

MAKÜ|BTF

BUCAK TEKNOLOJİ FAKÜLTESİ

Web Tasarımı ve Programlama

Ders 7:EJS

Dr. Öğr. Üyesi Elvan DUMAN

eduman@mehmetakif.edu.tr
Yazılım Mühendisliği Bölümü

1. EJS'nin Tanımı ve Amacı
2. Kurulum ve Temel Çalışma Mantığı
3. EJS Sözdizimi ve Veri Kullanımı
4. Koşullar ve Döngülerle
5. Dinamik İçerik
6. HTML İçerik Basma ve Partial Kullanımı
7. Include Yapısı ve Sayfa Düzeni

EJS (Embedded JavaScript Templates), Node.js tabanlı uygulamalarda dinamik HTML sayfaları oluşturmak için kullanılan bir template engine'dir.

HTML içinde JavaScript kodlarını gömülü şekilde yazmamıza olanak tanır.

.ejs uzantılı dosyalarla çalışır.

Özellikle **Express.js** ile birlikte kullanılır.

Amaç:

Sunucu tarafında dinamik içerik oluşturmak (örneğin kullanıcı adı, ürün listesi, vs.)

HTML dosyalarını tekrar kullanılabilir hale getirir.

JavaScript ile verileri doğrudan sayfaya yerleştirmeyi sağlar.

`<%= %>` gibi özel sözdizimiyle değişkenleri doğrudan HTML içinde gösterebiliriz.

Statik HTML yerine, **veriye dayalı dinamik sayfalar** üretir.

Örnek:

`<h1>Welcome <%= username %></h1>`

Sunucudan gelen username değişkeni HTML içinde görüntülenir.

Kurulum
npm install ejs

Express.js projesine dahil etme

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

Views klasörü oluşturma

EJS dosyaları varsayılan olarak views/ klasöründe tutulur.
Örnek: views/index.ejs

app.js içinde bir route tanımlanır.

Route bir .ejs dosyasını render eder.

Sunucudan gelen veriler EJS sayfasına gönderilir.

Örnek:

```
app.get('/', (req, res) => {  
  res.render('index', { name: 'Elvan' });  
});
```

index.ejs:

```
<h1>Hello, <%= name %>!</h1>
```

- `<%= %>` → Değişkenin değerini yazdırır (escape ederek)
- `<%- %>` → Değişkenin değerini HTML olarak işler (escape etmez)
- `<% %>` → JavaScript kodunu çalıştırır ama ekrana yazdırmaz

Örnek:

```
<h1><%= title %></h1>
<% if(showList) { %>
  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
  </ul>
<% } %>
```

EJS şablonları HTML içinde JavaScript ifadeleri çalıştırarak dinamik içerik oluşturur.

Sunucudan gönderilen veriler, EJS dosyasında doğrudan kullanılabilir.

Js dosyası

```
app.get('/user', (req, res) => {  
  res.render('user', { username: 'Mehmet', age: 35 });  
});
```

user.ejs

```
<p>Name: <%= username %></p>  
<p>Age: <%= age %></p>
```

- Sunucudan res.render() ile gönderilen nesne anahtarları, .ejs sayfasında değişken olarak erişilir.

EJS, koşul ifadelerini normal JavaScript gibi çalıştırır.

```
<% if (loggedIn) { %>  
  <h2>Welcome back, <%= username %>!</h2>  
<% } else { %>  
  <h2>Please log in to continue.</h2>  
<% } %>
```

if, else if, else blokları HTML içinde koşullu içerik üretmek için kullanılır. Her koşul açıldığında <% ve kapandığında %> yazılır.

Diziler üzerinde döngü kurarak dinamik liste oluşturabiliriz.

Js dosyası

```
app.get('/products', (req, res) => {  
  res.render('products', { items: ['Apple', 'Banana', 'Orange'] });  
});
```

products.ejs

```
<ul>  
  <% items.forEach(function(item){ %>  
    <li><%= item %></li>  
  <% }) %>  
</ul>
```

Bazı veriler HTML etiketleri içerebilir.
Bu durumda <%- %> etiketi kullanılır.

<%- htmlContent %>

<%- %> kullanıldığında değişken içindeki HTML kodları (örneğin text) yorumlanır.

EJS’de **partial** dosyalar, sayfanın tekrar eden bölümlerini (örneğin başlık, menü, footer) ayrı dosyalara bölmek için kullanılır.

Amaç:

- Kod tekrarını azaltmak
- Sayfa düzenini kolayca güncellemek
- Tüm sayfalarda ortak yapıyı korumak

EJS'de bir partial dosyayı dahil etmek için:

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

include() fonksiyonu, belirtilen .ejs dosyasını o noktaya gömer.
partials/header → views/partials/header.ejs yolunu ifade eder.
<%- %> kullanılır, çünkü HTML çıktısı direkt gömülmelidir.

header.ejs

```
<header>  
  <h1>My Website</h1>  
  <nav>  
    <a href="/">Home</a>  
    <a href="/about">About</a>  
  </nav>  
</header>
```

Footer.ejs

```
<footer>  
  <p>&copy; 2025 My Website</p>  
</footer>
```

Ders 7 Sunum Sonu

İletişim : eduman@mehmetakif.edu.tr

