



12. Dynamic UI using EJS

1. Need of Dynamic UI
2. Sharing using Global Variables
3. What is EJS
4. Installing EJS
5. Using EJS Templates
6. Working with Partials



12.1 Need of Dynamic UI



- **Personalized Content:** Tailors responses based on user profiles, preferences, or behaviors to enhance user experience.
- **Dynamic Data Delivery:** Provides real-time information that updates with each request, such as live scores or stock prices.
- **Security and Access Control:** Delivers different content based on user authentication and authorization levels.
- **Localization and Internationalization:** Adjusts responses to accommodate different languages, cultures, or regional settings.
- **API Versatility:** Supports multiple client types (web, mobile, IoT) by providing appropriate data formats and structures.



12.2 Sharing using Global Variables

```
const homes = [];
```

```
// POST path
```

```
router.post('/add-home', (req, res, next) => {  
  homes.push({houseName: req.body.houseName});  
  res.redirect('/');  
});
```

Variables are shared across
users

```
exports.hostRouter = router;  
exports.homes = homes;
```

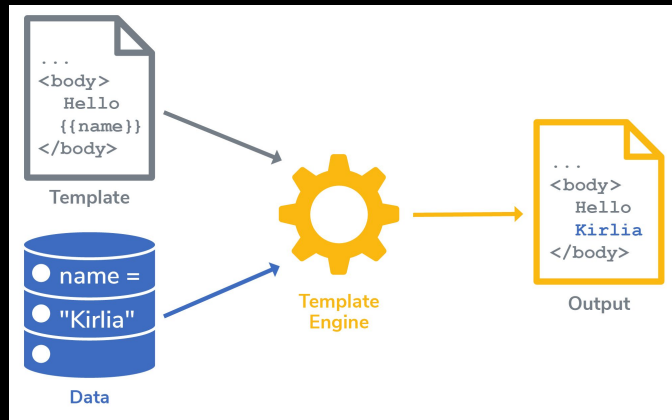
```
const {homes} = require('../routes/host');
```

```
userRouter.get('/', (req, res, next) => {  
  console.log('homes', homes);  
  res.sendFile(path.join(rootDir, 'views', 'user.html'));  
});
```

12.3 What is EJS



- **HTML with JS:** EJS lets you embed JavaScript code within HTML.
- **Simple Syntax:** Uses `<% %>` for control flow and `<%= %>` for output.
- **Easy to Learn:** Familiar to those who know HTML and JavaScript.
- **Template Reuse:** Supports **partials** for reusing code snippets.
- **Flexible Logic:** Allows full JavaScript expressions in templates.



12.4 Installing EJS



```
prashantjain@Prashants-Mac-mini air-bnb % npm install --save ejs
```

```
added 6 packages, and audited 232 packages in 879ms
```

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

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

```
app.set('views', 'views');
```

12.4 Installing EJS



app.set(name, value)

Assigns setting `name` to `value`. You may store any value that you want, but certain names can be used to configure the behavior of the server. These special names are listed in the [app settings table](#).

Calling `app.set('foo', true)` for a Boolean property is the same as calling `app.enable('foo')`. Similarly, calling `app.set('foo', false)` for a Boolean property is the same as calling `app.disable('foo')`.

Retrieve the value of a setting with `app.get()`.

```
app.set('title', 'My Site')
app.get('title') // "My Site"
```

views	String or Array	A directory or an array of directories for the application's views. If an array, the views are looked up in the order they occur in the array.	<code>process.cwd() + '/views'</code>
view cache	Boolean	Enables view template compilation caching. NOTE: Sub-apps will not inherit the value of this setting in production (when <code>NODE_ENV</code> is "production").	<code>true</code> in production, otherwise undefined.
view engine	String	The default engine extension to use when omitted. NOTE: Sub-apps will inherit the value of this setting.	N/A (undefined)

12.5 Using EJS Templates



views

404.html

add-home.html

input.css

user.ejs

```
// Local Modules
```




```
const {homes} = require('../routes/host');
```

```
userRouter.get('/', (req, res, next) => {  
  res.render('user', { homes: homes });  
});
```

```
<main class="container mx-auto bg-white shadow-lg rounded-lg p-8 mt-10 max-w-xl">  
  <h2 class="text-5xl text-gray-800 font-bold text-center mb-6">  
    Available Homes  
  </h2>  
  <ul class="list-disc list-inside space-y-2">  
    <% homes.forEach(function(home) { %>  
      <li class="text-gray-700 hover:text-red-500 transition-colors duration-300 text-2xl">  
        <%= home.houseName %>  
      </li>  
    <% }); %>  
  </ul>  
</main>
```

12.6 Working with Partials



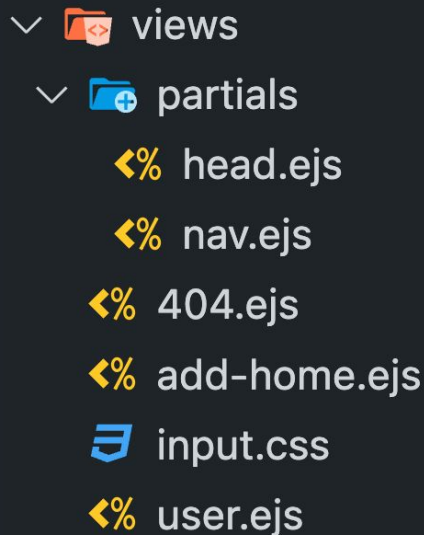
- ✓  views
 - ✓  partials
 - ⌞% head.ejs
 - ⌞% nav.ejs
 - ⌞% 404.ejs
 - ⌞% add-home.ejs
 -  input.css
 - ⌞% user.ejs

```
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><%= title %></title>
7     <link rel="stylesheet" href="/output.css" />
8
```


12.6 Working with Partials



```
1 <header class="bg-red-500 text-white p-4 shadow-md">
2   <nav class="container mx-auto flex justify-between items-center">
3     <ul class="flex space-x-4">
4       <li>
5         <a
6           href="/"
7           class="hover:bg-red-400 py-2 px-4 rounded transition duration-300"
8         >Go to Home</a>
9       >
10    </li>
11    <li>
12      <a
13        href="/host/add-home"
14        class="hover:bg-red-400 py-2 px-4 rounded transition duration-300"
15      >Add Home</a>
16    >
17  </li>
18 </ul>
19 </nav>
20 </header>
```



12.6 Working with Partials



air-bnb > views > 404.ejs > ?

```
<%- include('partials/head') %>
```

```
</head>
```

```
<body class="bg-gray-100 font-sans">
```

```
<%- include('partials/nav') %>
```

```
<main class="container mx-auto mt-20 text-center">
```

```
<h2 class="text-6xl font-bold text-red-500 mb-4">404</h2>
```

```
<p class="text-2xl text-gray-700 mb-8">Oops! Page Not Found</p>
```

```
<a href="/" class="bg-red-500 text-white py-2 px-6 rounded-lg  
hover:bg-red-600 transition duration-300">Go Back Home</a>
```

```
</main>
```

```
</body>
```

```
</html>
```

```
app.use((req, res, next) => {
```

```
  res.status(404).render('404', {title: 'Page Not Found'});
```

```
});
```

12.6 Working with Partials



```
userRouter.get('/', (req, res, next) => {  
  res.render('user', { homes: homes, title: 'Available Homes' });  
});
```

```
<%- include('partials/head') %>  
  </head>  
  <body class="■bg-gray-100 font-sans">  
    <%- include('partials/nav') %>  
    <main class="container mx-auto ■bg-white shadow-lg rounded-lg p-8 mt-10  
max-w-xl">  
      <h2 class="text-5xl □text-gray-800 font-bold text-center mb-6">  
        Available Homes  
      </h2>
```



Practise Set

Reuse the app from the **last assignment**

1. **Add** more fields in the add home page like price per night, location, rating, photo.
2. **Design** the home card to show all of this information
3. **Make** the selected tab active on top.





Revision

1. Need of Dynamic UI
2. Sharing using Global Variables
3. What is EJS
4. Installing EJS
5. Using EJS Templates
6. Working with Partial

