

# ISAPI UYGULAMALARINDA, SUNUCU SİSTEMLERDE OTURUMLAR OLUŞTURARAK KULLANICI ERİŞİM KONTROLU VE BİR UYGULAMASI

Erdem Uçar<sup>1</sup>, Erdiñç Uzun<sup>2</sup>, Veysel Karlı<sup>3</sup>

<sup>1</sup> Yrd.Doç.Dr., Trakya Üniversitesi, Mühendislik Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümü  
[erdemu@trakya.edu.tr](mailto:erdemu@trakya.edu.tr)

<sup>2</sup> Ar.Gör., Trakya Üniversitesi, Mühendislik Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümü  
[erdinc@trakya.edu.tr](mailto:erdinc@trakya.edu.tr)

<sup>3</sup> Öğr., Trakya Üniversitesi, Mühendislik Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümü  
[veyselkarli@trakya.edu.tr](mailto:veyselkarli@trakya.edu.tr)

Anahtar Kelimeler: ISAPI, Session, Cookie Oturum Konusu: internet

## 1. GİRİŞ

İnternet sunucu programları ilk olarak Unix işletim sistemi kurulu bilgisayarlar için geliştirildi. Bu yüzden, bu programlar Microsoft, IIS (Internet Information Server)'i tanıtmadan önce standartlarda uzun süre yer aldı. HTTP'nin esas parçası olan CGI (Common Gateway Interface) standardı, sunucu üzerinde çalışan ayrı çalışabilir programlarla veya kod parçalarıyla (script) etkileşmek için tarayıcı(browser) programlarına bir yol olarak geliştirildi. HTTP/CGI mimarisi değişmeksizin, Microsoft bir sunucu DLL'nin yüklenmesi ve çalıştırması için browsera izin veren IIS'i tasarladı. DLL'ler IIS işleminin parçasıdır ve bu yüzden ayrı çalışabilir programları yüklemek için gerekli olan kod parçalarından daha hızlıdır. Ayrıca C++'da bir ISAPI DLL yazmak, sunucular için standart ağ kod parçaları dili olan PERL'de bir kod yazmaktan çok daha kolaydır.

ISAPI sunucu uzantıları, internet sunucuları için CGI uygulamalarının kullanımına bir alternatif sağlar. CGI uygulamalarından farkı, ISA(Internet Server Application)'lar HTTP sunucusu gibi aynı adres bölgesinde çalışır ve HTTP sunucusunda mevcut tüm kaynaklara erişime sahiptir. ISA'lar CGI uygulamalarından daha az masraflıdır. Çünkü onlar ilave işlemlerin oluşturulmasına ihtiyaç duymazlar ve işlem sınırları içerisinde vakit alan iletişimleri icra etmezler. Eğer bellek, diğer işlem tarafından ihtiyaç duyulursa, her iki uzantı(Extension) ve filtre(Filter) DLL'leri bellekten boşaltılabilir. Bir ISA, bir tarayıcı uygulama tarafından çağrılır. Bir internet istemcisi, bir CGI uygulamasını çağırdığı aynı yolla HTTP sunucusu içinden bir ISA'yı da çağırır. Örneğin, bir sunucu bir CGI uygulamasını aşağıdaki gibi çağırabilir.

<http://sample/example.exe?Param1&Param2>

Aynı fonksiyonu icra eden bir ISA'yı da aşağıdaki gibi çağıracaktır.

<http://sample/example.dll?Param1&Param2>

ISAPI, bir DLL içerisinde çoklu komutlara izin verir. CGI ise, ayrı bir isme ve her görev için ayrı bir çalışabilir dosya ile URL eşlemesi yapmaya ihtiyaç duyar. Her yeni CGI

isteğinde yeni bir işlem başlatılır. Her farklı istek, her istekte belleğe yüklenen ve bellekten boşaltılan kendi çalışabilir dosyası içerisinde icra edilir. Bundan dolayı ISA'dan daha masraflıdır.

## 2. OTURUM(SESSION)

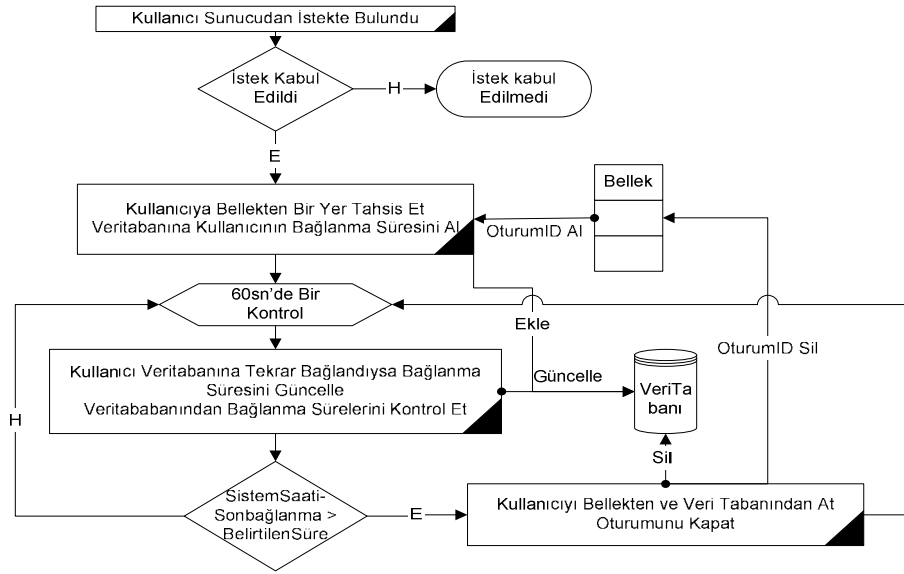
Başarılı bir Web uygulaması geliştirme kurallarından biri, o uygulama içerisinde sayfadan sayfaya geçiş yapan kullanıcı ziyaretinin veya oturumunun, gidişatı boyunca kullanıcı bilgisini muhafaza etmektir. Web sunucusunun, bir sayfadan gelen her HTTP isteğinde bağımsız bir istekmiş gibi davranması anlamında HTTP belirsiz bir protokoldür. Sunucu, eğer istekler geçerli isteklerden önceki saniyelerde meydana geldiyse önceki isteklerin hiçbir bilgisini tutamaz. Önceki istekleri hatırlamadaki bu yeteneksizlik, çeşitli katalog sayfaları arasında geçiş yaparken, bir kullanıcının seçmiş olduğu katalog ürünlerini izlemeye ihtiyaç duyan on-line uygulamaları yazmadaki zorluğu ifade eder. İşte bu zorluğu ortadan kaldırmak için oturum(session) geliştirilmelidir. Kullanıcı için gerekli özel bilgileri saklamak için bu oturumlar kullanılabilir. Oturumda saklanan değişkenler, kullanıcı uygulama içerisinde sayfalar arası geçişler yaptığı zaman bellekten boşaltılmazlar. Böylece sayfalar arasındaki geçişte bilgiler kaybolmaz.

Oturumlarla ilişkili olarak çerezler(cookie) kullanılmaktadır. Çerezler, kullanıcıyı teşhis etmek için Web sunucusu tarafından bu kullanıcının tarayıcısına yerleştirilen bir işaretir. Bir daha ki sefer aynı tarayıcı bir sayfa isteğinde bulunduğu zaman, tarayıcı Web sunucusundan aldığı çerezi de gönderir. Her bir tarayıcı için kullanılan çerezler farklıdır. Fakat, aynı tarayıcı tarafından bir uygulama içerisindeki sayfalar arasında dolaşılsa dahi aynı çerezler işlem görür. Böylece, ilk istekte bulunan bir tarayıcıya, sunucuda üretilen tekrarsız bir *id* numarası göndererek ve diğer isteklerinde bu *id* numarasını alıp kontrol ederek, isteğin hangi tarayıcıdan geldiğini kolayca anlayabiliriz.

## 3. UYGULAMA

ISAPI uzantı DLL'lerini kullanarak bir Web uygulaması geliştirmek istediğimizde *asp*, *php* ve *jsp* gibi Web tabanlı programlama dillerinde hazır olarak gelen ve kod içerisinde kolayca kullanılabilen oturum nesnesi bulunmadığından, kendi uygulamamız için bir nesne geliştirmeye karar verdik. Bu nesne, her kullanıcının kendi özel sayfalarını görebilmesi için bir istek yaptığında, sunucu bellek bölgesinde bir yer tahsis eder. Bellek bölgesinde tahsis edilen yerin adresi her seferinde farklı olacağından oturum *id*'si olarak kullanılır. Daha sonra bu oturum *id*'si istekte bulunan tarayıcının çerezine gönderilir. Artık bu kullanıcı, sayfalar arasında geçişler yaptığında tarayıcının çerezi içerisindeki oturum *id*'si her seferinde sunucuya gönderileceğinden ve gelen bu oturum *id*'si bellek bölgesinde kontrol edileceğinden diğer kullanıcılardan ayrılmış olur. Kullanıcı, kendi özel sayfasından çıkmak için işlem yaptığında sunucu bunu anlayıp o kullanıcıya ait bellek bölgesini kapatır. Böylece, bu kullanıcının oturumu kapatılmış olur. Fakat, kullanıcı belli bir süre bir istekte bulunmadığında veya gerekli çıkış işlemini yapmadan sayfayı kapattığında, sunucu bu kullanıcının oturumunu kapatamayacaktır. Bu durum, sunucu bellek bölgesinde gereksiz bilgilerin birikmesine yol açacaktır. Bu yüzden, her kullanıcının oturumuna belli bir zaman

verilir ve bu zaman süresi içerisinde herhangi bir istekte bulunulmadığında o kullanıcıya ait bellek bölgesi kapatılmış olur. Bunu yapabilmek için, her kullanıcının oturum *id*'si ve en son istekte bulunduğu zaman veritabanında tutulur. Veritabanında tutulan bu kayıtlar her 60 sn'de bir kontrol edilerek oturum süresi dolmuş olanlar hem bellekten hem de veritabanından silinerek oturumları kapatılmış olur. (Şekil1)



Şekil1: Kullanıcı Oturum İşlemleri Algoritması

#### 4. SONUÇ

Geliştirdiğimiz bu uygulama sayesinde, ISAPI uygulamalarını kullanacak olan yazılımcıların kendilerine özel amaçlarını gerçekleştirebilecek bir kullanıcı oturumu oluşturma imkanı sağlanacaktır. Bu sayede, DLL'lerin hızlı çalışmasından da faydalanarak geniş kapsamlı bir kullanıcı oturum yönetimi sağlanmış olur. Ayrıca, bu uygulama bazı eklentilerle Web sunucu uygulaması olarak kullanılabilir. <sup>1,2</sup>

<sup>1</sup> MSDN Library Visual Studio 6.0

<sup>2</sup> Programming Microsoft Visual C++ by David J. Kruglinski 1998