

MAKÜ|BTF

BUCAK TEKNOLOJİ FAKÜLTESİ

Web Tasarımı ve Programlama

Ders 10: Çerez ve Oturum

Dr. Öğr. Üyesi Elvan DUMAN

eduman@mehmetakif.edu.tr
Yazılım Mühendisliği Bölümü

Web Uygulamalarında Veri Saklama

Şimdiye kadar dosya ve veritabanı üzerinde veri saklamayı öğrendik.

Ancak her veri veritabanına veya dosya sistemine yazılmak zorunda değildir.

Bazı veriler

- Kısa süreli
- Kullanıcıya özel
- Geçici

Olabilir. Bu tür veriler için bellek veya tarayıcı tabanlı çözümler kullanılır.

Kullanıcıya Özel Durum (State) Yönetimi

- Web uygulamaları **stateless** yapıdadır (Her HTTP isteği bağımsızdır)
- Ancak gerçek uygulamalarda:
 - Kullanıcı girişi (login)
 - Sepet bilgisi
 - Oturum durumu
 - Kullanıcı tercihleri gibi bilgileri **hatırlamamız gerekir**

Bu noktada iki temel teknoloji devreye girer:

- **Cookie**
- **Session**

Gerçek Hayat Web Uygulamaları İçin

- Kimlik doğrulama (Authentication)
- Yetkilendirme (Authorization)
- Kullanıcı deneyimi (UX)
- Güvenlik (Security)

gibi konular **doğrudan session ve cookie yönetimine bağlıdır.**

Bu yüzden:

- Session ve cookie'leri anlamadan modern web uygulaması geliştirmek mümkün değildir.

Cookie, kullanıcıya ait küçük bilgilerin tarayıcıda (client-side) saklanmasını sağlayan bir yapıdır.

Bu bilgiler:

- Kullanıcının **giriş yapıp yapmadığı**
- Bazı **kısa süreli durum bilgileri** olabilir.

Cookie'ler:

- Sunucu tarafından oluşturulur,
- Tarayıcı tarafından saklanır,
- Sonraki isteklerde otomatik olarak sunucuya gönderilir.

Cookie, kullanıcıya ait küçük bilgilerin tarayıcıda (client-side) saklanmasını sağlayan bir yapıdır.

Bu bilgiler:

- Bazı **kısa süreli durum bilgileri** olabilir.

Cookie'ler:

- Sunucu tarafından oluşturulur,
- Tarayıcı tarafından saklanır,
- Sonraki isteklerde otomatik olarak sunucuya gönderilir.

Cookie sadece **key-value** bilgisinden ibaret değildir.

Ayarlanabilen bazı özellikleri bulunur.

- **expires** Belirli bir tarihte cookie sona erer.
- **max-age** Kaç saniye geçerli olacağını belirler
- **secure** Cookie sadece **HTTPS** üzerinden gönderilir.
- **httpOnly** Cookie'ye **tarayıcı JavaScript'i erişemez**. XSS (Cross-Site Scripting) saldırılarına karşı koruma sağlar.

Eğer bu değerler ayarlanmazsa: Cookie, **tarayıcı kapatılınca silinir**.

Cookie ile:

- Kullanıcının ne kadar süre takip edileceği,
- Ne kadar süre oturumda kalacağı kontrol edilebilir (ör. bankacılık uygulamalarındaki timeout)..

Kullanıcı:

- Developer Tools üzerinden kendi cookie'sini görebilir, değiştirebilir veya silebilir.

Bu yüzden: “Giriş yaptım” gibi kritik bilgiler. Doğrudan cookie içinde tutulmaz.

Gerçek uygulamalarda:

- Cookie'ler genellikle **doğrudan değil, Authentication / Session paketleri** tarafından yönetilir.

Session, kullanıcıya ait bilgilerin sunucu tarafında (server-side) saklanmasını sağlayan bir yapıdır.

Kullanıcı giriş yaptığında, “Giriş yaptı mı?”, “hangi kullanıcı?” gibi bilgiler **tarayıcıda değil, backend’de** tutulur.

Bu bilgiler:

- Request içinde tutulmaz (çalışmaz),
- Express uygulamasındaki bir değişkende tutulma (tüm kullanıcılarla paylaşılacağından güvenli değil).

Amaç:

- **Aynı kullanıcıya ait tüm isteklerde** bilgiyi korumak,
- Diğer kullanıcıların bu bilgilere erişmesini engellemektir.

Session = kullanıcıya özel, sunucu taraflı durum bilgisi

Session: Sunucuda tutulur (memory veya database).

Ancak, Sunucu, gelen isteğin **hangi session'a ait olduğunu bilmek zorundadır.**

Bunun için tarayıcıya bir **cookie gönderilir.** Bu cookie Session'ın **ID'sini** içerir. (Gerçekte bu ID **hash'lenmiş / imzalanmış** haldedir).

Tarayıcı, bu çerezi her istekte gönderir. Sunucu çerezden oturumu bulur. Kullanıcıya ait gizli bilgileri oturumdan okur.

Cookie **client-side**, Session **server-side** tutulur.

Cookie, session'a giden **güvenli anahtardır.**

Session kullanmak için ek bir third-party paket gerekir: **express-session**
Session middleware, uygulama başlarken **app.js** içinde tanımlanır. Böylece session, tüm gelen request'ler için aktif olur.

```
app.use(  
  session({secret: 'ben bir ceviz ağacıym', resave:false, saveUninitialized:false,  
  store:store})  
);
```

secret, session ID'nin cookie içinde imzalanması için kullanılır ve gerçek projelerde uzun ve gizli bir değer olmalıdır

resave: false, session'ın her istekte yeniden kaydedilmesini engeller, yalnızca değişiklik varsa kaydedilmesini sağlar

saveUninitialized: false, hiç kullanılmayan session'ların gereksiz yere saklanmasını önler.

İsteğe bağlı olarak session cookie için **maxAge** veya **expires** ayarlanabilir.

Bu ayarlardan sonra session middleware hazırdır ve uygulama içinde kullanılabilir

- Session middleware eklendikten sonra sunucu yeniden başlatılır.
- Session nesnesi artık her request'te req.session üzerinden erişilebilir
- Login işleminde Cookie set etmek yerine req.session.isLoggedIn = true gibi bir değer atanır.
- Bu değer Sunucu tarafında saklanır istekler arasında korunur. Sayfa yenilense bile Session içindeki bilgi kaybolmaz

Session, Cookie ve Kullanıcı Ayrımı

- Login sonrası tarayıcıda **connect.sid** adlı bir cookie oluşur
- Bu cookie: Session ID'yi içerir. Karmaşık ve imzalanmış bir değerdir.
- Tarayıcı bu cookie'yi her request'te gönderir. Sunucu çerez üzerinden doğru session'ı bulur.
- Aynı tarayıcıda farklı sayfalara gidilse bile session korunur

- Varsayılan olarak session'lar **hafızada (memory)** tutulur. Geliştirme ortamı için bu yeterlidir.
- Ürün ortamında, çok sayıda kullanıcı olduğunda sessionların hafızada tutulması belleğin hızla dolmasına ve sunucunun çökmesine neden olabilir. Güvenlik açısından da ideal değildir
- Bu yüzden sessionlar **kalıcı bir disk üzerinde** tutulmalıdır
- express-session farklı **session store**'ları destekler. MongoDB kullanmak için: **connect-mongodb-session** paketi kullanılır. Bu paket oturum bilgilerini veritabanına kaydeder.
- MongoDB bağlantı adresi (URI) ile yapılandırılır.
- Session middleware'e store eklendiğinde MongoDB'de tutulur.

```
const MONGODB_URI='buraya gelecek';
const store= new MongoDBStore( {
  uri:MONGODB_URI,
  collection: 'sessions'
});
```

Ders 10 Sunum Sonu

İletişim : eduman@mehmetakif.edu.tr

