

EJS# Templating →

EJS (Embedded Javascript templates)

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

Using EJS

```

const express = require("express");
const app = express();
const path = require("path");

const port = 8080;

app.set("view engine", "ejs");
app.set("views", path.join(__dirname, "views"));

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

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

app.listen(port, () => {
  console.log(`listening on port ${port}`);
});
  
```

Interpolation Syntax

Interpolation refers to embedding expressions into marked up text.

~~for short~~ ~~you can't do it with spaces or tabs~~
 <head> ~~you can't mix punctuation with~~
 — — .

</head>

<body>

<h1> — — </h1>

<h3> $\{ \text{age} = 1 + 2 * \% \}$ </h3>

→ ③ → Output

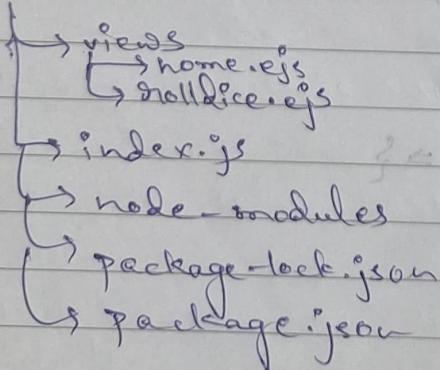
<p> — — </p>

</body>

</html>

Passing Data to EJS

EJSDir



→ file structure

index.js

const express = require ("express");

```
const app = express();
```

```
const port = 8080;
```

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

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

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

```
app.listen(port, () => {
  console.log(`Listening on port ${port}`);
});
```

rollDice.ejs

```
<html>
```

```
  <head> -- </head>
```

```
  <body>
```

```
    <h1> Dice gave value: <% =diceVal %> </h1>
```

```
  </body>
```

```
</html>
```

(#) Instagram EJS :-

Basic template for Instagram page based on following route:

/ig/:username

In last index.js code, add:

```
app.get("/ig/:username", (req, res) => {
  let {username} = req.params;
  console.log(username);
}); res.render("instagram.ejs", {username});
```

Inside views folder, create `instagram.ejs`:

```
instagram.ejs:
<html>
  <head> ... </head>
  <body>
    <h2> This page belongs to <% =username %>
    <button> Follow </button>
    <button> Message </button>
  </body>
</html>
```

conditional Statements in EJS

- Adding conditions inside EJS.

In last code, keep `index.js` as it is:

In ~~Instagram~~ rollDice.ejs:

```
<html>
  <head> ... </head>
  <body>
    <h1> Dice gave value : <% =diceVal %> </h1>
```

```

<% if (diceVal == 6) { %>       spot: incomplete (1)
    <h2> Nice! Roll dice again. </h2>
<% } %>                         spot: incomplete
</body>                           check
</html>                           check

```

Loops:

Eg: <html>

<head> -> </head>

<body>

<h3> Accounts that follow you: </h3>

<% for (let name of followers) { %>

 <% = name %>

<% } %>

</body>

</html>

<?> (imported to log file) out :>

In <index.ejs> <?> =

app.get("/og/:username", (req, res) => {

const followers = ["adam", "bob", "charles", "steve"];

let {username} = req.params;

res.render("instagram.ejs", {username, followers});

out[0] =

out[1] =

out[2] =

out[3] =

④ Instagram Page with EJS

→ `instagram.ejs`

`<html>`

`<head>`

`<title> --> </title>`

`<style>`

`img {`

`height: 100px;`

`width: 100px;`

`}`

`</style>`

`</head>`

`<body>`

`<h1> This page belongs to @ <% = data.name %> </h1>`

`<p> Followers: <% = data.followers %> &nbsp &nbsp &nbsp`

~~`<p> following: <% = data.following %>`~~

`Following: <% = data.following %>`

`</p>`

``

`<% for(let post of data.posts) { %>`

``

`</p>`

`Likes: <% = post.likes %> &nbsp &nbsp &nbsp &nbsp Comments: <% = post.comments %>`

`; </p>`

`<% } %>`

`<button> Follow </button>`

`<button> Message </button>`

`</body>`

`</html>`

→ error.ejs

<html>

<head>...</head>

<body>

No such acc exists!

</body>

</html>

→ index.js

```
const express = require ("express");
```

```
const app = express();
```

```
const port = 8080;
```

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

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

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

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

```
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.listen(port, () => {
    console.log(`Listening on port ${port}`);
})

```

Serving static files:

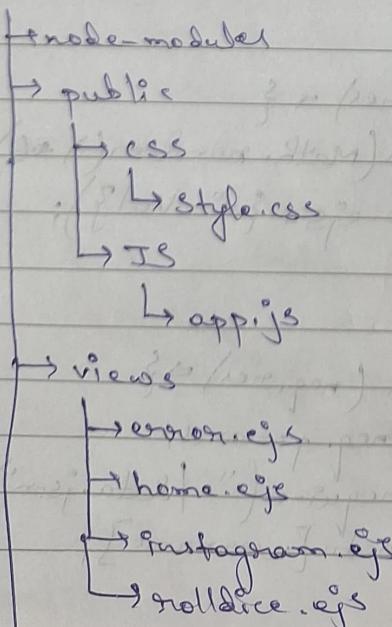
```

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

```

File Structure:

ES6 dir



→ data.json
 → index.js
 → package-lock.json
 → package.json

Keep all files as it is, just make following changes

index.js :

NOTE: Add these to existing code

```
app.use(express.static("public/css"));
app.use(express.static ("public/js"));
```

style.css :

```
body {
  background-color: red;
}
```

```
img {
  height: 100px;
  width: 100px;
}
```

app.js :

```
const btns = document.querySelectorAll("button");
for (btn of btns) {
  btn.addEventListener("click", () => {
    console.log ("button was clicked");
});
```

Includes

Create a new folder `includes` inside `views` folder

↳ `views`
↳ `includes`
↳ `head.ejs`

`head.ejs`

`<html>`

`<head>` Same as earlier `</head>`
`</html>`

`instagram.ejs`

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

`<body>`

Same as earlier

`</body>`

`</html>`

NOTE: We can similarly use `includes` for any section of code which we want to include at multiple places like header, footer, etc.