

Spring MVC ile Form İşlemleri



Spring Form Tag Kütüphanesi

- Spring'in sağladığı bir **form tag kütüphanesi** mevcuttur
- Bu kütüphane ile **form elemanları ile model verisi arasında binding** yapılabilir
- Bind edilen model verisi ile ilgili **hataları ekranda göstermeyi** de sağlar
- **JSP sayfaları** içerisinde kullanılabilir
- Her HTML tag'ine karşılık gelen bir **form tag'i** mevcuttur

Spring Form Tag

Kütüphanesi: form Tag

```
<%@ taglib prefix="form"  
uri="http://www.springframework.org/tags/form" %>
```

```
<form:form modelAttribute="vet">  
  <table>
```

```
    ...  
    <tr>
```

```
      <td>First Name:</td>
```

```
      <td>
```

```
        <form:input path="firstName" />
```

```
        <form:errors path="firstName" />
```

```
      </td>
```

```
    </tr>
```

```
    ...
```

```
  </table>
```

```
  <form:errors path="*" />
```


```
</form:form>
```

Inner tag'ler için bir "binding path" oluşturur
Bütün diğer tag'ler form tag'i içerisinde yer alırlar

Errors tag'inin hataları gösterebilmesi için dönen model verisi içerisinde **BindingResult** nesnesi olmalıdır

@ModelAttribute ve HTML Form'lar

```
@ModelAttribute("vet")  
private Vet initFormModel() {  
    Vet vet = new Vet();  
    return vet;  
}
```



Metot düzeyinde kullanıldığında Form model object oluşturulması sağlanır

@ModelAttribute ile işaretlenmiş metotlar, **@RequestMapping** ile eşlenen metot öncesinde mutlaka çalıştırılırlar

Daha sonra bu değerler, **@ModelAttribute** ile işaretlenen **metot parametrelerine** de aktarılabilirler

@ModelAttribute ve HTML Form'lar

Model verisinin controller'a iletilmesi sağlanır

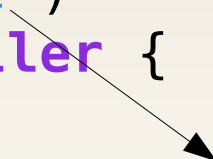
```
@RequestMapping(value="/saveVet",method=RequestMethod.POST)
public ModelAndView handleSubmit(@ModelAttribute("vet") Vet
vet, BindingResult bindingResult) {

    if(bindingResult.hasErrors()) {
        return "saveVet"
    }
    ...
    petClinicService.saveVet(vet);
    ...
}
```

Form ile submit edilen veri model'e aktarılırken oluşabilecek hatalar vs BindingResult üzerinden tespit edilebilir

HttpSession'da Saklanması

```
@Controller
@SessionAttributes("vet")
public class VetController {
    // ...
}
```

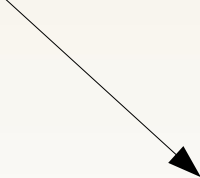
 @SessionAttributes ModelAttribute nesnelerinin HTTP Session'da saklanmasını sağlar. Birden fazla ModelAttribute değişkeni belirtilebilir

```
@Controller
@SessionAttributes(types={Vet.class})
public class VetController {
    // ...
}
```

ModelAttribute nesnelerinin tipi de belirtilebilir
Bu durumda bu tip ile eşleşen bütün model attribute nesneleri HttpSession'da saklanacaktır

Model Verisinin HttpSession'dan Silinmesi

```
@RequestMapping(value="/saveVet",method=RequestMethod.POST)
public String handleSubmit(@ModelAttribute("vet")
    Vet vet, SessionStatus status) {
    ...
    status.setComplete();
    ...
}
```



Http session'da tutulan ModelAttribute nesnesinin kullanıcı senaryosu sonunda Session'dan çıkarılmasını sağlar

Diğer Tag'ler

- `<form:radio button path="sex" value="M" />`
- `<form:input path="firstName" />`
- `<form:textarea path="comment" rows="3" cols="20" />`
- `<form:hidden path="secret" />`
- `<form:password path="password" />`
- `<form:select path="format">`
 - `<form:option value="DVD" />`
 - `<form:option value="MPEG" />``</form:select>`
- `<form:checkbox`
`path="preferences.receiveNewsletter"`
`value="true" />`

MVC ile Form Örneği

1

Insert title here x

localhost:8080/petclinic/mvc/saveVet

First Name:

Last Name:

Create

2

Insert title here x

localhost:8080/petclinic/mvc/saveVet

First Name: may not be empty

Last Name: may not be empty

Create

3

Insert title here x

localhost:8080/petclinic/mvc/saveVet

First Name:

Last Name:

Create

Vet created

MVC ile Form Örneği

saveVet.jsp

```
<form:form modelAttribute="vet" method="post">
    First Name:<form:input path="firstName" />
    <form:errors path="firstName" cssStyle="color:red" />
<br>

    Last Name:<form:input path="lastName" />
    <form:errors path="lastName" cssStyle="color:red"/>
<br>

    <form:button name="submit">Create</form:button>
</form:form>

<c:if test="${not empty message}">
    <div style="color:blue">${message}</div>
</c:if>
```

MVC ile Form Örneği

```
@RequestMapping(value="/saveVet",method=RequestMethod.GET)
public String showForm() {
    return "saveVet";
}

@ModelAttribute("vet")
public Vet initFormModel() {
    return new Vet();
}

@RequestMapping(value="/saveVet",method=RequestMethod.POST)
public String handleFormSubmit(@ModelAttribute("vet") @Valid Vet
vet, BindingResult bindingResult, RedirectAttributes
redirectAttributes) {

    if (bindingResult.hasErrors()) {
        return "saveVet";
    }

    petClinicService.saveVet(vet);
    redirectAttributes.addFlashAttribute("message", "Vet created");
    return "redirect:saveVet";
}
```

POST + Redirect + GET Örüntüsü (PRG)

- Kullanıcının POST metot ile submit ettiği form verisini yanlılıkla **ikinci defa submit etmesinin** önüne geçmek için kullanılır
 - Refresh veya geri butonu ile gereksiz yere form resubmit tetiklenebilir
- POST isteklerinden sonra kullanıcı görüntülenecek sayfaya **REDIRECT komutu** ile yönlendirilir
 - **Location** header kullanılarak GET ile erişilecek yeni URL belirtilir

İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- info@java-egitimleri.com

