



Burak KUTBAY

<http://burakkutbay.com>

<http://blog.burakkutbay.com>

JAVA SERVER PAGES

Blog'da yazılmış olan JSP Derslerinden oluşmaktadır

ücretsiz e-kitap

JAVA SERVER PAGES

TEKNOLOJİSİ

<http://blog.burakutbay.com>

hayatı, yaşanır kılan

iki kadın..

ve

bir adama...

Burak KUTBAY

1.Yayın Ocak 2014

info@burakkutbay.com

<http://burakkutbay.com>

<http://blog.burakkutbay.com>

Kapak ve Sayfa Düzeni : Burak KUTBAY

Bu kitabın bütün hakları saklıdır. Kaynak gösterilerek kullanılabilir.

Burak KUTBAY Kimdir?

Merhaba ben Burak KUTBAY, benim kim olduğum hakkında fikir edinmek isterseniz (<http://blog.burakutbay.com/hakkinda/>) linkinden ulaşabilirsiniz.

Kısaca kendimden bahsetmek gerekirse, Lise yaşantımdan bu tarafa kendimi yazılıma adanmış programlama üzerine kendimi geliştirmeyi seven, yazılımı bir hobi olarak görmekteyim. Yazılımla tanışmam Pascal ile oldu ardından Delphi ile büyük bir devrim yaşadım. Daha iyisini görmek için 3 sene C Sharp Desktop Application üstünde çalıştıktan sonra Java'nın eşsiz dünyasındaki Web çözümü olan Java Server Pages ile tanıştım. Java Web Teknolojileri üzerine kendimi geliştirmeye, projeler yapmaya devam etmekteyim.

ÖNSÖZ

Şu an okumakta olduğunuz belkide olacağınız kitap Java Server Pages Teknolojisi hakkında bilgiler vermektedir.

Bu bilgiler kendi blogumda yazmış olduğum (<http://blog.burakutbay.com>) Java Server Pages Dersleri'nin derlenip toparlanmış halidir.

Kendimi JSP üzerine geliştirirken öğrendiğim/tecrübe ettiğim konuları bloguma aktarırken Şu an ki haliyle bile Java Server Pages Teknolojisi'ni kavrayıp bir proje yapabileceğinize bile inanmaktayım. En temel bilgilerden tutun daha ileri seviye olabilecek bilgileri elimden geldiğince detaylıca anlatmaya çalıştım.Bu kitapta bulduğunuz/bulabileceğiniz yazım hataları için sizden şimdiden özür diliyorum.

Bu kitap ücretsizdir gönlünüzce kullanabilirsiniz. Kendi sitem haricinde bir dağıtım yapılacak ise kaynak belirtmenizi rica ederim.

Eksik bulduğunuz yerleri, istediklerinizi eleştiri ve şikayetleri aklınıza gelen her türlü iletiry benimle lüften (<http://blog.burakutbay.com/iletisim/>) adresinden paylaşın.

Blogum'da Java Server Pages yazıları eklediğimde bu PDF'de anlık olarak güncellenecektir. Bu güncellemelerden haberdar olmak için sitemi ziyaret edebilirsiniz.

Faydalı olması dileği ile...

İÇİNDEKİLER

<u>JAVA SERVER PAGES NEDİR?.....</u>	<u>1</u>
<u>APACHE TOMCAT KURULUMU VE KULLANIMI</u>	<u>1</u>
<u>MYSQL SERVER MYSQL WORKBENCH KURULUMU</u>	<u>3</u>
<u>JSP TEMELLERİ VE MERHABA DÜNYA</u>	<u>4</u>
<u>JSP ETİKETLERİ – JSP TAG ‘LAR</u>	<u>5</u>
<u>JAVA STANDART TAG LİBRARY NEDİR? JSTL KULLANIMI</u>	<u>6</u>
<u>JSP JSTL < C : CATCH > < C : İF ></u>	<u>8</u>
<u>JSP JSTL < C : FOREACH > < C : OUT ></u>	<u>9</u>
<u>JSP DE DOSYA İŞLEMLERİ OKUMA, YAZMA, OLUŞTURMA</u>	<u>10</u>
DOSYA OLUŞTURMA.....	10
DOSYA OKUMA	11
<u>JSP QUERY STRİNG KULLANIMI</u>	<u>12</u>
<u>JAVA SERVER PAGES (JSP) SWİTCH CASE KULLANIMI.....</u>	<u>13</u>
<u>RECURSİVE FONKSİYON JAVA SERVER PAGES</u>	<u>15</u>
<u>JAVA SERVER PAGES ACTIONS JSP.....</u>	<u>16</u>
<u>JSP’DE METHOD OLUŞTURMAK ÖRNEK</u>	<u>17</u>
<u>JSP’DE DİZİ KULLANIMI</u>	<u>18</u>
<u>JSP RADİO BUTTON KULLANIMI</u>	<u>19</u>
<u>JSP CHECKBOX KULLANIMI.....</u>	<u>20</u>
<u>JSP SAYFALARDA JAVASCRIPT KULLANMAK.....</u>	<u>22</u>

<u>COOKİES (ÇEREZ) OLUŞTURMA VE KULLANIMI</u>	<u>23</u>
<u>JSP SESSION OLUŞTURMA.....</u>	<u>25</u>
<u>JSP SESSION KAPATMAK (JAVA SERVER PAGES).....</u>	<u>27</u>
<u>JSP SAYFASI ÇAĞIRMA (JSP INCLUDE FILE)</u>	<u>27</u>
<u>JAVA SERVER PAGES (JSP) CACHE CONTROL</u>	<u>29</u>
<u>REQUEST METODU</u>	<u>30</u>
<u>JSP YEREL TARİH SAAT BİLGİLERİNİZİ ALMAK.....</u>	<u>31</u>
<u>JAVA SERVER PAGES OTOMATİK YENİLEME.....</u>	<u>32</u>
<u>JSP URL BULMA (GETREQUESTURL)</u>	<u>33</u>
<u>JSP KULLANICININ DİL VE ÜLKE BİLGİSİNİ BULMAK</u>	<u>34</u>
<u>JAVA SERVER PAGES HATA SAYFASI OLUŞTURMA</u>	<u>35</u>
<u>JSP HATA SAYFALARINI YÖNETMEK.....</u>	<u>36</u>
<u>JSP HATA MESAJI GÖSTERMEK.....</u>	<u>37</u>
<u>JSP MYSQL BAĞLANTISI KURMAK</u>	<u>39</u>
<u>VERİ TABANINA KAYIT EKLEME (MYSQL)</u>	<u>41</u>
<u>VERİ TABANINDA KAYIT ARAMA (MYSQL)</u>	<u>44</u>
<u>JSP TÜRKÇE KARAKTER SORUNU.....</u>	<u>47</u>
<u>JSP VERİLERİ DİNAMİK OLARAK SAYFALATMA.....</u>	<u>48</u>
<u>JSP 'DE XML VERİ GÖSTERİMİ</u>	<u>51</u>
<u>UYGULAMA.....</u>	<u>51</u>
<u>JAVA SERVER PAGES THREAD SAFE</u>	<u>53</u>

<u>THREAD NEDİR?</u>	<u>53</u>
<u>JAVA SERVER PAGES İLE JAVA BEANS KULLANIMI</u>	<u>54</u>
<u>ORM OBJECT RELATIONAL MAPPING NEDİR?</u>	<u>55</u>
<u>JSP'DE ÜYE KAYIT FORMU ÖRNEĞİ.....</u>	<u>57</u>
<u>JSP SAYFA SAYACI YAPIMI (JSP ÖRNEK)</u>	<u>61</u>
<u>JSP JAVA BEAN ÖRNEĞİ</u>	<u>62</u>

Java Server Pages Nedir?

Web sayfalarında Java dilini kullanarak dinamik web sayfaları oluşturmamızı sağlayan bir Java teknolojisidir. Jsp (Java Server Page) Html dili içine yazılır ve özel taglar kullanarak " <% %> " yazılır.

JSP, Servlet uzantısıdır. Html içerisine yazdığımız JSP kodları sunucuda Servlet'e çevrilir. JSP'ler Java Beans'ler le çalışır ve daha rahat bir geliştirme imkanı sunar.Eğer Java biliyorsanız Html içerisine zorlanmadan JSP kodlayabilirsiniz. JSP Platform bağımsızdır. Java desteği olan her yerde rahatlıkla çalışabilir. Başka bir güzel yanı ise Java Class'ları ile birlikte çalışabilmektedir.

JSP'nin Çalıştığı Yer Uygulama Sunucusudur.

JSP'de yazdığınız kodların çalışması için Web Sunucularına değil Uygulama Sunucu'suna atmanız gerekir. JSP bir Java uygulamasıdır. Bu yüzden JSP'nin çalışması için gerekli olan Uygulama Sunucu'sudur. Apache Tomcat bir uygulama sunucusudur.

JSP hakkında biraz bilgilendiysek diğer bölüm de JSP için gerekli uygulama geliştiricimizi ve uygulama sunucumuzu kuracağız.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/jspye-giris-java-server-pages-nedir-ders-1.html/>

Apache Tomcat Kurulumu ve Kullanımı

Apache Tomcat, Java Server Pages yani JSP sayfalarımızın çalışması için bilgisayarımıza kurduğumuz bir server diyebiliriz. Apache Tomcat gibi JSP sayfalarımızın çalışması için başka server ürünleri mevcut. Ama Apache Tomcat'i önermemin sebeplerinden biriside Netbeans'da projemizi çalıştırdığımız zaman aktif olması ve Netbeans'ta ki projemizi kapatınca kendini kapatması. Diğer server tiplerinde bu özellik var mı bilmiyorum. Sistemde arkaplan da çalışmaması gerçekten çok hoş.

Apache Tomcat Kurulumu

1. Öncelikle yapmamız gereken Apache Tomcat'i bilgisayarımıza indirmek. Bunun için <http://tomcat.apache.org/> sitesinde Tomcat'in en son sürümünü indiriyoruz.
2. Zipli dosyayı bilgisayarımıza indirdikten sonra zipten çıkaralım ve C:\ dizinimize kopyalayalım.
3. Kurulum aşaması bu kadar.

Apache Tomcat Ayarları

Tomcat'imize Java'yı tanıtmamız gerekmekte. Bunun için Ortam değişkenlerimizi tanıtmamız gerekiyor.

1. Bilgisayarım' a sağ tıklayıp özellikler dedikten sonra Gelişmiş sekmesinden Ortam Değişkenleri butonuna tıklıyoruz.
2. kullanıcı değişkenleri altında bulunan Yeni butonuna tıklıyoruz ve karşımıza gelen Değişken adına **JAVA_HOME**, değişken değerine de jdk adresimizi yani **C:\ProgramFiles (x86) \Java\jdk1.6.0_23** yazıyoruz ve tamam diyoruz. (Bu yol bilgisayarınızın türüne göre değişebilir. 32-64 bit farklılıkları olabilir.)
3. Ardından Sistem Değişkenlerinde **Path** i bulup düzenle butonuna tıklıyoruz. Daha sonra değişken değeri satırının başına **C:\ProgramFiles (x86) \Java\jdk1.6.0_23bin;** adresini ekliyoruz.

Ortam Değişkenlerinin Kontrolünün Yapılması

Bu yolları ekledikten sonra doğruluğundan emin olmamız gerekmekte. Bu yolların gerçekten Java dizinine doğru şekilde eklediğimizden emin olmak için yapmamız gerekenler ise şunlar.

1. Başlat->Çalıştır->cmd yazıp Enter tuşuna basalım.
2. Karşımıza çıkan komut sisteminde ise C:\> echo%java_home% yazdıktan sonra Java yolunu doğru veriyorsa sorunsuz bir şekilde tanıttık demektir.

Apache Tomcat'in Çalıştırılması

Tomcat'i kurduk. Ortam değişkenlerini ayarladık. Buraya kadar bir sorun yok ise Sıra server'imizi çalıştırmaya geldi.

1. **C:\tomcat\bin\startup.bat** dosyasını çalıştıralım.
2. Ardından herhangi bir internet tarayıcımızın adres çubuğuna <http://localhost:8080/> yazalım ve Enter'a basalım.

3. Karşınıza Apache Tomcat sayfası geliyorsa sorunsuz bir şekilde server'ınızı kurdunuz demektir.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/apcahe-tomcat-kurulumu-ve-kullanimi.html/>

MySql Server MySql WorkBench Kurulumu

Veritabanları programlarımızın vazgeçilmezidir. Bir yazılımda veri herşeydir.

MySql veritabanı kurulumuna geçelim ardından nasıl kullanılır ona bakalım.

MySql Server bilgisayarımızı bir veritabanı servisi oluşturmamızı sağlayacak. MySql Workbench ise MySql Server'ımızı kolayca yönetebilmek için bir arayüzdür.

MySql Server Kurulumu

1. Bu adım MySql'i bilgisayarımıza indirmek olacak. <http://www.mysql.com/downloads/mysql/> adresinden sisteminizin gereksinimlerinize uygun olan versiyonu seçiniz.
2. İndirdiğiniz dosyayı bilgisayarınıza kurun.
3. Kurulum işlemi tamamlandıktan sonra konfigüre etmemiz gerekiyor.

MySql Server Konfigüre Etmek

1. Karşımıza bir sihirbaz gelecek bu sihirbaz bize yardımcı olacak.
2. **Detailed Configuration**'u seçelim ve next tuşuna basalım.
3. Bu adımda **Server Machine** seçeneğini işaretleyelim. (Bu adımda makinemizi server olarak tanıttık)
4. **Multifunctional Database** seçeneğini seçelim ve next tuşuna basalım.
5. Bu adımda ne kadar yer kaplayacağı belirtiliyor next tuşuna basalım.
6. OLAP seçeneğini seçelim. (Kaç kişilik bir sistem kullandığımızı soruyor)
7. Bu adım çok önemlidir. MySql Server'ımızın kullanacağı port ve TCP/IP kullanımına izin vermemiz gerekiyor. Port otomatik olarak gelir bir değişiklik yapmamaya çalışalım. Ardından **Enable TCP/IP Networking** ve **Enable Strict Mode**'yi seçelim.

8. Veritabanımızın kullanacağı dili belirtelim. **Best Support For Multilanguazim**'im seçelim.
9. Son adım ise MySql Şifresi oluşturmak ve kurulumu tamamlamak.

MySql Server WorkBench Kurulumu

1. <http://www.mysql.com/downloads/workbench/> adresinden indiriyoruz ve kuruyoruz

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/mysql-server-mysql-workbench-kurulumu.html/>

JSP Temelleri ve Merhaba Dünya

Bugün Java Server Page (JSP) ile ilk web sitemizi yapacağız ve bu tabiki “Hello World” yazısı olacak. JSP kodlamaya geçmeden önce bir hatırlatma yapalım.

Jsp öğrenmek, ilerlemek isteyen kişiler “Html” dilini ve “Html” de bulunan sıklıkla karşılaşacağınız “tag” ları bilmenizde fayda var. JSP (Java Server Pages) Html taglarının arasına yazılır ve Html ile paralel çalışır diyebilirim. İleri ki derslerde detaylı olarak göreceksiniz.

Şimdi sadece şunu bilelim JSP `<body> </body>` arasında kodlanır. (Şimdilik bunu bilmek yeterli ileriki yazılarımda bu söz geçerliliğini kaybedecek).

<body> tag'ı: Html formatta tarayıcıda gösterilmek istenen yazı resim vb. gösterilmesini sağlayan bir tag dır. Gelelim “Merhaba Dünya” yazmaya.

Jsp (Java Server Pages) için Java kodlarını nereye yazacağımızı açıkladık. Şimdi ise sıra geldi JSP sayfamızda Java yazmaya. Unutmamalıyız ki JSP'ler de tag diye tabir ettiğimiz etiketlerle yazılır. JSP için ise bu tag `<% %>` taglarıdır. Bu taglar arasında Java yazabileceğiz.

Genel Görünüm şöyle olacak:

```
1 <% %>
```

ve biz Java kodlarımızı

```
1 <% %>
```

tagları arasına yazacağız.

Merhaba dünya yazalım ve Jsp Sayfamızı çalıştıralım.

```
1 <% out.print("Merhaba Dünya"); %>
```

Java aşinalığı olanlar bilir Java Console uygulamalarında ekrana yazı yazdırmak için kullanılan System.out.print komutu yerine JSP de sadece out.print yazmamız gerekmekte.

JSP'ye genel bir giriş yaptık. Tag özelliğinden bahsettik. Diğer Jsp derslerinde biraz daha derine ineceğiz.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/jsp-temelleri-ve-merhaba-dunya.html/>

JSP Etiketleri – JSP Tag 'lar

[Java Server Pages Dersleri](#)imizin bu bölümünde çok önemli bir konudan bahsedeceğim.

Java Server Pages kodlama stili taglar arasında olmaktadır. Her tag'ın belirli bir işlevi bulunmaktadır. Bu taglar kod okunmasını kolaylaştırır ve profesyonel projeler de JSP kullanımı taglar özelliklerine göre kodlanmaktadır.

Bu tagları kısaca açıklayalım. İleriki derslerde tagları detaylı olarak yazacağım

- **1. Declaration Tag (Bildiri Tagı)**

Bu tag arasında değişken belirleme ve ya o değişkene değer atama işlemleri yapılmaktadır. Tag'ın ifadesi `<%! %>` şeklindedir.

- **2. Expression Tag (İfade Tagı)**

JSP sayfalarımızda Java ifadelerini yazmak için başka bir yöntem. Hiç kullanmadım demsem yeridir. Hml kodlama satırında tek adımlık Java ifadeleri için kullanmaktayım. İfade `<%= %>` şeklindedir.

- **3. Directive Tag (Emir Tagı)**

JSP sayfamıza özellikler eklemeye kullanılır. Bu özellikler kütüphane ekleme, extends kullanma, hata sayfaları eklemek, JSP sayfamızın dil özellikleri gibi bir sürü emirler vererek

özelleştirebiliriz. Kullanımı ise `<%@ %>` şeklindedir.

- **4. Scriptlet Tag (Kodlama Tagı)**

JSP sayfamızda Java kod parçalarını yazmak için kullanılır. Kullanımı `<% %>` şeklindedir.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-server-pages-etiketleri-jsp-tag-lar.html/>

Java Standart Tag Library Nedir? JSTL Kullanımı

Java ile web programcılığına JSP ile giriş yaptım bunları ders halinde blogumdan anlatmaya devam etmekteyim diğer Java Server Pages derslerine ulaşmak için [buradan](#).

Java Standart Tag Library ifadesini biraz açalım. Java Server Pages ve Java Server Faces Java ile Web programlama yapabildiğimiz aracı kodlarımız.

Bu JSP ve JSF kodlamada Java kodlamaktayız. Java Standart Tag Library ile beraber JSP sayfamızda daha az kodlama yapıyoruz. Bu daha az kodlama bizim işlerimizi kolaylaştırmaktadır.

Tag Library'yi projemizde kullanmak için proje kütüphanemize eklememiz gerekebilir. Kullanılan / kullandığım library tag'lar ise şunlar.

1. **Core (prefix: c)** : En çok kullandığım kütüphanelerden birisi. Akış kontrolleri, url yönetimi gibi işlemleri rahatça yapabilmekteyiz.
2. **XML (prefix: x)** : Adından da anlaşılacağı üzere XML ile ilgili işlemler yapabilmemizi sağlamaktadır.
3. **Internationalization (prefix: fmt)** : Tarih , sayı işlemleri için kullanılır.
4. **Database (prefix: sql)** : Bu da çok sık kullandığım ve kullanacağınız Veri tabanı işlemleri için gerekli olan neredeyse tüm işlemleri yapabiliyoruz.

Maddelere geçmeden öncede söylediğim gibi bu tag'ları kullanmadan önce import etmemiz gerekmekte. JSP sayfalarımızda bu tag'ları kullanmamız için gerekli import komutları ise şöyle.

```
1 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
2 <%@ taglib prefix="x" uri="http://java.sun.com/jsp/jstl/xml"%>
3 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
```

- Bu prefixlerin kullanım tag'ları ise şöyle;

Core Etiketleri

c:catch
c:choose
c:forEach
c:forEachTokens
c:if
c:import
c:otherwise
c:out
c:param
c:redirect
c:remove
c:set
c:url
c:when

XML Etiketleri

x:choose
x:forEach
x:if
x:otherwise
x:out
x:param
x:parse
x:set
x:transform
x:when

i18n Etiketleri

fmt:bundle
fmt:formatDate
fmt:formatNumber
fmt:message
fmt:param
fmt:parseDate
fmt:parseNumber
fmt:requestEncoding
fmt:setBundle
fmt:setLocale
fmt:setTimeZone
fmt:timeZone

SQL Etiketleri

sql:dateParam
sql:param
sql:query
sql:setDataSource
sql:transaction
sql:update

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-standart-tag-library-nedir-jstl-kullanimi.html/>

JSP JSTL < c : catch > < c : if >

[Java Server Pages Dersleri](#)'nde bugün Daha önce ki derslerde işlediğimiz [Java Standart Tag Library Nedir? JSTL Kullanımı](#) ile ilgili JSTL'in etiketlerinden Catch ve if etiketlerini bu dersimizde inceleyeceğiz.

JSTL kullanmak JSP projelerimizde bize kullanım kolaylığı sağlamaktadır.

<c:catch>

Catch Tag adından da anlaşılacağı üzere "yakalama" üzerine olan işlemleri gerçekleştirmektedir. Yakaladığı durumlar ise hatadır. Yaptığımız bir projede bir tekrar etmesi gereken işlemler döngüler varsa bir hata meydana geldiğinde bu hatayı yakalatıp, sonucunu bize gönderen bir etikettir. Tek başına kullanılabilir fakat anlamsız kalabilir. Bu hatanın sonucunu bir değerlendirmeye bağlayarak yapılacak olan işlemleri devam edebiliriz. Bunu da aşağıda açıklayacağım <c:if> tagı ile bir bütün haline getirip hatayı daha da anlamlaştırabilmekteyiz.

Bu tag'ı daha iyi anlamamız için bir örnek yapalım. Bu yapacağımız örnekte programımıza hata oluşturacağız.

Programımıza matematiksel bir hata verelim.

```
1 <c:catch var ="catchException">
2 <% int sayi = 18/0; %>
3 </c:catch>
```

Bu kod bloğu hata verecektir. Bu hatayı catch yakaladı. Bu hata ile neler yapabiliriz yazının devamında açıklayacağım Şimdi öğrenmemiz gereken bir tag var.

<c:if>

if tagımız bir koşula bağlı olayı yaptırmamızda kullanılmaktadır. Bir karşılaştırma ya da bir sonuca bağlı durumlarda isterleri bu tag blogunda yazılmaktadır.

Bir örnek ile açıklayalım.

Yukarıda Catch tag'ını açıklamak için bir örnek yaptık. Bu örneğimizde bir hata var bu hatayı bir koşula bağlayalım. Yani bu işlem de bir hata var ise ekrana bir mesaj yazdıralım.

```
1 <c:if sonuc = "${catchException != null}">
2 Bir hata meydana geldi. Lütfen İşleminizi kontrol edin.
3 </c:if>
```


Bu kod blogu ise yukarıda meydana gelen hatayı yani "catchException" 'u kontrol etti ve "catchException" da "null" değeri var ise <c:if> bloğu içerisindeki mesajı tarayıcıda gösterdi.

JSP'de JSTL kullanmak örnekler de de görüldüğü gibi zaman ve kod karmaşıklığı büyük bir ölçüde engellemekte.

* Bu konu hakkında sorularınız/yorumlarınız için:
<http://blog.burakkutbay.com/jsp-jstl-catch-kullanimi.html/>

JSP JSTL < c : forEach > < c : out >

Java Server Pages Dersleri'ne devam ediyoruz. Bu dersimizde daha önce açıkladığımız [Java Standart Tag Library Nedir? JSTL Kullanımı](#)'na detaylıca inceliyoruz. JSTL ile alakalı diğer ders ise [JSP JSTL < c : catch > < c : if >](#) idi . Bu dersde ise forEach ve out kavramlarını inceleyelim. JSTL kullanmak JSP projelerimizde bize kullanım kolaylığı sağlamaktadır.

<c:forEach>

Bu JSTL' bir dögüdür. For dögüsü gibi işlem görmektedir ancak kullanımı daha kolay olmaktadır.

```
1 <c:forEach var="degisken_adi" begin="1" end="80">
```

Yapımız bu şekilde önceden alışık olduğumuz dögü yapısındadır. Bu yapı genelde veri çekerken kullanılmakta.

<c:out>

```
1 <c:out value="{degisken_adi}"/><p>
```

Bu tag ise bize dögüden gelen sonuçları ekrana yansıtmanızı sağlamaktadır.

Veritabanımızdan verileri çekerken bu kullanım bize çok kolaylık sağlamaktadır.

Toparlarsak....

```
1 <c:forEach var="degisken_adi" begin="1" end="80">
2   Liste <c:out value="{degisken_adi}"/><p>
3 </c:forEach>
```

forEach blokları arasında ki kodlar döngü devam ettiği sürece çalışmaya devam etmektedir. Döngü sonucu ise out tag'ı ile kullanabilmekteyiz.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/jsp-istl-c-foreach-c-out.html/>

Jsp de Dosya İşlemleri Okuma, Yazma, Oluşturma

[Java Server Pages Ders](#)imizin bu yazısında, Java Server Pages üzerinden;

Dosya Oluşturma;

Dosya Yazma;

Dosya Okuma;

işlemlerini göreceğiz.

Dosya işlemlerinde kullanacağımız kütüphane java.io.* kütüphaneleridir. Dosya yazma işlemine başlamadan önce bu kütüphaneyi programımıza eklemeyi ihmal etmeyelim.

Dosya Oluşturma

Java Server Pages'te Dosya Oluşturma işlemi FileOutputStream komutu ile olmaktadır. JSP'de bir dosya oluşturmak istiyorsak komutumuz.

```
1 <%@page import="java.io.*"%>
2 <%@page contentType="text/html" pageEncoding="UTF-8"%>
3 <%@ page import="javax.servlet.http.HttpUtils,java.util.*" %>
4
5 <%
6 try{
7   FileOutputStream dosya = new FileOutputStream("C://Deneme.txt");
8   out.println ("Dosya Oluşturuldu");
9 }
10 catch (IOException e){
11   out.println ("Dosya Oluşturulamadı");
12 }
```

Dosya Yazma

Dosyamızı oluşturduk. Oluşan dosyamıza bir metin ya da kayıt eklemek için kullanılan komut ise `PrintStream` ile gerçekleşmektedir.

```
1 <%@page import="java.io.*"%>
2 <%@page contentType="text/html" pageEncoding="UTF-8"%>
3 <%@ page import="javax.servlet.http.HttpUtils,java.util.*" %>
4
5 <%
6 try{
7   FileOutputStream dosya = new FileOutputStream("C://Deneme.txt");
8   out.println ("Dosya Oluşturuldu");
9   try{
10    new PrintStream(dosya).println("Burak Kutbay");
11    out.println ("Verileriniz Dosyaya Eklendi.");
12    dosya.close();
13   }
14   catch(Exception e){
15    out.print("Verileriniz Dosyaya Eklenemedi.");
16   }
17  }
18  catch (IOException e){
19   out.println ("Dosya Oluşturulamadı");
20  }
21 %>
```

Dosya Okuma

Verilerimizi dosyamıza kayıt ettik. Bu verileri okumak istersek `BufferedReader` komutu ile gerçekleştiriyoruz.

```
1 <%@page import="java.io.*"%>
2 <%@page contentType="text/html" pageEncoding="UTF-8"%>
3 <%@ page import="javax.servlet.http.HttpUtils,java.util.*" %>
4
5 <%
6 try{
7   out.println("Dosya Bilgileri"+"
8   ");
9   BufferedReader dosya = new BufferedReader(new FileReader("C://Deneme.txt"));
```

```
10 String bosluk = "";
11 while ((bosluk = dosya.readLine()) != null) {
12     out.println("*"+bosluk+"");
13 }
14 dosya.close();
15 catch(IOException e)
16 {
17     out.print("");
18 }
19 %>
```

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/jsp-de-dosya-islemleri-okuma-yazma-olusturma.html/>

Jsp Query String Kullanımı

Query String denilen kavram nedir ilk önce bu kavramı inceleyelim.

Query String tarayıcımızın adres çubuğundan oluşturduğumuz başka bir sayfaya veri taşıma işlemidir. Adres çubuğundan veri taşıma işlemi "?" ve "&" işaretleri olur ve bu bilgileri ayırır.

Adres çubuğu'nda bilgiler bu şekilde gözükür.

```
1 kullanıcıbilgi.jsp?adi=Burak&soyadi=kutbay
```

Bu adres çubuğunda ki bilgileri sayfamıza almamız gerekmektedir. Adres çubuğunda ki verileri Request.QueryString metodu ile almaktayız.

```
1 String kullanıcıadi=request.getParameter("adi");
2 String kullanıcısoyadi=request.getParameter("soyadi");
```

request.getParameter metodu ile bilgileri JSP sayfamıza çağırıp kullanabiliyoruz.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/jsp-query-string-kullanimi-java-server-pages-request-getparameter.html/>

Java Server Pages (JSP) Switch Case Kullanımı

Java Server Pages Derslerimize devam ediyoruz. Bu ders temel programlama ifadeleri diyebileceğim (seçim üzerine) ifadelerin JSP sayfalarımızda nasıl kullanıldığı ile alakalı olacak. Bu dersler biraz basit gibi görünse de neden yazıyorsun bunları diye sorabilirsiniz. Gelen geribildirimlere göre programlamayı yeni öğrenen dostlarımız bu ifadelerin hem HTML hem de Java ile nasıl çalıştığını merak etmekte. O dostları kırmamak adına hemde JSP ile alakalı ne varsa basit, zor kavramı yapmadan buraya not etmek istiyorum.

Uzun bir dipnot'tan sonra konumuza dönelim. "Switch Case" kullanıcıya bir seçim yaptırdıktan sonra o programımıza o işlemi yaptırmak ve sonra kesmek üstüne kurulu bir ifadedir. Bu kullanım genel de kullanıcıya kısıtlı seçenekler sunan ve o seçeneklerin kesin olarak bir uygulama yöntemi olup başka bir kodun çalışmasına gerek olmayan kesin ifadelerde kullanılır. Yani programımızda o seçime göre kesin bir ifade olacak ve bunun sonucunda kullanıcıya kesin bir sonuç vermesi planlanan durumlarda kullanılır.

Switch-Case'nin detaylı anlatımını bu ders üzerinde yapmayı doğru bulmuyorum zaten JSP ile ilgili bir örnek üzerinde gördüğünüzde anlayacağınızı umuyorum.

Java Server Pages örneğimiz de kullanacağımız senaryo ise şu olacak. Sayfamız da bir form olacak bu formda kullanıcıdan bir giriş isteyeceğiz. Kullanıcıya "Hangi Ay" diye soracağız ve rakam cinsinden bir değer girecek bu değere karşılık gelen ayı ise JSP sayfamız bize söyleyecek şimdi formumuzu tasarlayalım.

index.jsp

```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <title>Burak Kutbay Com Switch Case Uygulaması</title>
7   </head>
8   <body>
9
10    <form name="form1" method="get" action="sonuc.jsp">
11      <label>Hangi Ay</label>
12      <input type="text" name="ay" id="ay">
13      <input type="submit" name="gonder" id="gonder" value="Gönder">
14    </form>
```

```
15
16     </body>
17 </html>
```

index.jsp sayfamızdan aldığımız değeri sonuc.jsp formuna taşıyacağız bu işlemi GET metodunu kullanarak gerçekleştireceğiz. Önceki dersimizde bahsettiğim [Jsp Query String Kullanımı](#) ile kullanıcıdan gelecek ay cevabı bilgisini alıp Swicth – Case blokuna alcağz.

sonuc.jsp

```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4     <head>
5         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6         <title>Switch Case Sonuç</title>
7     </head>
8     <body>
9         <% String formdangelen="0";
10        formdangelen=request.getParameter("ay");
11        int a=Integer.parseInt(formdangelen);
12        switch(a) {
13            case 1:
14                out.println("Ocak");
15                break;
16            case 2:
17                out.println("Şubat");
18                break;
19            case 3:
20                out.println("Mart");
21                break;
22            case 4:
23                out.println("Nisan");
24                break;
25            case 5:
26                out.println("Mayıs");
27                break;
28            case 6:
29                out.println("Haziran");
30                break;
31            case 7:
32                out.println("Temmuz");
33                break;
34            case 8:
35                out.println("Ağustos");
36                break;
37            case 9:
```

```
38     out.println("Eylül");
39     break;
40     case 10:
41         out.println("Ekim");
42         break;
43     case 11:
44         out.println("Kasım");
45         break;
46     case 12:
47         out.println("Aralık");
48         break;
49     default:
50         out.println("Ay Seçelim. Güzelleşelim");
51 }
52
53 %>
54 </body>
55 </html>
```

index.jsp sayfamızdan gelen değeri sonuc.jsp sayfamıza aldık. Alınan değerin karşılığına gelen veriyi kullanıcıya gösterdik. Güzel kodlamalar. (:

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-server-pages-jsp-switch-case-kullanimi.html/>

Recursive Fonksiyon Java Server Pages

Java Server Pages Dersimizin bu günkü konusu özyinelemeli fonksiyon olarak tabir ettiğimiz recursive fonksiyonlardan olacaktır. Fonksiyon kavramını biraz açalım. Programlama dillerinde fonksiyon ana program dışı ve ya içinde ayrı bir kod bloklarıdır. Bu bloklar ana programdan değer alıp değer gönderebilirler.

Programlarımızda tekrar tekrar kullandığımız işlemler için kullanılır ve bu sistemimizi yorar. Recursive fonksiyon sistemi olabilecek en düşük seviyede çalışmasını sağlar. Kendi kendini çağırarak fonksiyonumuz bir döngü şeklinde çalışmaya devam eder.

Java Server Pages'de Recursive Fonksiyonları kullanımının yazmamdaki neden gelen istek. JSP'de bu işlemin nasıl olduğunu ([iletisim](#)) soranlar oldukça fazla idi.

Aslında bilinen kullanım şekline hiçbir farkı bulunmamaktadır. Şimdi örneğimize geçelim. Bu örnek oldukça basit ve recursive fonksiyonlar'ı kavramak için verilen örneği yapalım. Faktöriyel örneğini [Java Server Pages](#)'de yapalım.

```
1 <HTML>
2 <HEAD>
3 <TITLE>Burak Kutbay Blog</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>Recursive Fonksiyon</H1>
7 <%!
8 int faktor(int n)
9 {
10     if (n == 1) {
11         return n;
12     }
13     else {
14         return n * faktor(n - 1);
15     }
16 }
17 %>
18 <%
19     out.println("15 in faktoriyeli... " + faktor(15));
20 %>
21 </BODY>
22 </HTML>
```

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/recursive-fonksiyon-jsp-java-server-pages.html/>

Java Server Pages Actions JSP

[Java Server Pages Dersleri](#)'nde bu yazıda Java Server Pages'te kullanılan eylemleri sizlere inceleyeceğiz.

Özetle yazacak olursak. Java kodları oluşturmadan nesne oluşturma o nesnelere kullanma eylemidir. Bu yöntem servlet tarafına yani sunucu tarafına gönderi yapılması için kullanabiliriz. Bu "action" kullanımı zorunlu değil burada açıklayacağım kullanım türleri Java kodları ile de rahatça yapılmaktadır.

Kullanımı:

```
1 <%jsp:action_name attribute="value" %>;
```


Eylemler Syntax	Açıklama
<code>jsp:include</code>	Bir jsp sayfasını bulunduğunuz başka bir sayfayı dahil edebiliriz.
<code>jsp:useBean</code>	JavaBean'i Jsp sayfamıza tanımlıyoruz.
<code>jsp:setProperty</code>	JavaBean'lerin özelliklerini değiştirmek için kullanılan eylem.
<code>jsp:getProperty</code>	JavaBean'lerin giriş özelliklerini değiştirebileceğimiz eylem.
<code>jsp:forward</code>	Bir jsp sayfasından başka bir jsp sayfasına yönlendirme yapmaktadır.
<code>jsp:plugin</code>	Sayfamıza Java Eklentisi yerleştirebiliriz. Object ve ya Embed olarak.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-server-pages-actions-jsp-eylemler.html/>

JSP'de Method Oluşturmak Örnek

Method Nedir?

Method kısaca bizim programımızda kullanmak üzere yazdığımız küçük program parçalarıdır. Nesneye Yönelimli Programlama'lar da kullanılan en temel yöntemdir. Metod tekrarlanan işlemleri yapacağımız gibi ayrıca bize zaman ve sistemlerimizin de kaynaklarını az kullanmamızı sağlamaktadır.

Method kavramı oldukça geniş bir kavram ancak bizim konumuz JSP ve bu dersimizde Java Server Pages'te nasıl kullanıldığına bakacağız.

Method oluşturmayı bildiğimizi umarak Java Server Pages'te nasıl kullanılır bakalım.

Öncelikle kullanacağımız JSP tag'ı

```
1 <%!  
2 .....  
3 ...  
4 %>
```

Bu JSP tag'ını daha önceki derste görmüştük. Bu tag bildiri tagı'dır. ([oku](#))Method'umuzu yazacağımız tag'ı öğrendik ve bu taglar arasına method'larımızı yazdıktan sonra Kod tag'ımızdan verileri gönderip ekranda sonucu göstereceğiz.

```
1 <html>  
2 <head>  
3 <title>Burak Kutbay</title>  
4 </head>
```

```
5 <body>
6 <%!
7     int topla(int sayi1, int sayi2) {
8         return sayi1 + sayi2;
9     }
10 %>
11 <%
12     int sonuc = topla(5, 9);
13     out.println(sonuc);
14 %>
15 </body>
16 </html>
```

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-server-pages-method-olusturmak.html/>

JSP'de Dizi Kullanımı

Bu yazıda Java Server Pages'de dizi kullanımına bakacağız.

Programlama da dizi demek birden fazla değeri aynı değişkende depolamaya ve depo'dan kullanmamıza olanak tanımaktadır.

Java Server Pages'de dizi oluşturma işlemi;

```
1 <%
2 String[] dizi={"bu","ra","k"};
3 %>
```

kullanılmaktadır.

Bizim işlemlerimizde belirli bir değer bulunmuyor olabilir. Bu durumlarda başka sayfalardan gelen değerleri'de dizimize atmamız mümkün belirteyim.

Bu dizimizi her amaç için kullanmamız mümkün herhangi bir program dilinde yapmak istediğiniz tüm işlemleri JSP içinde yapabilirsiniz. Biz bu dizimizdeki değerleri ekrana yazdıralım.

```
1 <%for(int x=0;x<dizi.length;j++){
2     out.println(dizi[x]);
3 }
4 %>
```

Basit bir örnek ile açıkladım. Java Server Pages ile dizi oluşturup, işlemlerinize göre kullanmanız bu kadar kolay

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/jspde-dizi-kullanimi-java-server-pages.html/>

JSP Radio Button Kullanımı

Bu konumuzda sayfamızdan oluşturduğumuz formumuzda Radio Button var ise bu seçilen değeri JSP sayfamızda nasıl kullanacağız inceleyelim.

Radio Button kısaca açıklamak gerekirse kullanıcıya çok seçenekten sadece bir seçenek işaretletirmek istiyorsak Radio Button kullanırız.

Html Form'dan gelecek verileri Java Server Pages'te nasıl kullanıldığına daha önceki derslerimizde bakmıştık. Bu dersler;

- [JSP ile Veri Tabanına Kayıt Ekleme \(MySql\)](#)
- [JSP'de Üye Kayıt Formu Örneği](#)

gibi yazılarımızda yapmıştık. Yine asıl mantık formdan gelen değeri

1 request.getParameter

olarak almak olacak. Yukarıda paylaştığım dersler de de bunu yapmıştık.

Radio Button oluşturalım ve bu formdan gelen değeri Jsp sayfamıza alalım.

index.jsp

```
1 <html>
2 <head>
3 <title>Burakkutbay.com</title>
4 </head>
5 <body>
6 <form name="secenek_formu" action="sonucgonder.jsp" method="post" >
7 <input name="radioButton" type="radio" value="dusunuyorum"
8 checked="checked" /> a
9 <input name="radioButton" type="radio" value="dusunmuyorum" /> b
10 <input name="radioButton" type="radio" value="daha sonra hatırlat" /> c
11 <input name="kaydet" type="submit" value="gonder" />
12 </form>
```

```
13 </body>
14 </html>
```

Ana sayfamızda form'mumuzu oluşturduk. Şimdi ise bu değeri jsp sayfasında kullanmak. sonucgonder.jsp sayfasında Radio Button'dan gelen değeri alalım.

sonucgonder.jsp

```
1 <%
2     String kullanıcı_secenek= request.getParameter("radioButton");
3     System.out.println("Seçeneğiniz ::"+kullanıcı_secenek);
4 %>
```

İşlem tamam.

1 **request.getParameter**

ile seçeneğimizi aldık ve bu değer ile işlemlerimizi devam ettirebilmekteyiz.

* Bu konu hakkında sorularınız/yorumlarınız için:
<http://blog.burakkutbay.com/jsp-radio-button-kullanimi-java-server-pages.html/>

JSP CheckBox Kullanımı

[Java Server Pages Dersleri'](#)nde ki bu yazıda CheckBox kullanımından bahsedeceğim. Java Server Pages'in temel yapısını çalışma mantığını eğer bu yazıya kadar okuduysanız fikir edinip uygulamaya geçmişsinizdir diye düşünüyorum. Yaklaşık 40 ders'te Html ile içli dışlı oluyoruz. Temel Html kullanımının yanında bize çok lazım olan bileşenlerin kullanımını da Java Server Pages Derslerim de yer ayırmayı düşünüyorum.

Önceki derslerimiz de zaten form bileşenlerinden [JSP Radio Button Kullanımı](#)'ni incelemiştik. CheckBox RadioButton'dan farklı olarak birden fazla seçeneği işaretlememize olanak tanımaktadır. Yazılımlarımızda kullanıcıya birden fazla değer seçenek sunmak için bu bileşen kullanılmaktadır.

Basit bir form oluşturup formun içine CheckBox bileşenimizi yerleştireceğiz. Formumuz CheckBox'dan gelen değerleri bir diziye atacağız çünkü seçilen veriler eğer birden fazla ise bunu unutmamalıyız. Bu değeri ya da değerleri bir döngü ile (dizimizin boyutu kadar

olacak) alacağız. Alınan bu değerleri isteğimize göre kullanabilmek mümkün olacaktır. Ben kullanıcıya seçtiği değerler için kullanacağım.

Formumuz şu şekilde olacak.

Seç Beğen Al

Ben

Seçilmem

Seçerim

Kodlarımız ise;

index.jsp

```
1 <%--
2 Document : index
3 Created on : 10.Eki.2013, 21:42:47
4 Author : Burak
5 --%>
6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE html>
9 <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>JSP Page</title>
13 </head>
14 <body>
15 <form name="form" onsubmit="index.jsp">
16 <h3>Seç Beğen Al</h3>
17 <p><input type="checkbox" name="sec" value="Ben"/>Ben</p>
18 <p><input type="checkbox" name="sec" value="Secilmem"/>Seçilmem</p>
19 <p><input type="checkbox" name="sec" value="Secerim"/>Seçerim</p>
20 <p><input type="submit" value="Gönderici"/>
21 </form>
22 <%
23 String secimler[] = request.getParameterValues("sec");
24 if (secimler != null) {
25     for (int i = 0; i < secimler.length; i++) {
26         out.println(secimler[i]);
27     }
28 }
```

```
29     %>
30     </body>
31 </html>
```

olacaktır.

Burada dikkat etmemiz gereken nokta değerimizi getparameter ile aldıktan sonra bir işlem yapmadan gelen dizinin içi boş mu değil mi bunu kontrol etmeliyiz eğer boş bir değer gelirse sistemlerimiz ve biz zor duruma düşebiliriz.

Güzel kodlamalar yazılımcılar.

* Bu konu hakkında sorularınız/yorumlarınız için:
<http://blog.burakkutbay.com/jsp-checkbox-kullanimi.html/>

JSP Sayfalarda JavaScript Kullanmak

JSP sayfalarımızda Javascript nasıl kullanabiliriz, değişken değerlerini nasıl gönderip, alırs gibi soruları cevaplandırmaya çalışacağım.

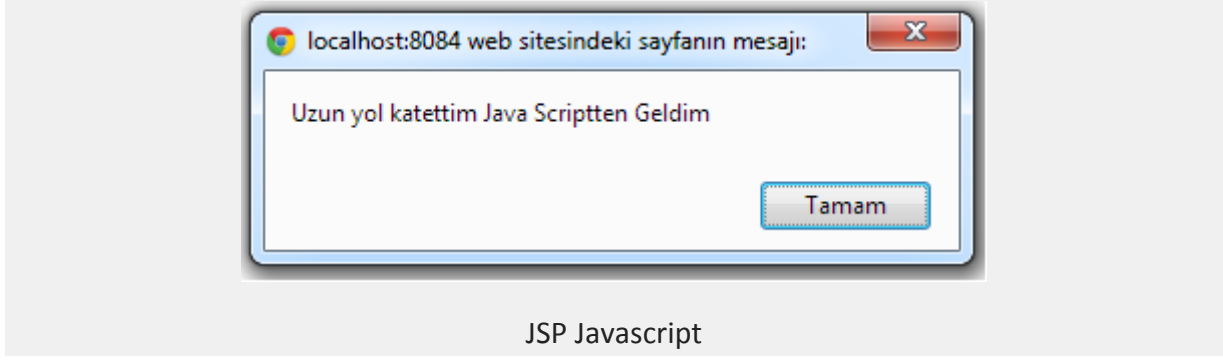
Javascript bir betiktir. Sadece o yazılan sayfa üzerinde anlık olarak çalışır ve sonlanır.

Java Server Pages Dersi'nde Javascript'i nasıl çalıştırırız. Bunun cevabı oldukça kolay. Aynı html sayfalarda nasıl bir işlem uygulanıyorsa aynısını yapmamız yeterli. Peki veri alışverişi nasıl olmakta sorusuna bir örnek yaparak cevap vereceğim.

Standart bir Java Server Pages sayfası açalım. Bu sayfaya yapacaklarımız ise şöyle.

- - *Java Server Pages sayfası oluşturacağız.*
- - *Javascript yazacağız bir fonksiyon yazacağız. Bu fonksiyonun adı mesaj olacak.*
- - *Jsp sayfamızda oluşturduğumuz "ileti" değişkenimizin değerini Javascript'te ki sonuc değişkenine atıyoruz.*
- - *Bu değişkenimizi uyarıyı ekranında göstermek için sayfamızın dolmasını beklemekteyiz.*

Sonuç sayfamızı çalıştırdığımızda şu görüntü ile karşılaşacağız.



Ve kodlarımız.

```
1 <html>
2 <head>
3 <script language="javascript">
4     function mesaj(){
5         <% String ileti="Uzun yol katettim Java Scriptten Geldim"; %>
6         var sonuc="<%=ileti%>";
7         alert(sonuc);
8     }
9 </script>
10 </head>
11 <body onload="mesaj()">
12     Burak Kutbay Blog
13 </body>
14 </html>
```

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/jsp-java-server-pages-java-script-kullanmak.html/>

Cookies (Çerez) Oluşturma ve Kullanımı

[Jsp Dersimizin](#) bu yazısında Çerez oluşturma ve oluşturduğumuz bilgileri oluşturma işlemlerini inceleyeceğiz.

Cookies yani Çerez olarak adlandırdığımız şey. İnternet sitelerinin bilgisayarımızda bıraktığı bilgilerdir. Bu bilgiler ise o internet sitesine bilgilerini giren kişilerin çerezleridir.

Çerez kavramını detaylı olarak anlatmayacağım ama kullanım kolaylığını kısaca bir örnekle açıklayayım. Çerez bırakılan bilgisayarda çerezin özelliklerine göre kapattığınız bir internet sitesini tekrar giriş yapmadan kendi sayfanızın getirilmesi sağlanabilir. Bir sayfadan başka bir sayfaya bilgilerimiz taşınmasında da çerez kavramı kullanılabilir.

Java Server Pages'te Çerez Oluşturma

```
1 <% String kullanıcı= "Burak";  
2  
3 Cookie cookie = new Cookie ("Cerez_kullanici",kullanici); %>
```

Çerez oluşturma mantığı kısaca bu.

"Cerez_kullanici" bizim oluşturduğumuz bir isim ve bu ismin karşılığında "kullanici" adlı değişkeni eşleştirdik. Bunu daha sonra kullanacağız.

Java Server Pages'te Çerez Süresi Belirleme

Bu çerezlerin bir süresi vardır. Bu belirlediğimiz süre kadar saklanır ve işleme konulur. Bu süreyi belirlemek için ise çerezimi oluşturduktan sonra;

```
1 <% cookie.setMaxAge(10000); %>
```

Süremizi oluşturduk. Bu çerezimizin yaşam süresidir.

Java Server Pages'te Çerez'i Kaydetme

Kaydetme tabiri ne kadar doğru bilemiyorum ama çerezimizi oluşturduk, süresini belirledik ve ardından bu değerleri kullanıcının bilgisayarına kayıt edeceğiz.

```
1 <% response.addCookie(cookie); %>
```

Bu kod ile kullanıcının bilgisayarına kaydetmiş olduk.

Java Server Pages'te Çerez Bulma İşlemi

Çerez'i bulma işlemi bize verilerin tekrar geri gelmesini ve o kullanıcı eğer çerezin süresi bitmemiş ise kendine izin verilen sayfalarda gezinirken internet sitesi gezen kişinin bilgilerini hatırlayıp o kullanıcıya özgü bilgileri göstermiş olacak.

```
1 <% String saklanan_cerez= "Cerez_kullanici";  
2  
3 Cookie cookies [] = request.getCookies ();  
4  
5 for (int i = 0; i < cookies.length; i++){  
6  
7 if (cookies [i].getName().equals (saklanan_cerez)){  
8  
9 String bulunan_cerez_ad = cookies[i];
```



```
10
11 break;
12
13 }} %>
```

Burada dikkat etmemiz gereken nokta biz bulunan çerezin adını aldık. Yani .getName fonksiyonu ile almış olduk. getValue() deyeydik bize bu çerezin değerini gönderecekti.

Java Server Pages'te Bulunan Çerezi Kullanma

Çerezimi bulduk ve bu çerezimizden gelen değeri hoşgeldiniz değeri

```
1 <% out.println("Hoşgeldiniz Sayın" , bulunan_cerez_ad); %>
```

* Bu konu hakkında sorularınız/yorumlarınız için:
<http://blog.burakkutbay.com/jsp-de-cookies-cerez-olusturma-ve-kullanimi-java-server-pages.html/>

JSP Session Oluşturma

[Java Server Pages Dersleri](#)'mizde bugün Session oluşturmayı göreceğiz.

Yaptığımız internet sitemizde kullanıcı veri girişi yaptıktan sonra kendi özel sayfaları arasında dolaşmalıdır. Bu dolaşım işlemi kullanıcı bilgilerini kaybetmeden gezmelidir ki kullanıcı kendi bilgilerini görerek ve ya kendine özel sayfalarında sağlıklı bir şekilde dolaşmalı ve diğer kullanıcılar başka kullanıcılarının bilgilerini görmemelidir.

Session oturum yönetimi her kullanıcıya özel olarak oluşturulur. Diğer kullanıcılar için ise farklı farklı sessionlar açılmaktadır. Oturum süresi değiştirilmediği sürece 30 dk dır. Şimdi örneğimize geçelim. Oluşturduğumuz sayfanın kullanıcı giriş sayfası olduğunu ve bu kullanıcı giriş yaptıktan sonra kendine özel sayfasına yönlendirilecek.

Kullanıcı Giriş Formu muz yapalım.

```
1 <form name=login Action=giriskontrol.jsp metod="post">
2   Kullanıcı Adınız: <input type="text" name="adi" > <BR>
3   Şifreniz:<input type="text" name="sifre" > <BR>
4   <input type="submit" value="login">
5 </form>
```

Kullanıcımız bu formdan bilgilerini girecek. Ardından kontrol sayfamız veya bean diye tabir ettiğimiz java sınıfımızda kontrol ettireceğiz. Bean kavramına daha sonra geleceğiz. Kullanıcı “login” tuşuna bastığı zaman “giriskontrol.jsp” sayfamıza çağırıp kontrol ettireceğiz.

Eğer girilen veriler doğru ise Session oluşturup kullanıcının ana sayfasına yönlendireceğiz. Bu kullanıcı ana sayfasına session oluşmayan kullanıcı giremeyecektir.

giriskontrol.jsp sayfamız şöyle olacaktır.

```
1 String kullanıcıadi=request.getParameter("adi");
2 String kullanıcısifre=request.getParameter("sifre");
```

Parameter metodu ile bilgileri aldık şimdi yapacağımız işlem ise elimizdeki veriler ile kullanıcıdan gelen verileri karşılaştırmak. Bu veriler eğer doğru ise session açıp kullanıcı sayfasına yönlendireceğiz.

```
1 if(kullanıcıadi.equals("burak") && kullanıcısifre.equals("1234"))
2 {
3     HttpSession anasayfa = request.getSession(true);
4     anasayfa.putValue("kullanıcı_giris", "gecerli");
5     response.sendRedirect("./kullanıcıanasayfa.jsp");
6 }
7 else
8 {
9     out.println("Oturum açma başarısız");
10 }
```

HttpSession metodu ile oturum yönetimi açtık ve anasayfamıza yönlendirdik. kullanıcıanasayfa.jsp sayfasında oturum sadece oturum açan kullanıcı görüntüleyebilecek. Oturum açmayan kullanıcı kullanıcıya ait sayfayı görüntülemeyecek. Bunun kontrolünü yapalım.

```
1 if(anasayfa.getValue("kullanıcı_giris")!="gecerli")
2 {
3     response.sendRedirect("index.jsp");
4 }
```

Kodları ile oturumu kontrol etmeyi sağlamaktadır.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/jsp-session-olusturma-java-server-pages.html/>

JSP Session Kapatmak

[JSP Dersleri](#)'mizin bu konusunda JSP sayfalarında açtığımız "session"ları kapatma işlemi nasıl yapılır ona bakalım. [JSP Session Oluşturma](#) Session oluşturma ve o bilgileri nasıl kullanabileceğimizi bahsetmişim.

Kısaca açıklamak gerekir ise Session kullanıcının internet sitenizde gezindiği sürece bilgilerinin kayıt atında tutulması diyebilirim.

Bu session' u açtık ve kullanıcı işlemi bittiğinde logout olmak isterse bu session' u kapatmalıyız. Kullanıcının hesabını güvenle kapatsın. Logout sayfanıza eklemeniz gereken kod budur.

```
1 <%@page language="java" session="false"%>
```

* Bu konu hakkında sorularınız/yorumlarınız için:

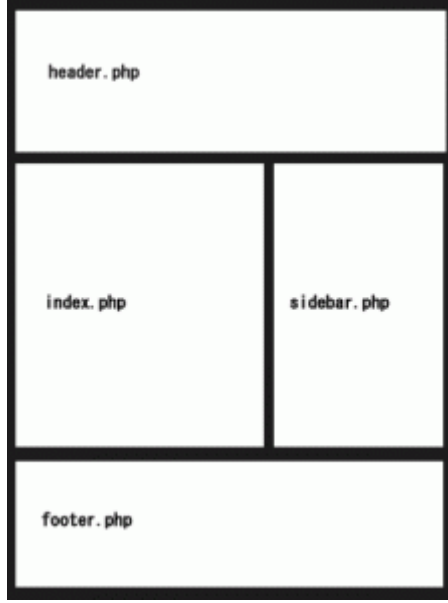
<http://blog.burakkutbay.com/jsp-session-kapatmak-java-server-pages.html/>

JSP Sayfası Çağırma (JSP Include File)

[Java Server Pages Dersleri](#)imize devam ediyoruz. Bugünkü dersi bir dip not olarak görebiliriz. Java Server Pages sayfalarımız bir blok halinde oluşmamakta. Genellikle ayrı ayrı bölümler halinde. Jsp sayfalarımı kullanmak istediğim sayfalara çağırıp kullanıyorum. Böylelikle bir blokta yapılan değişiklik diğer sayfaları etkilememekte.

Günümüz de de web sayfaları bölümler halindedir. Örnek verecek olursam şu an bu sitede yazıyı okuduğunuz bölüm **single.php** sayfasıdır. Üst kısımda görünen başlık ve menüler **header.php** sayfanın en altında görünen blok ise **footer.php** sağ blok ise **sidebar.php** sayfalarıdır.

Bu sayfalar bir bütün halinde gözükmesinin ise **include** denilen yöntem ile gerçekleşmekte. Bir sayfaya diğer sayfaları çağırarak bi bütün oluşturmaktadır. WordPress'ten yola çıkarak bizim de JSP projelerimizde sayfaları kullanıp ana sayfamıza ya da kullanmak istediğimiz sayfalara çağırarak elbette mümkün.



Bu işlem oldukça basit ve tek satırlık bir kod. Ancak tasarımcı olmadığım için ve CSS bilgim oldukça zayıf olduğu ben sadece sayfa çağırırım gerisi beni ilgilendirmez.

Üst resimde olduğu gibi tasarım uygunluğunu sağlayacak olan tasarımcı ile bu include etme işlemlerinde çok fazla diyalog içerisine girmezsiniz umarım.

- **JSP sayfasına başka bir JSP Sayfası Çağırma**

index.jsp

```
1 <%@ include file="altsayfa.jsp" %>
2 <%@ include file="altsayfa2.html" %>
```

Bu komut yardımı ile sayfamızı çağırabilirsiniz. Ben JSP sayfalarını çağırma diye başlık atsam da .html uzantılı sayfalarınızda ve destekleyen diğer sayfalarınızı bu komut yardımı ile çağırabilmeniz mümkün.

Başka bir JSP dersinde görüşmek üzere.

* Bu konu hakkında sorularınız/yorumlarınız için:
<http://blog.burakkutbay.com/jsp-sayfasi-cagirma-jsp-include-file.html/>

Java Server Pages (JSP) Cache Control

[Java Server Pages Dersleri](#)imizin bugünkü konusu güvenlik, verimlilik ile alakalı olacak. Cache Control yaptığımız sayfaların tarayıcılar ile önbelleklenme süresini ayarlamamaız mümkün. Cache Control'ün aşağıda açıklayacağım özellikleri ile tarayıcımıza yaptığımız yazılımın geçerlilik süresini belirlemek mümkün. İnternet tarayıcıları genellikle bu bilgileri ister bu bilgileri tarayıcımıza vermek internet tarayıcısının yazılımımızı tanınması ayrıca kullandığımız sistemlerin az kaynak harcamasına olanak sağlamaktadır.

Cache yapısına belirli bir süre vererek sayfalarımızın tekrar sunucumuzdan çekilmesine izin vermememiz mümkün ancak dikkat edeceğimiz durumlar bulunmakta. Bu Cache yapısı uzun süre güncellenmeyeceğinizden emin olduğunuz sayfalarda kullanmak daha sağlıklı olacaktır. Sık güncellenen sayfalarda varsa özellikle yazılımsal değişikliklerde bu Cache süresini uzattığınız anda yaptığınız değişiklikler görüntülenmeyebilmektedir. Özellikle yüksek güvenlik isteyen yazılımlarınızda Cache'lemeyi durdurmak daha sağlıklı olacaktır.Çünkü güvenlik açısından yapılan bir değişiklik Cache durdurulmadı ise kullanıcılarda o açığın devam etmesine neden olabilmektedir.

Cache sistemini kısaca anlatmaya çalıştım şimdi ise Cache Control'ü Java Server Pages'te nasıl yapılır inceleyelim.

Cache Control sistemi Setheader yöntemi uygulanmaktadır.

```
1 <%  
2 response.setHeader ("Cache-Control", "public");  
3 %>
```

Sayfamızın Önbellekleme istemiyorsak

```
1 <% httpServletResponse.setHeader("Cache-Control","no-cache"); %>
```

Belirli bir süre vermek istiyorsak

```
1 <%  
2 response.setHeader ("Cache-Control", "max-age = 1500");  
3 %>
```

yazmamız yeterli olacaktır.

Cache varsa tamamen temizlemek ve cache durdurmak için

```
1 <%
```

```
2 response.setHeader("Cache-Control","no-cache");
3 response.setHeader("Pragma","no-cache");
4 response.setDateHeader ("Expires", 0);
5 %>
```

Bu kontroller ile birlikte sistemimizin kaynaklarını daha verimli kullanmaya başlayabiliriz.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-server-pages-jsp-cache-control.html/>

Jsp Request Metod'ları

Request Sınıfı bizim isteklerimizi yerine getirmektedir. Request demek istek,bilgi demektir. Biz bu bilgiyi almak için Request sınıfının metodlarını kullanmaktayız.Bu sınıf ile ister sunucudan isterseniz de kullanıcıdan bilgi alımı konusunda bu sınıf ve metodlarını kullanacağız.

```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <%@ page import="javax.servlet.http.HttpUtils,java.util.*" %>
3 Request Metodu
4 JSP İstek Metodu: <%= request.getMethod() %>
5 İstek URI: <%= request.getRequestURI() %>
6 İstek Protokolü: <%= request.getProtocol() %>
7 Server Adı: <%= request.getServerName() %>
8 Server Port: <%= request.getServerPort() %>
9 Uzak Adres: <%= request.getRemoteAddr() %>
10 Uzak Host: <%= request.getRemoteHost() %>
11 Local: <%= request.getLocale() %>
12 Bulunduğun URL: <%=request.getRequestURL()%>
```

Çıktısı ise şöyle.

Request Metodu

JSP İstek Metodu: GET

İstek URI: /WebApplication1/

İstek Protokolü: HTTP/1.1

Server Adı: localhost

Server Port: 8084

Uzak Adres: 127.0.0.1 Uzak Host: 127.0.0.1

Local: tr_TR Bulunduğun URL: http://localhost:8084/WebApplication1/

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/jsp-request-metodlari-java-server-pages.html/>

JSP Yerel Tarih Saat Bilgilerinizi Almak

Bu gün bir uygulama yapacağız bu uygulama ile kullanıcı bilgisayarının tarih ve saat bilgisini alacağız. Bu bilgileri kullanma ihtiyacı hissedebilirsiniz.

Çok basit bir yöntem ile bu bilgileri alacağız. Java dilinde yapılan her durum Java Server Pages ile yapmak mümkündür. Bu örnek ile beraber derslerimizin bir ilkinin gerçekleştireceğiz. Kullanıcı bilgisayarından bir bilgi alacağız (:

Adım adım neler yapacağız açıklayım o halde.

KÜTÜPHANELER

JSP sayfamızda kullanacağımız kütüphaneler 4 tane bunlar;

```
1 <%@ page import="java.io.*,java.util.Locale" %>
2 <%@ page import="java.text.DateFormat,java.util.Date" %>
```

java.io : Giriş, çıkış işlemleri için.

java.util.Locale : Kullanıcı bilgisayarına erişmek için.

java.text.DateFormat : Alacağımız tarih ve saat verisinin yayınlama formatını düzenlemek için.

java.util.Date : Tarih bilgisini almak için.

JSP SAYFASI

index.jsp

```
1 <%@ page import="java.io.*,java.util.Locale" %>
2 <%@ page import="java.text.DateFormat,java.util.Date" %>
3
4 <%
5     Locale yerel = request.getLocale( );
```

```
6     String tarihsaat = DateFormat.getDateTimeInstance(  
7         DateFormat.FULL,  
8         DateFormat.SHORT,  
9         yerel).format(new Date( ));  
10    %>  
11    <html>  
12    <head>  
13        <title>Burak Kutbay Yazılım</title>  
14    </head>  
15    <body>  
16        <center>  
17            <h1>  
18                Tarih , saat: <% out.print(tarihsaat); %>  
19            </h1>  
20        </center>  
21    </body>  
22 </html>
```

Kodu yakından incelerseniz çok fazla yaptığımız bir şey yok. Bir değişken oluşturuyorsunuz bu değişkeni tanımlarken eklediğimiz kütüphanenin özelliklerini tanımlayarak ve ardından bu değişkeni ekranı yazdırmaktan ibaret.

Peki bu değişken'e neler tanımladık biraz açıklamaya çalışayım. DateFormat'ın tarih kısmının hepsini saat kısmını sadece saat ve dakikasını aldık. Full ve Short kısımları ile bunu sağlamış olduk.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/jsp-yerel-tarih-saat-bilgilerinizi-almak.html/>

Java Server Pages Otomatik Yenileme

Jsp Derslerimize kaldığımız yerden devam ediyoruz. Konumuz "Java Server Pages" sayfalarımızda otomatik yenileme nasıl olur buna bakacağız. Oluşturduğunuz sayfaları otomatik yenileme devri artık çoktan geçti bu yenilemeler ajax gibi teknolojiler ile sağlanmaktasa da konumuz **Java Server Pages** olduğu için her ayrıntıyı kaçırmadan blog'uma not etmek istiyorum.

Otomatik yenileme kodları sayfaları belirli bir süre içerisinde tekrar yenileyerek amacınıza hizmet etmektedirler.

Java Server Pages'de otomatik yenileme işlemi aşağıdaki kod bloğu ile olmakta.

```
1 <%response.setHeader("Refresh", 1); %>
```

Otomatik yenileme kodundaki rakam saniye cinsinden otomatik yenileme süresini belirlemektedir. Bu kodun fazla açıklanacak bir durum olduğunu düşünmüyorum.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-server-pages-otomatik-yenileme.html/>

JSP URL Bulma (getRequestURL)

Dersimizin bugünkü konusu Java Server Pages sayfamızın aktif olduğu URL adresini bulmak olacak. Yazının başlığı konuyu tam anlatmamakta maalesef. Düzgün bir başlık bulamadım.

Sayfamızın bulunduğu linki bulmak için

```
1 <% = Request.getRequestURL (%)>
```

yöntemini kullanmamız yeterlidir.

Bu yöntem ile istek edilen ve ya istek alınmış olan URI ve ya URL'leri alarak bir kısayol butonu, fonksiyon yönlendirme işlemleri yapabiliriz.

Bu yöntem bütün web konteynerleri için aynıdır. Bir değişiklik yoktur. İyi kodlamalar.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-server-pages-url-bulma-getrequesturl.html/>

JSP Kullanıcının Dil ve Ülke Bilgisini Bulmak

JSP sayfamızı kullanan kişinin bulunduğu ülke ve dilini nasıl öğreniriz buna bakacağız. Kullanıcılarımızın belirli istatistiklerini tutmak ve ya bana göre en önemlisi bulunduğu konum veya dile göre Java Server Pages sayfalarımızda farklı içerik, farklı sayfa, farklı yönlendirme ve ya isteğimize bağlı ne var ise bunu yapabilmemize imkan sağlayacaktır.

Bu kullanıcı bilgileri ya IP adresinden alarak bir sonuç çıkartırız ya da tarayıcı bilgilerinden ulaşarak temin edebiliriz. Kullanılan sistem isteğinize göre değişebilir. Bu dersimizde IP'den değil Java ve tarayıcı bilgilerinden bu bilgilere erişeceğiz.

java.util.Locale

Locale sınıfını kullanarak erişeceğiz. Bunlar;

- **locale.getLanguage();**
- **locale.getCountry();**
- **locale.getDisplayCountry();**

metodlarını kullanarak erişeceğiz.

Metodları kullanmamız için Java Server Pages sayfamıza eklememiz gereken kütüphaneler ise;

```
1 <%@ page import="java.io.*"%>
2 <%@ page import="java.util.Locale"%>
3 <%@ page import="javax.servlet.http.*"%>
```

Java Server Pages sayfamıza ekledikten sonra yapmamız gereken Localize sınıfının metodlarını kullanmak olacak.

- **locale.getLanguage();**

Kullanıcının yazım dilini ülke kodu olarak gösterir.

- **locale.getCountry();**

Kullanıcının ülkesini kod olarak gösterir.

- **locale.getDisplayCountry();**

Kullanıcın ülkesini gösterir.

Java Server Pages sayfamıza

```
1 <%
```

```
2   Locale kullanıcı = request.getLocale();
3   String dil = kullanıcı.getLanguage();
4   String ulke_kod = kullanıcı.getCountry();
5   String ulke = kullanıcı.getDisplayCountry();
6   out.println(dil);
7   out.println(ulke_kod);
8   out.println(ulke);
9   %>
```

eklememiz yeterli olacaktır.

* Bu konu hakkında sorularınız/yorumlarınız için:
<http://blog.burakkutbay.com/java-server-pages-kullanıcının-dil-ve-ülke-bilgisini-bulmak.html/>

Java Server Pages Hata Sayfası Oluşturma

Bu dersimizde oluşturduğumuz JSP sayfalarımıza hata sayfası oluşturmayı deneyelim. Bu hata sayfası internet sayfalarında görülen HTML sayfası hatası değil. Java Server Pages tarafından oluşan hataları karşımıza çıkarmaya yönelik özel bir sayfa yapacağız.

Bu hata sayfasını kullanıcı tarafına göstermek ne kadar mantıklı bilemiyorum ama hata sayfasını özelleştirerek kullanıcıya anlayacağı bir dil ile iletebilir. Jsp sayfalarımız da beklenmedik bir hata ile karşılaştığımızda da bu hata sayfasını çağırabiliriz. Şimdi örneğimize geçelim.

Bir hata yaratmamız gerek, bu nasıl sağlarız sizce? Bir değişkende saklanmaması gereken değeri saklamak istediğimizde ve ya matematiksel işlemlerde ki yanlış durumlar ile oluşturmak istediğimiz hatayı sağlayabiliriz.

Bu örneğimiz için hemen oluşturulalım.

Bu hatamız **index.jsp** sayfamızda olsun. Hata sayfamız ise **errorpage.jsp** olacak. **index.jsp** sayfasında bir hata olduğu zaman **errorpage.jsp** sayfasına yönlenecek ve gerekli uyarıyı vereceğiz.

```
1   <%@ page errorpage="errorpage.jsp" %>
2
3   <title>blog.burakkutbay.com Jsp Dersi</title>
```

```
4
5 <%
6 int sayi = 1;
7 sayi =sayi/ 0;
8 %>
```

index.jsp sayfamızda **errorpage.jsp** sayfamızı kullanmak için tanıttık.

```
1 <%@ page isErrorpage="true" %>;
2
3 HATA
4
5 yanlış bir işlem gerçekleştirildi.
```

errorpage.jsp sayfamıza bilgilendirme mesajını yazdık. Böylelikle istenmeyen durumlara karşı önlemimizi aldık. Her sayfa için ayrı bir hata sayfası oluşturmaya gerek yok bir hata sayfası yaptıktan sonra gerekli sayfaya import ederek kullanabiliriz.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-server-pages-hata-sayfasi-error-page-olusturma.html/>

JSP Hata Sayfalarını Yönetmek

JSP Dersinde bugün Java Server Pages'te yazmış olduğumuz kodların yanlış yazılması, kullanıcı tarafından gelen hatalı girişler ve ya server yüzünden gelen sorunlarda hata sayfaları ile karşılaştırırız. Bu hata sayfaları Derleyici tarafından tarayıcıda gözüktür. Kötü niyetli kişiler bu hata kodlarından faydalanarak sistemi aşmaya çalışabilirler.

Daha önceden yazdığım Java Server Pages Hata Sayfası Oluşturma yazısında kendi programımızda ki kullanıcı tarafından veri girişi sırasında oluşan hatalarda oluşturduğumuz hata sayfası idi. Bu gün yapacağımız hata sayfaları ise kendi istediğimiz dışında (kontroll ettiğimiz sayfalar dışında) kalan hata uyarılarının hepsini kullanıcıya göstermemek.

UYGULAMA

Çok basit bir şekilde hata sayfalarını istediğimiz düzene sokabiliriz. web.xml sayfası oluşturacağız bu xml sayfasında bir hata olduğunda server'ın ne yapması gerektiğini söyleyeceğiz. Diyeceğiz ki, hata.jsp sayfasına git.

web.xml

```
1 <error-page>
2   <exception-type>java.lang.Throwable</exception-type>
3   <location>/hata.jsp</location>
4 </error-page>
```

Bu xml sayfasında hata tipini Throwable olarak seçtik. Bu tip hatalarda hata.jsp sayfasına yönlenecek.

hata.jsp

```
1 <%@ page isErrorPage="true" import="java.io.*" contentType="text/plain"%>
2 <%
3   exception.printStackTrace(printWriter);
4   out.println("istenmeyen bir durum oluştu lütfen tekrar deneyiniz");
5 %>
```

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-server-pages-hata-sayfalarini-yonetmek.html/>

JSP Hata Mesajı Göstermek

Java Server Pages Dersleri'mizin bu yazısında sayfalarımızda hata mesajları göstermek istediğimizde neler yapmalıyız inceleyeceğiz.

Bu yazıda Java kodu olmayacaktır. Html ve Javascript kullanılarak gerçekleştirilecektir. Bu yapıyı JSP tarafına modülünüze göre değişkenlik gösterebilir.

Bir formumuz olacak bu formda boş geçilemez alanlar olduğunu varsayarak kullanıcıya boş geçilemez olduğunu hatırlatacağız. Alanları tam doldurduğunda ise "Hoşgeldiniz" mesajı vereceğiz. Bu mesaj verme durumunu Javascript ile gerçekleştireceğiz.

Javascript ile yapacağımız kontrol ve uyarı verme durumu basit bir if ile gerçekleştirirken Html formumuzdan Javascript fonksiyonuna formumuzdaki değerleri göndereceğiz. Bu hata mesajı gösterme durumunu Javascript yerine Ajax kullanarak

Basit bir Html Formu ve Javascript fonksiyonu yapacağız.

```
1 <html>
2   <head>
3     <title>Burak Kutbay</title>
4   </head>
5   <body>
6     <html>
7       <head>
8         <script>
9           function kontrol(){
10            if (document.form.adi.value == ""){
11              alert ( "Adınızı Yazınız" );
12              return false;
13            }
14            if (document.form.sifre.value == ""){
15              alert ( "Sifrenizi Giriniz" );
16              return false;
17            }
18            alert ( "Hoşgeldiniz" );
19            return true;
20          }
21        </script>
22      </head>
23      <body>
24        <form name="form" method="post" onsubmit="return kontrol();">
25          <table>
26            <tr><td>Adiniz</td>
27            <td><input type="text" name="adi" value=""></td>
28          </tr>
29          <tr><td>Sifre:</td>
30          <td><input type="password" name="sifre" value=""></td>
31        </tr>
32        <tr>
33          <td></td>
34          <td><input type="submit" name="Submit" value="Giris"></td>
35        </tr>
36      </table>
37    </form>
38  </body>
39 </html>
40 </body>
```

* Bu konu hakkında sorularınız/yorumlarınız için:

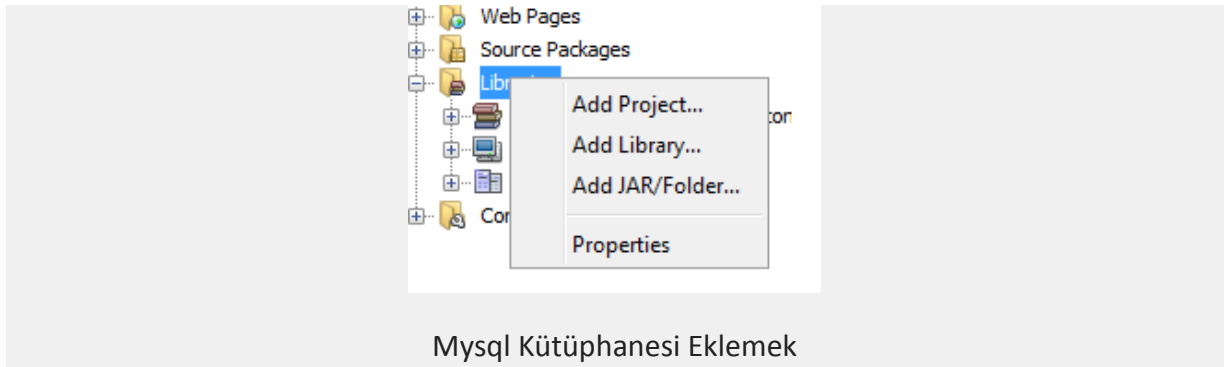
<http://blog.burakkutbay.com/java-server-pages-hata-mesaji-gostermek.html/>

JSP MySql Bağlantısı Kurmak

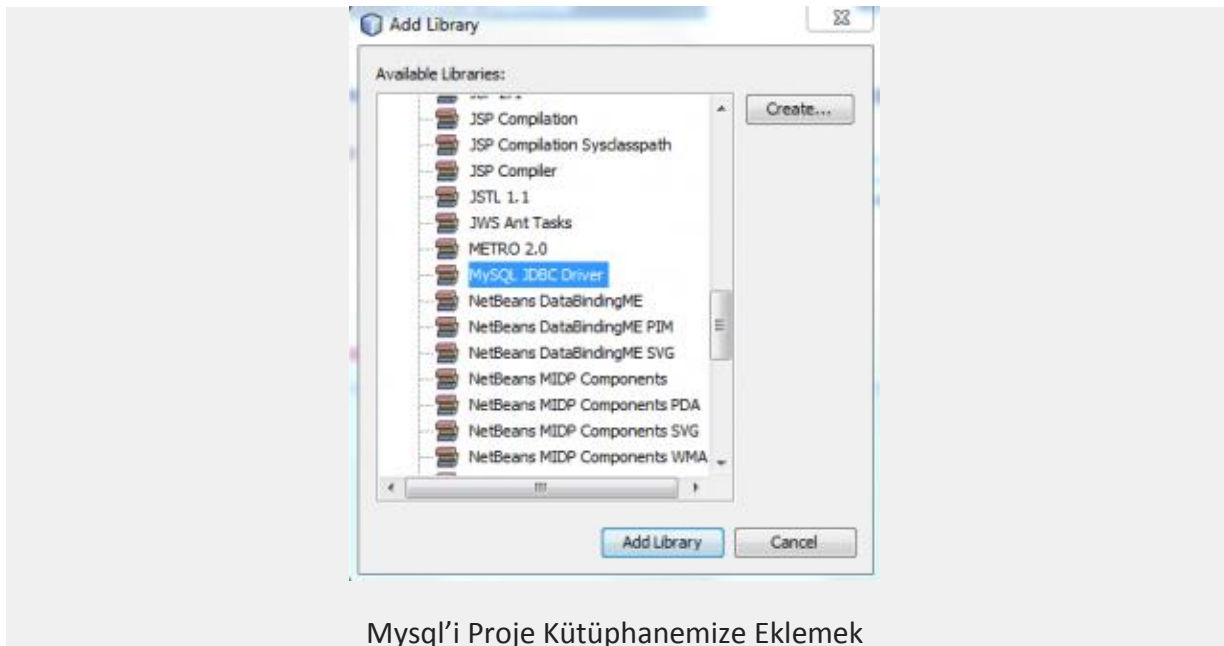
JSP Sayfamızı Veritabanı Yönetim Sistemlerinden biri olan MySql'e nasıl bağlayacağız, onu göreceğiz. Veritabanı ile yapacağımız Jsp sayfaları arasında ki bağ nasıl kurulur? Sorularının cevaplarını arayalım.

JSP derslerimizde NetBeans kullanıyorum. Öncelikle yapmamız gereken şey MySql kütüphanesini kendi projemizin kütüphanesine eklememiz gerekiyor.

Projemizin **Library** bölüme sağ tıkladıktan sonra **Add Library...** seçeneğine tıklayalım



Ardından karşımıza çıkan pencerede **MySQL JDBC Driver** seçeneğini bulduktan sonra Add Library butonuna basarak projemizin kütüphanesine eklemiş bulunuyoruz.



```
1 <% String driver = "com.mysql.jdbc.Driver";
```

```
2 String url = "jdbc:mysql://localhost/veritabaniadi";
3 Connection con = null; %>
```

Projemize MySql kütüphanemizi ekledikten sonra JSP sayfamıza yapmamız gereken işlemlere geçelim.

Dikkat: Veritabanı bağlantıları Bean denilen kavramlarla yapılması güvenlik için daha sağlıklıdır. İleri ki derslerde değineceğim.

Jsp Sayfamızda Veritabanı bağlantısını kuralım yapacaklarımızı kısaca özetleyecek olursak

1. Jsp sayfamızda veritabanına bağlantı için gerekli olan sürücüyü çalıştıracakız
2. Yüklenip, yüklenmediğini kontrol edeceğiz.
3. Mysql Bağlantımızı kuracağız.
4. Jsp ile Mysql bağlantımız kurulup kurulmadığını kontrol edeceğiz

Bu yazmış olduğum her bir adımı **try – catch** bloğuna alıp kontrol ettireceğim. JDBC yüklenmesi sırasında ve ya Mysql bağlantısı sağlanırken oluşan hataları kontrollü bir biçimde görmüş olacağım.

Bağlantı sırasında kullanacağımız tanımlamaları yazalım ardından açıklayalım.

JDBC sürümümüzü, bağlanacağımız veri tabanı yolunu ve bağlantı için gerekli olan con adlı değişkenimizi oluşturduk. Burada dikkat edilmesi gereken nokta “veritabaniadi” . Mysql de oluşturduğunuz “Shema” ismidir bu.

Şimdi JDBC sürümümüzü yükleyelim try-catch blogu içine alarak her durumu kontrol ettirelim.

```
1 <%try {
2 Class.forName(driver);
3 out.println("JDBC surucu basari ile yüklendi.");
4 } catch (Exception e) {
5 out.println("JDBC surucu Yüklenemedi.");
6 System.exit(0);
7 }%>
```

Şimdi Jsp ile Mysql bağlantımızı kurmaya geldi kodlarımızı yazalım ve açıklamamızı yapalım.

```
1 <%try {
2 con = DriverManager.getConnection(url, "root", "root");
3 out.println("Veritabanına bağlanıldı.");
4 } catch (Exception e) {
5 System.out.println("Bağlantı Kurulamadı");
```



```
6 System.exit(0);
7 }%>
```

Burada dikkat edeceğimiz husus con değişkenindeki “root” ve “root” bölümleridir. İlk “root” Mysql veritabanını yöneten kişinin kullanıcı ismidir. İkinci “root” ise şifremizdir. Bu değişiklikleri yaparak veritabanı bağlantımızı oluşturabiliriz.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/jsp-ile-mysql-islemleri-baglantis-kurmak.html/>

Veri Tabanına Kayıt Ekleme (MySql)

[JSP Dersi](#)imizin bu bölümünde JSP ile Veri tabanına kayıt ekleyeceğiz.

JSP Sayfamızda Ad, Soyad ve İl alanlarımız olacak. Kullanıcı bu formu doldurup kaydet butonuna basacak ve bu bilgiler veritabanımıza kayıt olacak.

Jsp ile Kayıt Formu

Bu tasarımın kodları ise

```
1 <head>
2 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
3 <title>Kayıt Formu Burak Kutbay</title>
4 </head>
5
6 <body>
7 Kayıt Formunu Lütfen Doldurunuz.
8
9 <form id="form2" name="form2" method="post" action="kontrol.jsp">
10 <table>
11 <table width="200" border="1">
12 <tr>
13 <th><label for="AD">ADINIZ </label></th>
14 <th><input type="text" name="AD" id="AD" /></th>
15 </tr>
16 <tr>
17 <th><label for="SOYAD">SOYADINIZ</label></th>
18 <th><input type="text" name="SOYAD" id="SOYAD" /></th>
```

```
19 </tr>
20 <tr>
21 <th><label for="il">iLiNiZ</label></th>
22 <th><input type="text" name="il" id="il" /></th>
23 </tr>
24 <tr><th></th><th><input type="submit" name="gonder" id="gonder" value="KAYIT
25 OL" /></th></tr>
26 </table>
27 </form>
28 </body>
29 </html>
```

Bu sayfamızın form elementi “post” yolu ile verilerimizi kontrol.jsp sayfasına gönderecek. Bütün işlem kontrol ettirme sayfasında olacak ve haliyle de java kodlarımız da o sayfada olacak.

Konrol.jsp sayfasında yapılacak işlem eğer verilerimiz veritabanımıza kayıt oldu ise Kaydınız Veritabanımıza Yapıldı. Yapılmadığı bir durumda ise Kaydınız Veritabanımıza yapılmadı yazısı gösterecektir.

Şimdi Kontrol.Jsp sayfamızı yazalım.

Bu sayfamızda dikkat edilecek husus kayıt ol formundaki elementlerin “name” lerini almak olacak.

Jsp Sayfamızda import edeceğimiz kütüphaneleri ekleyelim öncelikle.

```
1 <%@ page import="java.io.*" %>
2 <%@ page language="java" import="java.sql.*"%>
3 <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
```

Bu kodlarla dil tanımlarımızı ve kütüphanelerimizi ekledik. Şimdi yapacağımız işlem ise formdan kontrol.jsp sayfamıza gelen verileri almak olacak.

```
1 String ADI = request.getParameter("ADI");
2 String SOYAD = request.getParameter("SOYAD");
3 String il = request.getParameter("il");
```

Koddan da anlaşıldığı üzere request.getParameter fonksiyonu diğer formdan gelen verileri almaktadır.

Şimdi Veritabanı bağlantımızı yapalım.

```
1 <%
2 String driver = "com.mysql.jdbc.Driver";
```

```

3 String url = "jdbc:mysql://localhost/test?useUnicode=true&characterEncoding=UTF-
4 8";
5 Connection con = null;
6 int id = 0;
7 try {
8 Class.forName(driver);
9 } catch (Exception e) {
10 System.exit(0);
11 }
12 try {
13 con = DriverManager.getConnection(url, "root", "root");
14 } catch (Exception e) {
15 System.exit(0);
16 }
17 %>

```

Buraya kadar neler yaptık bir bakalım.

- Kayıt Formu oluşturduk.
- Kontrol sayfamızdan parametreleri aldık
- Mysql bağlantımızı yaptık.

Şimdi sıra geldi bu verileri veritabanımıza kayıt etmeye.

Verilerimizi veritabanımıza kayıt etmenin bir çok yolu var ama ben **PreparedStatement** denilen yöntemle yapacağım.

Veritabanımıza kayıt işlemi için gerekli kodlarımızı yazalım. Açıklamayı ise ardından.

```

1 <%try {
2 PreparedStatement preStmt;
3 preStmt = con.prepareStatement("INSERT INTO KULLANICI(ADI,SOYAD,il) VALUES
4 (?, ?, ?)");
5 preStmt.setString(1, AD);
6 preStmt.setString(2, SOYAD);
7 preStmt.setString(3, il);
8 preStmt.executeUpdate();
9 preStmt.close();
10 con.close();
11 out.println("Kaydınız Veritabanımıza Yapıldı")
12 }
13 catch (Exception e) {
14 out.println("Kaydınız Veritabanımıza yapılmadı")
15 }
16 %>

```

PreparedStatement deęişkenimizi yazdık ve o deęişkene SQL komutunu yazdık. Values kısmındaki deęişkenleri aynı veri tipine göre sırasına göre yazdık. preparedStatement.executeUpdate() komutu ile veritabanımıza gönderdik ve kapattık. En son olarak ise bağlantımızı kapattık.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/jsp-ile-veri-tabanina-kayit-ekleme-mysql.html/>

Veri Tabanında Kayıt Arama (MySql)

Jsp Derslerimize devam ediyoruz. Bu derse kadar ne yaptık önce bir gözden geçirelim.

[JSP'ye giriş](#) yaptık "Merhaba Dünya" dedik.

[JSP ve Mysql Bağlantısı](#) yaptık.

[Jsp ile Veri Tabanına Kayıt Ekledik.](#)

Sırası ile devam ediyoruz. Jsp ile veritabanımıza kayıt ekledikten sonra bu kayıtlarımızdan bir arama işlemi gerçekleştirmek zorunlu bir hal alacaktır. Onlarca kayıtlı bilgileri "adı" ya da "soyadı" na göre arattırıp o kullanıcının bilgilerini karşımıza getireceğiz.

Arama yaptırmak JSP ile Mysql arasında ki bağlantıda veri tabanımızdan deęer almamızı sağlar.

Bu arama işlemleri'nde aranacak bilginin önemine görea adına, iline veya soyadına göre arattırma yapabiliriz.

Bizim yapacağımız örnekte kullanıcı adına göre arama yapılacak ve bulunan deęere göre kullanıcı adı , soyadı ve ilini bir tablo içerisinde göstereceğiz.

JSP importlarımızı oluşturalım ilk önce.

```
1 <%@ page import="java.io.*" %>
2 <%@ page language="java" import="java.sql.*" %>
3 <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
```

İlk olarak yapacağımız işlem arama.jsp sayfasını tasarlamak. Görünüm resimdeki gibi olacaktır.

Aranacak Ad :

Arama Yap

JSP Arama Formu

Bu arama formumuzun kodlarını oluşturalım.

```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4 <title>Başlıksız Belge</title>
5 </head>
6
7 <body>
8 <form id="form1" name="form1" method="post" action="<b
9 style="color:white;background-color:#880000">arama</b>.<b
10 style="color:black;background-color:#fff66">jsp</b>">
11 <p>
12 <label for="arama_ad">Aranacak Ad</label>
13 :
14 <input type="text" name="arama_ad" id="arama_ad" />
15 <input type="submit" name="ara" id="ara" value="<b
16 style="color:white;background-color:#880000">Arama</b> Yap" />
17 </p>
18 </form>
19 </body>
20 </html>
```

Her zaman yaptığımız gibi JDBC sürücüsü ve MySql veritabanı bağlantımızı oluşturalım.

```
1 <% request.setCharacterEncoding("UTF-8");
2
3 String driver = "com.mysql.jdbc.Driver";
4 String url = "jdbc:mysql://localhost/test?useUnicode=true&characterEncoding=UTF-
5 8";
6 Connection con = null;
7
8 try {
9
10 Class.forName(driver);
11
12 } catch (Exception e) {
13
```

```

14 System.exit(0);
15 }
16 try {
17     con = DriverManager.getConnection(url, "root", "root");
18
19 } catch (Exception e) {
20     System.out.println("Mysql Bilgisi : Bağlantı Kurulamadı" + e + "");
21     System.exit(0);
22 }
23
    %>

```

Bağlantımızı oluşturduk. Formumuzu tasarladık. Arama Yap butonuna basıldığında formumuzun tekrar geleceği yol arama.jsp sayfasına yönlenecek. Jsp get.parameter yordamı ile gelen ad değerini alacağız. formumuzdan gelen “ad” değerini alalım ve bir String değişkenine atalım.

```

1 <%
2     request.setCharacterEncoding("UTF-8");
3     String kullanıcı_adi = request.getParameter("arama_ad");
4     %>

```

Kullanıcı adımızı formumuzdan aldık şimdi yapacağımız işlem Sql komutu ile arama işlemi yaptırmak.

```

1     request.setCharacterEncoding("utf-8");
2     Statement stmt = con.createStatement();
3
4     ResultSet rs = stmt.executeQuery("SELECT * FROM KULLANICI where ADI='" +
        kullanıcı_adi + "'");

```

Sql sorgusu işlemi yaptık. Oluşan değerler rs adlı değişkenimizde bu verileri çekmek için bir yol var. Bu yol while döngüsü içerisinde değer next komutu ile devam eder ve değer bittiğinde çıkar.

```

1     <&out.print("<table>");
2     while (rs.next()) {
3         out.println("<tr><td>" + id + "</td>");
4         out.println("<td>" + rs.getString("ADI") + "</td>");
5         out.println("<td>" + rs.getString("SOYADI") + "</td>");
6         out.println("<td>" + rs.getString("PAROLA") + "</td>");
7         out.println("<td>" + rs.getString("EMALI") + "</td>");

```

```
8 out.println("<td>" + rs.getString("il") + "</td></tr>");}
9 out.print("</table>");%>
```

Çıkan sonuçları bir tablo içerisinde gösterir. Birden fazla kayıt var ise bir alt satıra geçip değerleri yazdırmaya devam eder.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/jsp-ile-veri-tabaninda-kayit-arama-mysql.html/>

JSP Türkçe Karakter Sorunu

[Java Server Pages Dersleri](#)nde bugün çokça karşılaşılan bir sorunu çözeceğiz. Java Server Pages teknolojisi ile dinamik sayfalarımızdan veritabanımıza değerler gönderiyoruz. Bu değerler veritabanında sağlıklı bir şekilde tutulmak zorunda.

Dinamik sayfamızdan veritabanına değerler gönderdiğimizde JSP Türkçe Karakter Sorunu yaşamaktayız.

Türkçe karakterler ş,ğ,ü,ç,ö gibi harfler veritabanımızda değişik semboller ile gözükmektedir.

JSP Türkçe Karakter Sorunu 'nu çözmek için iki farklı yol bulunmaktadır.

Bunlardan ilki

```
1 <%@page contentType="text/html;charset=ISO-8859-9" pageEncoding="ISO-8859-9" %>
```

Bu kod sayfamızın karakter kodunu belirtmektedir.

Bu kod hala sorunumuzu çözümüyor ise ikinci bir yol is şudur.

```
1 request.setCharacterEncoding("utf-8");
```

Bu kod gelen parametrelerin utf-8 kodunda gözükmelerini sağlamakta.

Eğer bu kod da sorunumuzu çözümüyor ise veritabanınızın dil kodunu kontrol etmenizi öneririm.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/jsp-de-turkce-karakter-sorunu-java-server-pages.html/>

Jsp Verileri Dinamik Olarak Sayfalatma

[JSP Dersleri](#)'mizin bu yazısında Java Server Pages üzerinden veritabanında ki tüm verileri ya da istediğiniz belirli bir bölümü dinamik olarak sayfalatma işlemini yapacağız.

Bu derste ile beraber geçen yazıların biraz tekrarını yapmış olacağız ardından verilerimizi sayfa sayfa okumuş olacağız. Yapmak istediğim şeyi biraz daha açıklamak istersem; veritabanımızda 30 tane kayıt var diyelim bu 30 kayıdı bir anda görüntülemek istemiyorum. Bir buton koyup her sayfada bir kayıt göstermek istiyorum.

Bunun için 30 sayfa ve ya kayıt sayımıza göre farklı sayfalar yapmayacağız. Başlıkta da görüldüğü üzere Dinamik'ten kastım bu idi. Bu işlemi nasıl yapacağız biraz düşünelim. Her bir kayıt için sayfa oluşturmayacaksak yapacağımız şey bütün verileri alıp tek tek göstermek. Bunun cevabı ise yazının devamında.

Bu örneği uygulamak için bir senaryo oluşturalım. Bu senaryo da;

- Rehber adı altında bir veritabanımız olacak.
- Bilgi tablosunda,
 1. id (PK,NN)
 2. adi (VARCHAR 45)
 3. soyadi (VARCHAR 45)
 4. numara (VARCHAR 45) kolonları olsun.
- Bu tabloya biraz bilgi girişi yapalım ve uygulamamızda dönelim.

Biraz bilgi girdikten sonra verilerimiz bu şekilde;

	adi	soyad	numara	id
▶	Burak	Kutbay	1111	1
	Baris	Aribumu	22222	2
	Kemal	Atalay	33333	3
*	NULL	NULL	NULL	NULL

Bir Java Server Pages sayfası oluşturarak başlayalım. Bu sayfamız verilere ulaşmak için bağlantı yapmamızı sağlayacak. Bu bağlanma işlemini daha detaylı olarak [JSP MySQL Bağlantısı Kurmak](#) adlı dersimde bulabilirsiniz.

Yapacaklarımız.

1. Veritabanımıza bağlanacağız.
2. Bir verisayacımız olacak. Bu veri sayacımız sayesinde toplamda kaç kayıt olduğunu öğreneceğiz.

3. Bir form oluşturacağız bu form verisayacımızdan gelen deger ile get metodu ile aynı sayfaya yani index.jsp ye yönlenererek bir sonraki kayıta geçecek.
4. Tablo oluşturacağız çektiğimiz verileri tablomuzda göstereceğiz. Formdan gelen verisayacı arttığında Resultset ile gelen değere göre sıradaki verileri gösterecek. Bu method ise Jsp Query String Kullanımı'na benzemektedir. [Bu yazımda](#) detaylı olarak yazmıştım.

Özetle yapılacak işlem veritabanında ki verileri toplayıp bir sıra numarası vererek o sıra numarasına göre oluşturduğumuz formun butonundan gelen değere göre bir artarak sonraki kayıdı göstermek.

Kod sonrasi görüntü şu şekilde olacak;

Tum Kayitlar Tek Tek Listeleniyor...

Adi Burak

Soyadi Kutbay

Numara 1111

id 1

Sonraki Kayit

index.jsp

```
1 <%@ page import="java.sql.*" %>
2 <%@ page import="java.io.*" %>
3 <HTML>
4 <HEAD>
5 <TITLE>Burak Kutbay Yazılım Mühendisi </TITLE>
6 </HEAD>
7 <br>
8 <br><center>Tum Kayitlar Tek Tek Listeleniyor...</center>
9 <% int verisayac = 0;
10 Connection connection = null;
11 String connectionURL = "jdbc:mysql://localhost:3306/rehber";
12 ResultSet rs = null;
13 %>
14 <center>
15 <FORM ACTION="index.jsp" METHOD="get"><%
16 if (request.getParameter("verigoster") != null) {
17 verisayac = Integer.parseInt(request.getParameter("verigoster"));
18 }
19 Statement statement = null;
20 try {
```

```

21     Class.forName("com.mysql.jdbc.Driver").newInstance();
22     connection = DriverManager.getConnection(connectionURL, "root", "root");
23     statement = connection.createStatement();
24     rs = statement.executeQuery("select * from bilgi");
25     for (int i = 0; i < verisayac; i++) {
26         rs.next();
27     }
28     if (!rs.next()) {
29         %>
30         <%
31         out.println("Kayit Sonu");
32     } else {
33         %>
34         <TABLE>
35             <TR><TH>Adi</TH><TD> <%= rs.getString(1)%> </TD></tr>
36             <TR><TH>Soyadi</TH><TD> <%= rs.getString(2)%> </TD></tr>
37             <TR><TH>Numara</TH><TD> <%= rs.getString(3)%> </TD></tr>
38             <TR><TH>id</TH><TD> <%= rs.getInt(4)%> </TD></tr>
39         </TR>
40     </TABLE>
41     <BR>
42     <INPUT TYPE="hidden" NAME="verigoster" VALUE="<%=verisayac + 1%>">
43     <INPUT TYPE="submit" VALUE="Sonraki Kayit">
44 </FORM>
45 <%
46 }
47 }
48 catch (Exception ex) {
49 out.println("Veritabanına baglanilamadi.");
50 }
51 %>
52 </FORM>
53 </center>
54 </body>
55 </html>

```

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-server-pages-verileri-dinamik-olarak-sayfalatma.html/>

JSP 'de XML Veri Gösterimi

Java Server Pages Ders'leri devam ediyor. Şimdiye kadar JSP (Java Server Pages) hakkında bir bilginiz yok ise buradan fikir sahibi olabilirsiniz. Web projelerimizde "veriler" yer alır. Veriler belirli bir yerde saklanır bu veriler nerede saklanırsa saklansın bizim ona ulaşmamız gerekiyorsa muhakkak almalıyız. Daha önceki derslerde verilerimizi bir veri tabanından alarak işlemler yapmıştık. (1,2)

KISACA XML

Bu sefer verilerimizi XML'den alacağız ve Java Server Pages sayfamızda listeyeleceğiz, görüntüleyeceğiz. XML'den kısaca bahsedelim ki neyle uğraştığımızı bilelim. XML denilen teknoloji tarayıcı ile server arasında verileri taşıyan aracı bir sistem diyebiliriz. XML server yükünü azaltır. Belirli bir formatı vardır. Bu formata göre XML' "parse" ederek verilerin bir bölümünü ya da hepsini kullanabiliriz, gösterebiliriz.

UYGULAMA

Elimizde bir XML dosyası olduğunuz varsayalım. Bu XML dosyasından verileri çekmemiz gerekiyor ve bu çekilen verileri JSP sayfamızda göstereceğiz.

1) XML SAYFASI OLUŞTURMA

```
1 <uyelist>
2   <uye>
3     <ad>Burak</ad>
4     <soyad>Kutbay</soyad>
5     <yas>24</yas>
6   </uye>
7   <uye>
8     <ad>Karub</ad>
9     <soyad>Yatba</soyad>
10    <yas>26</yas>
11  </uye>
12 </uyelist>
```

2) JAVA SERVER PAGES SAYFASI

index.jsp

```
1 <%@ page contentType="text/html"%>
2 <%@ page import="javax.xml.parsers.DocumentBuilderFactory,
3 javax.xml.parsers.DocumentBuilder,org.w3c.dom.*"
4 %>
5 <%
6   DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
```

```

7  DocumentBuilder db = dbf.newDocumentBuilder();
8  Document doc = db.parse("http://localhost:8080/uye.xml");
9
10 NodeList node1= doc.getElementsByTagName("ad");
11 NodeList node2= doc.getElementsByTagName("soyad");
12 NodeList node3= doc.getElementsByTagName("yas");
13 %>
14 <html>
15 <body><center>
16 <table>
17 <tr>
18 <td font-weight:bold;">ADI</td>
19 <td font-weight:bold;">SOYADI</td>
20 <td font-weight:bold;">YASI</td>
21 </tr>
22 <tr>
23 <%
24 for(int i=0;i<2;i++)
25 {
26 %>
27 <td><%= node1.item(i).getFirstChild().getNodeValue() %></td>
28 <td><%= node2.item(i).getFirstChild().getNodeValue() %></td>
29 <td><%= nnode3.item(i).getFirstChild().getNodeValue() %></td>
30 </tr>
31 <%
32 }
33 %>
34 </table>
35 </center>
36 </body>
37 </html>

```

3) NELER YAPTIK?

Bu kodları satır satır açıklayalım.

2-3. satırlarda XML verilerini okumak ve bulunduğu konumdan programımızın içine göndermek için gerekli kütüphaneleri Jsp sayfamıza ekledik.

6-7. satırlarda ise gerekli kütüphanelerden nesne türetiyoruz. XML dökümanını açmak ve okumak için o nesnenin yetenekleri kullanmamız için gerekli kod bloğu.

8. satırda XML dosyamızın yolunu JSP sayfamıza söylüyoruz.

10-12 satırlar arası ise XML başlıklarını gerekli NodeList değişkenlerine atıyoruz. Nodelist bir "array" diyebiliriz.

27-29 satırlarda bir döngü ile Nodelist'te ki XML verilerini bir tablo halinde gösteriyoruz.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-server-pages-jsp-de-xml-veri-gosterimi.html/>

Java Server Pages Thread Safe

Thread Nedir?

Bu konuyu uzunca işlemek gerek ama kısaca bahsetmek gerekirse Thread dediğimiz kavram yaptığımız programların bir bütün olarak değil bölümlü olarak çalıştırmasını sağlamaktayız. Bir Thread ya da birden fazla Thread ile programlarımız işleyebilmekteyiz.

Bu konu ile blog’umda iki yazı yazmıştım.

- [Java: Thread Metodları](#)
- [Java: Multithreading](#)

Thread Safe Nedir?

Bu kavramda aslında tek başına incelememiz gerekmekte. Hangi kod bloklarımızı Thread Safe yapmalıyız nerelerde ihtiyaç duyulur bu soruların cevaplarını başka bir yazıda detaylıca açıklayacağım. Kısaca Thread Safe yaptığımız Thread’lerimizi güvenli olarak çalışmasını sağlamaktadır. Java Server Pages’te Thread’lerimizin aynı anda çalışması durumunda bu Thread’i bir sıraya almamız gerekmektedir(Bu durum sadece JSP için geçerli değildir. Herhangi bir yazılım dilinde de geçerlidir.). Aynı anda çalışan birden fazla Thread var ise sorun çıkartmaması, çakışmaması için kullanılan bir yöntemdir. Birden fazla aynı anda aynı Thread’e gelen istekleri sıraya alarak kontrollü bir yapı sağlamaktayız. Sayfamıza birden fazla gelen istekleri kontrol etmemize olanak sağlamaktadır.

Java Server Pages’te bu Thread Safe’yi sağlamak için kullanacağımız kod tek satırlık bir işlemdir.

```
1 <%@ page isThreadSafe="false" %>
```

Bu kod bloku ile Thread Safe’nin özelliğini True veya False olarak değiştirip kontrol edebilmekteyiz. Bu ifade False olursa JSP’ye gelen istekler kontrol altına alınmış demektir. Sayfamızdan isteklerimiz birer birer olarak gidecektir.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-server-pages-thread-safe.html/>

Java Server Pages İle Java Beans Kullanımı

Java Server Pages Derslerine **Java Server Pages**'te Java Bean kullanımı ile devam ediyoruz. Java Bean denilen kavram uzaktan bakıldığında çok "afilli" birşeymiş gibi görünsede bu Java Bean bizim Java'dan alışık olduğumuz sınıflardır aslında.

Bu sınıflar'ın **Java Server Pages**'te ki özelliği basitçe bu **Java Bean**'lara kolayca erişmemizi sağlamaktadır. Bean'ların paketleri farklı olmalıdır. Get ve Set'ler ile iletişimi sağlayabilmekteyiz.

Java Bean'ler JSP'de taglar yardımı ile kolayca işlemler yapılabilen. Basit bir Java Bean, yani Java sınıfı nasıl olur ona bir bakalım ardından ise Java Server Pages'te Java Bean'lar nasıl kullanılmakta inceleyelim.

Basit bir Java Bean oluşturmanın bazı şartları var. Bu şartın bana göre en önemlisi değişkenler private olmak zorunda. Yukarıda da söylediğim gibi Get-Set kesinlikle olmalı.

Şimdi bir Java Bean oluşturalım.

```
1 package Bean;
2 public class Bean_deneme {
3
4     private String isim=new String();
5
6     public String getisim() {
7         return isim;
8     }
9     public void setisim(String isim) {
10        this.isim = isim;
11    }
12 }
```

Java Server Pages'te kullanacağımız Java Bean'imizi oluşturduk. Bu kod blogunda neler yaptık biraz bakalım. Java'da bir sınıf oluştururken yaptığımız işlemlerin aynısı aslında. Get ve Set metodlarının da oluşturduk ve isim adlı değişkenimiz private olarak kaydettik.

```
1 <html>
2 <title>Burak KUTBAY</title>
3 <head>
4 </head>
5 <body>
6 <jsp:useBean id="Bean_deneme" class="Bean.Bean_deneme" scope="session" >
7 <jsp:setProperty name="Bean_deneme" property="isim" value=" Burak" />
```

```
8 </jsp:useBean>
9
10 <h1> <jsp:getProperty name="Bean_deneme" property="isim" /></h1>
11 </body>
12 </html>
```

Java Server Pages sayfamızda yaptıklarımıza şimdi bakalım. Jsp Tag kullanarak Bean'imizi çağırdık. SetProperty ile isim değişkenimize bir değer aktardık ve ardından getPoroperty ile isim adlı değişkenin değerini çekerek kullanıcıya gösteriyoruz.

Java Server Pages'te Java Bean'nın kullanımı bu şekilde olmakta. Sorularınız veya eklemek istedikleriniz için yorum yapmanız yeterli.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-server-pages-ile-javabeans-kullanimi-nedir.html/>

ORM Object Relational Mapping Nedir?

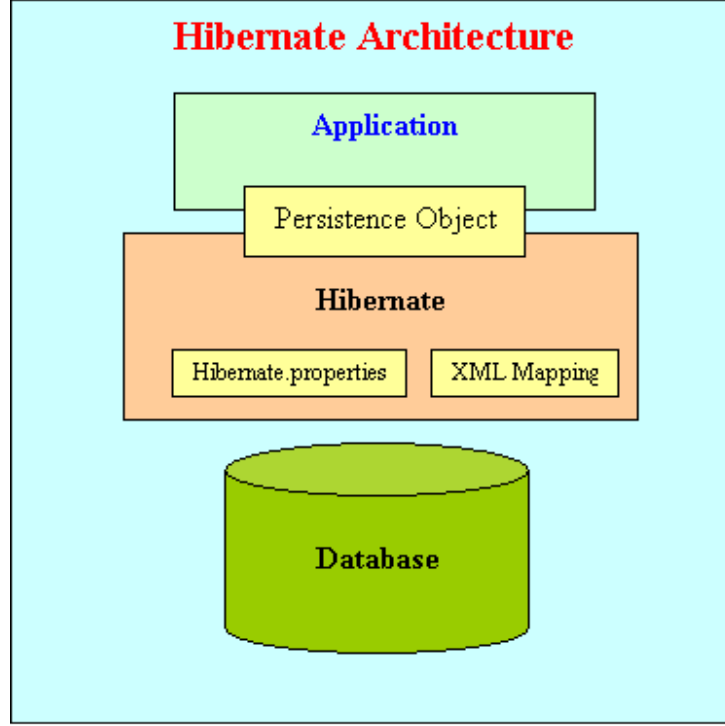
Bu yazıda ORM nedir Nasıl bir çalışma mantığı var bu konuları işledikten sonra avantajlarını sizinle paylaşacağım.

ORM ve ya Object Relational Mapping denilen kavram aslında bizim kod bloklarımız ile veritabanımız arasında oluşan bir köprüdür. Bu tanım ORM'nin en basite indirgenmiş hali.

Şimdi bu tanımdan sonra biraz işi detaylandırarak anlatmaya çalışacağım.

Veritabanı derken, ilişkisel veritabanından bahsediyorum. İlişkisel Veritabanımız, bizim katmanlara ayırdığımız programımız arasında (OOP) veri alışverişini sağlar. ORM veritabanımızda ki tablolarımızı class'lara çevirir. Bu class yapısı ile bizim programlama işleminde ki veritabanı bölümlerinde daha az zaman harcamamıza olanak sağlamakta. ORM ile işlemlerin daha kısılması ve daha düzenli bir şekilde devam etmesini sağlamaktadır.

Sql ile uğraşmamıza gerek kalmaz ORM sayesinde. Hangi ORM Çeşidini kullanıyorsak o ORM sistemine göre kullanım sistemi ile işleme devam edebiliriz. Kullanımı bir programcı için zor değildir.



ORM bizim veritabanımızı bir harita olarak önümüze koyar. Karmaşık bir veritabanı yapısını daha rahat görmemize olanak sağlamaktadır.

Örnek olarak Hibernate Java'da kullanılan bir ORM dir. Hibernate'nin sistemine bir bakalım.

Object Relational Mapping veritabanına sağdik bir bir parça değildir. Yani bağımsızdır. Bir veritabanınız için bir den fazla ORM kullanabilirsiniz. ORM uygulama katmanımızın altında bulunmaktadır. Bu durum tam tersi de olabilmektedir. Projenizde MS-SQL kullanıyorken karar değiştirdiniz diyelim MySql kullanmaya başladınız. Bu durum sizin ORM nize bir sıkıntıya neden olmaz aynı şekilde kaldığınız yerden devam edebilirsiniz.

ORM ile beraber programımız daha da nesneye yönelimli bir hal almakta, veritabanı modelleme kısmı için imkan sağlamaktadır. ORM'ler otomatik kod oluşturarak bize sadece gerekli düzenlemeleri yapmak kalmakta.

ORM ile beraber class'lar oluşuyor demiştik bu class lar,

- Bağlanma
- Ekleme
- Düzeltme
- Silme
- Kapatma
- Kontrol

Şeklinde olmaktadır.

Herşey elinizin altında bu sistem yazmayı kolaylaştırdığı gibi performansı maalesef olumsuz yönde etkilemektedir. Object Relational Mapping tüm programlama dillerinde mevcuttur.

JAVA için kullanılan ORM'ler

- Hibernate
- JPA
- EclipseLink
- Apache Cayenne

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/orm-object-relational-mapping-nedir-kullanimi.html/>

JSP'de Üye Kayıt Formu Örneği

[Java Server Pages Dersimizin](#) bu yazısında örnek yapacağız.

Üye olacak kişinin bilgilerini alıp veritabanımıza kayıt edeceğiz. Satır satır açıklamayacağım çünkü [önceki konularda](#) detaylı olarak anlatmaya çalışmışım.

index.jsp sayfası

```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2
3 <!DOCTYPE html>
4
5 <html>
6
7 <head>
8
9 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
10
11 <title>JSP Page</title>
12
13 </head>
14
15 <body><center>
16
```

```
17 <br>
18
19 <br>
20
21 <h3><b style="color:black;background-color:#a0fff">Üye</b> Kayıt <b
22 style="color:black;background-color:#ff9999">Formu</b></h3>
23
24 <form action="Kaydet.jsp" method="POST">
25
26 <table>
27
28 <tr>
29
30 <td>TC No:</td>
31
32 <td><input type="text" name="tcno" /></td>
33
34 </tr>
35
36 <tr>
37
38 <td>Ad:</td>
39
40 <td><input type="text" name="ad" /></td>
41
42 </tr>
43
44 <tr>
45
46 <td>Soyad:</td>
47
48 <td><input type="text" name="soyad" /></td>
49
50 </tr>
51
52 <tr>
53
54 <td>Adres:</td>
55
56 <td><textarea name="adres" rows="3" cols="16"></textarea></td>
57
58 </tr>
59
60 <tr>
61
62 <td>İli:</td>
63
```

```
64 <td><input type="text" name="il"/></td>
65
66 </tr>
67
68 <tr>
69
70 <td>ilçe:</td>
71
72 <td><input type="text" name="ilce"/></td>
73
74 </tr>
75
76 <tr>
77
78 <td><input type="submit" value="Kaydet" /></td>
79
80 </tr>
81
82 </table>
83
84 </form></center>
85
86 </body>
87
</html>
```

Kaydet.jsp Sayfası

```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2
3 <%@page import="java.sql.Connection, java.sql.PreparedStatement" %>
4
5 <%@page import="java.sql.SQLException,java.sql.DriverManager" %>
6
7 &nbsp;
8
9 <html>
10
11 <head>
12
13 <title>JSP Page</title>
14
15 </head>
16
17 <body>
18
19 <%!
```

```

20
21 Connection baglanti=null;
22
23 PreparedStatement ifade=null;
24
25 String tcNo,ad,soyad,il,ilce,adres;
26
27 %>
28
29 <%
30
31 try{
32
33 tcNo=request.getParameter("tcno");
34
35 ad=request.getParameter("ad");
36
37 soyad=request.getParameter("soyad");
38
39 adres=request.getParameter("adres");
40
41 il=request.getParameter("il");
42
43 ilce=request.getParameter("ilce");
44
45 try{
46
47 Class.forName("com.mysql.jdbc.Driver");
48
49 }catch(ClassNotFoundException e){
50
51 out.println("Jdbc Yüklenemedi");
52
53 }
54
55 baglanti=DriverManager.getConnection("jdbc:mysql://localhost/uyelik sistemi?useU
56 nicode=true&characterEncoding=UTF-8", "root", "root");
57
58 ifade=baglanti.prepareStatement("insert into
59 uye kayit(uyetc, uye ad, uye soyad, uye adres, uye il, uye ilce)
60 values(?, ?, ?, ?, ?, ?) ");
61
62 ifade.setString(1, tcNo);
63
64 ifade.setString(2, ad);
65
66 ifade.setString(3, soyad);

```

```
67
68 ifade.setString(4, adres);
69
70 ifade.setString(5, il);
71
72 ifade.setString(6, ilce);
73
74 ifade.executeUpdate();
75
76 }catch(SQLException e){
77
78 out.println("Hata");
79
80 }finally{
81
82 ifade.close();
83
84 baglanti.close();
85
86 out.println("Kayıt Yapıldı");
87
88 }
89
90 %>
91
92 </body>
93
94 </html>
```

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/jspde-uye-kayit-formu-ornegi-java-server-pages.html/>

JSP Sayfa Sayacı Yapımı (JSP Örnek)

Java Server Pages Derslerinde bu yazıda bir örnek yaparak devam edeceğim. Java Server Pages sayfalarımızda kullanmak üzere ziyaretçi sayacı yapacağız. Bu sayac o sayfanın yenilenmesine göre artmaktadır. Her kullanıcının o sayfaya girdiği sayıyı saymaktadır.

Sayaç ile veri tutarak istenilen değeri arttırıp azaltabiliriz. Bu örnekteki amaç sadece sayaç yapmak değil bir önceki cümle de bahsettiğim gibi veri saklama işlemini pekiştirmek.

Bu örnek ile öğrendiklerimizi pekiştireceğiz. Bu örnekte yapacağımız kodlamaların neredeyse tamamını önceki **Java Server Pages** Derslerinde aktarmıştım. Üst menüden ulaşabilirsiniz. **Java Server Pages** Örneğimiz ise burada. Sorularınızı **yorum** yaparak iletebilirsiniz.

sayac.jsp

```
1 <Html>
2   <Head>
3     <Title>Burak Kutbay</Title>
4   </Head>
5   <Body>
6     <%
7       Integer sayac = (Integer)application.getAttribute("sayac");
8       if (sayac == null) {
9         sayac = 1;
10      } else {
11        sayac= sayac + 1;
12      }
13      application.setAttribute("sayac", sayac);
14    %>
15    <br>
16    Bu sayfaya <%=sayac%> kez geldiniz
17  </br>
18 </Body>
19 </Html>
```

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-server-pages-jsp-sayfa-sayac-yapimi.html/>

JSP Java Bean Örneği

[Java Server Pages Dersleri](#)'ne devam ediyoruz.

Önceki JSP derslerimiz [Java Server Pages İle Java Beans Kullanımı](#) hakkında temel bilgi edinip nasıl kullanıldığına bakmıştık. Bu dersimizde ise **Java Bean**'leri kullanarak Basit bir **Java Server Pages Örnek** yapacağız. Java Server Pages ile programlama yaparken **Java Bean**'leri kullanmalıyız nedenlerinden zaten bahsetmiştik. Fazla detaya inmeden 40 küsür ders'te gördüklerimizi pekiştirerek **Java Server Pages Java Bean Örneği** yapalım.

Örneğimiz basit bir Vize – Final sorgulaması yapan bir program olacak.

İlk olarak Java Bean'imizi oluşturalım. (İşin kısası Java Class yapacağız.)

hesapla.java

```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6
7  /**
8   *
9   * @author http://blog.burakcutbay.com
10  */
11 package bean;
12
13 public class hesapla{
14
15     private String ad="";
16     private int vize=0;
17     private int finall=0;
18     private int sonuc=0;
19
20     public void setAd(String ad){
21         this.ad=ad;
22     }
23
24     public void setVize(int vize){
25         this.vize=vize;
26     }
27
28     public void setFinall(int finall){
29         this.finall=finall;
30     }
31
32     public void setSonuc(int sonuc){
33         this.sonuc=sonuc;
34     }
35
36     public String getAd(){
37         return (ad);
38     }
39
40     public int getVize(){
41         return (vize);
42     }
43
44     public int getFinall(){
45         return (finall);
46     }
47 }
```

```

47
48     public double getSonuc(){
49         return (sonuc);
50     }
51
52     public double getHesapla(){
53
54         sonuc=(int) ((vize*0.4)+(final*0.6));
55         return sonuc;
56     }
57 }

```

Getter ve Setter metodları bulunan bir Java Class'ı yazdık. Java Bean olarak kullanacağımız bu classımız hazır. Şimdi ise index sayfamız yapacağız.

Bu sayfamızda bir form olacak ve formda kullanıcının adı, vize ve final bilgilerini alacağız.

index.jsp

```

1  <%--
2  Document : index
3  Created on : 26.Ara.2013, 18:53:11
4  Author : http://blog.burakkutbay.com
5  --%>
6
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <!DOCTYPE html>
9  <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>burakkutbay.com</title>
13 </head>
14 <body>
15 <h1>Vize Final Hesapla</h1>
16 <form name="Form1" method="post" action="sonuc.jsp">
17 <table>
18 <tr>
19 <td>Adın</td>
20 <td><input type="text" name="ad"> </td>
21 </tr>
22 <tr>
23 <td>Vizen</td>
24 <td><input type="text" name="vize"> </td>
25 </tr>
26 <tr>
27 <td>Final</td>
28 <td><input type="text" name="final"> </td>

```



```
29     </tr>
30     <tr>
31         <input type="submit" name="Gonder" value="Hesapla">
32     </tr>
33 </table>
34 </form>
35 </body>
36 </html>
```

Anasayfamız hazır ve bu anasayfamızdan verilerimiz sonuc.jsp sayfasına gidecek ve o sayfada girilen değerler Java Bean'imimize ulaşacak ve sonuç işlemini görebileceğiz.

sonuc.jsp

```
1  <%--
2   Document   : sonuc
3   Created on : 26.Ara.2013, 20:37:44
4   Author    : http://blog.burakkutbay.com
5   --%>
6
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <!DOCTYPE html>
9  <html>
10 <head>
11   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12   <title>burakkutbay.com</title>
13 </head>
14 <body>
15   <h1>Vize Final Hesapla</h1>
16   <jsp:useBean id="hesapla" class="bean.hesapla" scope="request"/>
17   <jsp:setProperty name="hesapla" property="*" />
18   Sayın: <jsp:getProperty name="hesapla" property="ad" /><br>
19   Vize: <jsp:getProperty name="hesapla" property="vize" /><br>
20   Finalin: <jsp:getProperty name="hesapla" property="finall" /><br>
21   Sonuç: <jsp:getProperty name="hesapla" property="hesapla" />
22
23 </body>
24 </html>
```

Java Server Pages ile Java Bean kullanımı hakkında örneğimiz bu kadardı. Sorularınız için yorum yapmanız yeterli.

* Bu konu hakkında sorularınız/yorumlarınız için:

<http://blog.burakkutbay.com/java-server-pages-bean-ornek.html/>