15    const instaData = require("../data.json");
16    console.log(instaData);
17    res.render("instagram.ejs");
18  });
19
20  app.get("/hello", (req, res) => {
21    res.send("hello");
22  });
23
```

JS index.js

<> instagram.ejs ●

views > <> instagram.ejs > html > body > h2 > ?

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <title>Instagram</title>
7    </head>
8    <body>
9      <h2>This page belongs to @<%= username %></h2>
10     <button>Follow</button>
11     <button>Message</button>
12   </body>
13 </html>
14
```

**This page belongs to @**

Follow

Message





JS index.js X <> instagram.ejs

JS index.js >  app.get("/ig/:username") callback


```
3   const path = require('path');
4
5   const port = 8080;
6
7   app.set("view engine", "ejs");
8   app.set("views", path.join(__dirname, "/views"));
9
10  app.get("/", (req, res) => {
11    res.render("home.ejs");
12  });
13
14  app.get("/ig/:username", (req, res) => {
15    const instaData = require("../data.json");
16    res.render("instagram.ejs", { instaData });
17  });
18
19  app.get("/hello", (req, res) => {
20    res.send("hello");
```



```
app.get("/ig/:username", (req, res) => {  
  const instaData = require("../data.json");  
  res.render("instagram.ejs", { data: instaData });  
});
```

<> instagram.ejs

JS index.js X

JS index.js >  app.get("/ig/:username") callback

```
0
7   app.set("view engine", "ejs");
8   app.set("views", path.join(__dirname, "/views"));
9
10  app.get("/", (req, res) => {
11    res.render("home.ejs");
12  });
13
14  app.get("/ig/:username", (req, res) => {
15    let { username } = req.params;
16    const instaData = require("../data.json");
17    const data = instaData[username];
18    console.log(data);
19    res.render("instagram.ejs", { data });
20  });
21
```

← → ↻ ⓘ localhost:8080/ig/cats

**This page belongs to @[object Object]**

Follow

Message

deck.js

listening on port 8080

```
{  
  name: 'cats',  
  followers: 25000,  
  following: 5,  
  posts: [  
    {  
      image: 'https://plus.unsplash  
h.com/premium_photo-1677545182067-  
26ac518ef64f?ixlib=rb-4.0.3&ixid=M  
3wxMjA3fDB8MHxzZWFyY2h8MXx8Y2F0c3x  
lbnwwfHwwfHx8MA%3D%3D&auto=format&  
fit=crop&w=800&q=60',  
      likes: 200,  
      comments: 17  
    },  
    {  
      image: 'https://images.unspl  
ash.com/photo-1592194996308-7b4387
```

views > <> instagram.ejs > html > body > h2 > ?

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial
6      <title>Instagram</title>
7    </head>
8    <body>
9      <h2>This page belongs to @<%= data %></h2>
10     <button>Follow</button>
11     <button>Message</button>
12   </body>
13 </html>
14
```

<> instagram.ejs X JS index.js

views > <> instagram.ejs > html > body > h2 > ?

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Instagram</title>
7    </head>
8    <body>
9      <h2>This page belongs to @<%= data.name %></h2>
10     <button>Follow</button>
11     <button>Message</button>
12   </body>
13 </html>
14
```



**This page belongs to @cats**

Follow

Message

```
<meta name="viewport" content="width=device-width,
<title>Instagram</title>
</head>
<body>
  <h2>This page belongs to @<%= data.name %></h2>
  <button>Follow</button>
  <button>Message</button>

  <p>
    Followers : <%= data.followers %> &nbsp;&nbsp;&nbsp;&
    data.following %>
  </p>
  <hr />

  <% for(let post of data.posts) { %>
    
    <p>
      Likes : <%= post.likes %> &nbsp;&nbsp;&nbsp;&
      post.comments %>
    </p>
  <% } %>
</body>
</html>
```



```
12 });  
13  
14 app.get("/ig/:username", (req, res) => {  
15     let { username } = req.params;  
16     const instaData = require("../data.json");  
17     const data = instaData[username];  
18     console.log(data);  
19     res.render("instagram.ejs", { data });  
20 });  
21  
22 app.get("/hello", (req, res) => {  
23     res.send("hello");  
24 });
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Instagram</title>
<style>
  img {
    height: 100px;
    width: 100px;
  }
</style>
</head>
<body>
  <h2>This page belongs to @<%= data.name %>
  <button>Follow</button>
  <button>Message</button>

  <p>
    Followers : <%= data.followers %>
  </p>
</body>
</html>
```

Followers : 75000      Following : 150

---



Likes : 3000      comments : 41



Likes : 2500      comments : 32

---

```
app.set("views", path.join(__dirname, "/views"));
```

```
app.get("/", (req, res) => {  
  res.render("home.ejs");  
});
```

```
app.get("/ig/:username", (req, res) => {  
  let { username } = req.params;  
  const instaData = require("./data.json");  
  const data = instaData[username];  
  if (data) {  
    res.render("instagram.ejs", { data });  
  } else {  
    res.render("error.ejs");  
  }  
});
```

```
app.get("/hello", (req, res) => {  
  res.send("hello");  
});
```

<> instagram.ejs

JS index.js

<> error.ejs



views > <> error.ejs > html > body > h1

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,
    <title>No Account</title>
  </head>
  <body>
    <h1>No such account</h1>
  </body>
</html>
```

abc Account

## Serving Static Files

`app.use( express.static( folder_name ) )`

```
app.use(express.static(path.join(__dirname, "public")));
```

```
const path = require('path');
```

```
const port = 8080;
```

```
app.use(express.static('public'));
```

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

```
app.set('views', path.join(__dirname, '/views'));
```

hagupta7385@gmail.com

```
app.get('/', (req, res) => {  
  res.render('home.ejs');  
});
```



JS index.js

`<> instagram.ejs` ✕

views >  instagram.ejs >  html >  head >  link

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6     <title>Instagram</title>
7     <link rel="stylesheet" href="/style.css" />
8   </head>
9   <body>
10    <h2>This page belongs to @<%= data.name %></h2>
11    <button>Follow</button>
12    <button>Message</button>
13
14    <p>
15      Followers : <%= data.followers %> &nbsp;&nbsp;&nbsp;& Followed by : <%=
16      data.following %>
17    </p>
18    <hr />
19
20    <% for(let post of data.posts) { %>
21      
22      <p>
23        Likes : <%= post.likes %> &nbsp;&nbsp;&nbsp;& Comments : <%=
24        post.comments %>
25      </p>

```



JS index.js × <> instagram.ejs

JS index.js > ...

```
1  const express = require("express");
2  const app = express();
3  const path = require("path");
4
5  const port = 8080;
6
7  app.use(express.static(path.join(__dirname, "public")));
8  app.set("view engine", "ejs");
9  app.set("views", path.join(__dirname, "/views"));
10
11 app.get("/", (req, res) => {
12   res.render("home.ejs");
13 });
14
15 app.get("/ig/:username", (req, res) => {
16   let { username } = req.params;
17   const instaData = require("../data.json");
18   const data = instaData[username];
19   if (data) {
20     res.render("instagram.ejs", { data });
21   } else {
22     res.render("error.ejs");
23   }
24 });
25
26 app.get("/hello", (req, res) => {
```

# This page belongs to @dogs

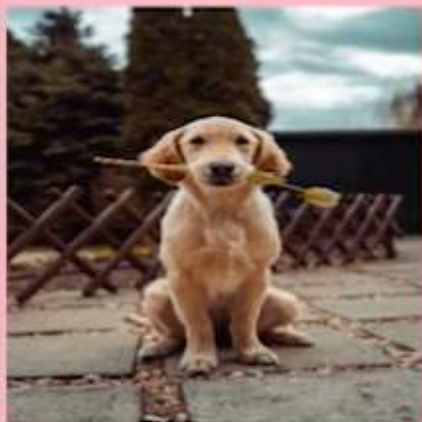
[Follow](#)[Message](#)

Followers : 75000      Following : 150

---



Likes : 3000      comments : 41



JS index.js

# style.css

<> instagram.ejs

JS index.js > ...

```
1  const express = require("express");
2  const app = express();
3  const path = require("path");
4
5  const port = 8080;
6
7  app.use(express.static(path.join(__dirname, "/public/js")));
8  app.use(express.static(path.join(__dirname, "/public/css")));
9  app.set("view engine", "ejs");
10 app.set("views", path.join(__dirname, "/views"));
11
12 app.get("/", (req, res) => {
13   res.render("home.ejs");
14 });
15
16 app.get("/ig/:username", (req, res) => {
17   let { username } = req.params;
18   const instaData = require("./data.json");
19   const data = instaData[username];
20   if (data) {
21     res.render("instagram.ejs", { data });
22   } else {
23     res.render("error.ejs");
24   }
25 }
```

# This page belongs to @cats

[Follow](#)[Message](#)

Followers : 25000      Following : 5



Likes : 200      comments : 17



JS index.js

JS app.js

# style.css

&lt;&gt; instagram.ejs

public &gt; js &gt; JS app.js &gt; btn.addEventListener("click") callback

```
1  const btns = document.querySelectorAll("button");
2
3  for(btn of btns) {
4      btn.addEventListener("click", () => {
5          console.log("button was clicked");
6      })
7  }
```





**This page belongs to @cats**

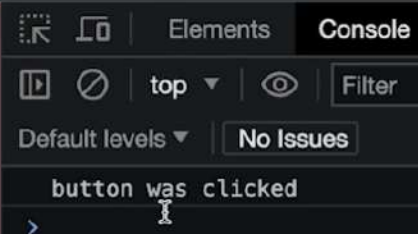
Follow

Follow Message

Followers : 25000      Following : 5



Likes : 200      comments : 17





Includes → subtemplates

```
<%- include("includes/head.ejs"); %>
```

## EXPLORER

✓ EJS DIR

&gt; node\_modules

&gt; public

✓ views

✓ includes

&lt;&gt; head.ejs

&lt;&gt; error.ejs

&lt;&gt; home.ejs

<> **instagram.ejs**

&lt;&gt; rolldice.ejs

{ } data.json

JS index.js

{ } package-lock.json

{ } package.json

&lt;&gt; instagram.ejs X


views &gt; &lt;&gt; instagram.ejs &gt; ?

```
1  <%= include("includes/head.ejs") %>
2
3  <body>
4    <h2>This page belongs to @<%= data
5    <button>Follow</button>
6    <button>Message</button>
7
8    <p>
9      Followers : <%= data.followers
10     data.following %>
11     snehagupta
12   </p>
13   <hr />
14
15   <% for(let post of data.posts) {
16     
18       Likes : <%= post.likes %> &nbsp;
19       post.comments %>
20     </p>
21     <% } %>
22   </body>
23   <script src="/app.js"></scrip
24 </html>
```

<> instagram.ejs

<> error.ejs



views > <> error.ejs >  ?

```
1 <%- include("includes/head.ejs") %>
2   <body>
3     <h1>No such account</h1>
4   </body>
5 </html>
6
```

**No such account**




<> instagram.ejs

<> error.ejs






<> rolldice.ejs

<> footer.ejs ×

views > includes > <> footer.ejs >  div

```
1 <div style="background-color:  lightblue">
2   Contact us at : hello@abc<br />
3   Phone no. - +91 xxxxxxxxxx
4 </div>
5
```

<> rolldice.ejs

```
views > <> instagram.ejs >  ? >  body >  ? >  ? >  ?
```

[illegible]