

# URL Seçim Listeleri

**Burak Bayramlı**

Bu belge,  $\text{\LaTeX}$  ile üretilmiştir

# Seimler

Kurumsal uygulamalarda önyüzde oldukça çok ortaya çıkan bir ihtiyaç (pattern), veri tabanından bir listeyi göstermek, ve bu listeye tıklanınca gösterilen kalemin detaylarını vermektir.

## Obje takibi

Eski Web mimarilerinde bir URL listesi oluşturulurken tanım alanı URL tanımı olarak gösterilir, URL href bağlantısına da bir ID konularak “okuma aksiyonuna” bu bilgi iletilmiş olurdu.

Okuma aksiyonu nesnenin tamamını o ID’yi kullanarak okur, bir sonraki sayfaya yönlendirirdi.

# Obje takibi

Web Beans teknolojisi bu genel kalıp için yeni `@DataModelSelection` annotation'ı sunuyor. Bu teknoloji ile programcının sayfa üzerinde id'ler ile uğraşmasına gerek yok.

# Bean

Daha önce @DataModel annotation'ını görmüştük. Bu annotation ile bir listeyi bean'den XHTML sayfamıza döndürmemiz mümkün oluyordu.

```
@DataModel  
List<XYZ> xyzList;
```

# Bean

Eğer bu listedeki elemanların “seçilebilir” olmasını istiyorsak, o zaman seçim sonucunu ID değil, obje olarak alabiliriz, bunun için listedeki objelerden bir tekini tutacak bir obje referansını `@Out` ve `@DataModelSelection` ile işaretlemek yeterli.

```
@DataModelSelection  
@Out(required=false)  
private XYZ selectedObject;
```

# Bean

Perde arkasına ne oluyor? Web Beans, `@DataModel` ile listesi verilen şeye tıklanınca, o tıklanan objenin `@DataModelSelection` ile işaretlenen referansa “enjekte” edilmesini sağlıyor. Bizim tek yapmamız gereken işaretleri gereken yerleri koymaktı, gerisini tamamen WB halletti.

# Sayfa

Bu işlemin sayfa ile bağlantısı şöyle yapılır. Eğer bir liste -> detay sayfa ikilisinden bahsediyorsak, genellikle bean üzerindeki bir “seçim metodu” listeden `<h:commandLink>` ile çağırılır. Bu link üzerinde parametre belirtmeye gerek yoktur, çünkü dediğimiz gibi seçilen obje perde arkasından bean’imiz üzerine enjekte ediliyor.

# Sayfa

Bu durumda seçim metotuna ne yapmak kalır? Seçilen obje tamamen geliyor olsa bile, belki bazı ek bilgiler Hibernate üzerinden alınacaktır, vs. Bu işlemler seçim metotunda yapılabilir. Ayrıca, detay sayfasına yönlendirilmek için seçim metodu bir XHTML ismi döndürebilir.

# Sayfa

## XHTML

```
<h:dataTable ... >  
  <h:column>...</h:column>  
  <h:column>  
    <h:commandLink value=".." action="#{xyzService.select}"/>  
  </h:column>  
</h:dataTable>
```

## Bean

```
public String select() {  
    ...  
    return "/detail.xhtml";  
}
```

# Sayfa

Yönlendirilen detay sayfasında artık @Out ile “deenjekte” (outject) edilmiş objeye *ismini kullanarak* erişebiliyoruz. Meselâ üstteki örnekte `selectedObject` ile deenjekte edilmiş nesnenin detaylarını

```
<h:outputText value="#{selectedObject.prop2}"/>
```

ile göstermemiz mümkündür.

# Görevler

- `home.xhtml` ve `garage.xhtml`'den oluşan bir uygulama yazalım. Bu uygulamada bir garaj listesini `GarageServiceBean`'den `home.xhtml`'e döndürelim.
- Bu liste tıklanabilir URL listesi olacak, garaj tanımı yanında tıklanabilir “edit” linkleri olacak.
- Bu linklerden herhangi birine tıklanınca bir sonraki ekran olan `garage.xhtml` içinde o seçilen garajın ID'si ve

tanımı gösterilecek.

# Son