

URI Fragmanları, Vaadin Navigator ve Arama Motorları ile Etkileşim

URI Fragment

- URL'de '#' karakterinden sonraki bölümdür
- Vaadin içerisinde UI altındaki **View** nesnelere **URI ile erişmek** için kullanılır
- Çünkü JS'de **page reload yapmadan** URL'de değişebilecek tek bölümdür
- URI fragment'ler **Page** nesnesi tarafından yönetilirler

```
String uriFragment = Page.getCurrent().getUriFragment();  
Page.getCurrent().setUriFragment("myview");
```

Navigator

- Vaadin gibi ajax uygulamaları **tek sayfadan** oluşurlar
- Bu sayfa yüklenir ve içeriği **sadece ajax** istekleri ve cevapları ile değişikliğe uğrar
- Klasik web uygulamalarındaki gibi **sayfalar arasında gezinme** söz konusu değildir
- Bazı Vaadin uygulamalarında kullanıcının **ekranlar arasında dolaşması** veya belirli bir ekranı **URL ile doğrudan açması** gerekebilir

Navigator

- **Navigator** sınıfı bu ihtiyacı karşılar
- Navigator içerisinde **View** olarak nitelenen ekranlar yönetilir
- Her bir view kendine ait bir **URI fragment**'e sahiptir
- Bu sayede ekran ile ilgili tarayıcı üzerinde **bookmark oluşturma, ileri, geri gezinme** gibi kabiliyetler sağlanabilir
- Yada ekranın belirli bir **state**'ine **tarayıcıdan erişmek** mümkündür

URI Fragment ve Navigator

- Navigator nesnesinin **iki türlü kullanımı** vardır
- Ya spesifik bir **view'a** erişilebilir
- Ya da view'ın **belirli bir state'ine** erişmeyi sağlar

Navigator Kullanımı

```
public class PetClinicUI extends UI {
```

```
@Override
```

```
protected void init(VaadinRequest request) {
```

```
...
```

```
VerticalLayout navigatorContainer = new VerticalLayout();
```

```
Navigator navigator = new Navigator(this, navigatorContainer);
```

```
navigator.addView("", DefaultView.class);  
navigator.addView("first", FirstView.class);  
navigator.addView("second", SecondView.class);
```

```
setNavigator(navigator);
```

```
...
```

```
}  
}
```

Son olarak oluşturulan **Navigator** nesnesi **UI**'a eklenir. Böylece herhangi bir yerden current UI üzerinden Navigator'a erişilebilir

Navigator nesnesi yaratılırken input arg verilen **Container** içerisinde navigate edilen View nesneleri gösterilecektir

Navigate edilebilecek View nesneleri navigator'a karşılık geldikleri URI fragman'ları ile register edilirler

İstenirse **ViewProvider**'da tanımlanabilir

View nesneleri **com.vaadin.navigator.View** arayüzünü implement eden normal Vaadin **Component**'leridir

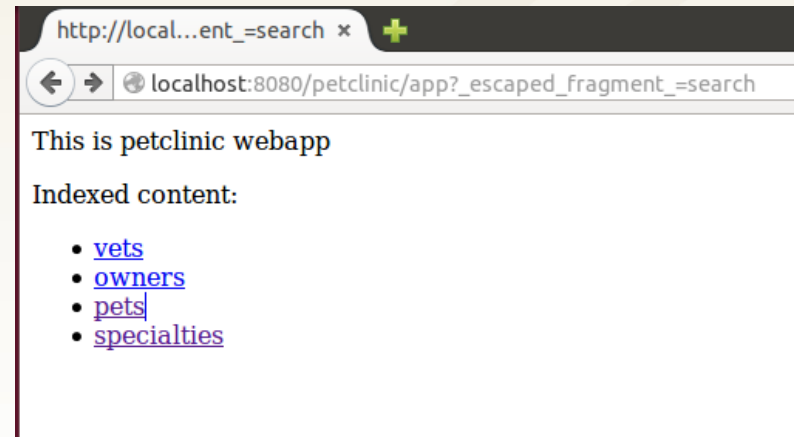
URI Fragment ve Arama Motorları

- '#' ifadesi spesifik bir View'a navigate etmenin yanı sıra bunun **stateful bir ajax sayfası** olduğunu da ifade eder
- Bu ifade ile uygulamanın aranabilir içeriğinin **arama motorları** tarafından ziyaret edilmesi sağlanabilir
- Arama motoru /myapp/myui#!myfragment URL ile karşılaşır sa sunucuya ayrıca /myapp/myui?_escaped_fragment_=myfragment şeklinde bir istek daha gönderir

URI Fragment ve Arama Motorları

```
public class SearchableVaadinServlet extends VaadinServlet {
    @Override
    protected void service(HttpServletRequest request,
        HttpServletResponse response) throws ServletException,
        IOException {
        String fragment =
            request.getParameter("_escaped_fragment_");
        if ("search".equals(fragment)) {
            response.setContentType("text/html");
            Writer writer = response.getWriter();
            writer.append("<html><body>");
            writer.append("<p>This is petclinic
webapp</p>");
            String items[] = { "vets", "owners", "pets",
                "specialties" };
            writer.append("<p>Indexed content:</p>");
            writer.append("<ul>");
            String contextPath =
                request.getContextPath();
            String servletPath =
                request.getServletPath();
            for (String item : items) {
                String url = contextPath +
                    servletPath + "#!" + item;
                writer.append("<li><a href='" +
                    url + "'" + item + "</a></li>");
            }
            writer.append("</ul></body>");
        } else
            super.service(request, response);
    }
}
```

Bu istek VaadinServlet üzerinden ele alınarak arama motoruna uygun bir içerik üretilerek dönülebilir



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

