

Vaadin Kullanıcı ve Sistem Mesajları

Kullanıcı Mesajları

- Bildirimler **Notification.show()** metodu ile yapılır:

```
Notification.show("This is the caption",  
"This is the <i>description</i>",  
Notification.Type.WARNING_MESSAGE);
```

→ HTML tagleri kullanılabilir!

- Bildirim tipleri:

Humanized message
For minimal annoyance

→ Bir süre sonra kendiliğinden kapanır

Warning message
For notifications of medium importance

Kullanıcı kapatıncaya kadar görüntülenir

Error message ✕
For important notifications

Sayfanın sağ alt kısmında görünür,
kendiliğinden kaybolur.

Tray notification
Stays up longer - but away

2

Kullanıcı Mesajları

- İstenirse Notification sınıfından bir nesne oluşturarak davranışı tamamen uygulamaya özel bir mesaj da ekranda görüntülenebilir
- Örneğin delayMsec değeri değiştirilebilir, farklı bir icon set edilebilir veya HTML içerik görüntülenebilir

```
Notification notification = new Notification("Success",
    "<b>Vet</b> created", Notification.Type.HUMANIZED_MESSAGE, true);

notification.setDelayMsec(5000);

notification.show(Page.getCurrent());
```

Hataların Ele Alınması

- Tüm bileşenlerin **setComponentError()** metodu ile aktive edilebilen hata indikatörleri vardır
- Sistemde yakalanmayan hatalar **DefaultErrorHandler** tarafından ele alınır
- Hataya sebep olan bileşenin indikatörü aktive edilir ve imleç üzerine geldiğinde **stack trace** görüntülenir
- Default Error handling davranışı override edilebilir

```
UI.getCurrent().setErrorHandler(new DefaultErrorHandler() {
    @Override
    public void error(com.vaadin.server.ErrorEvent event) {
        ...
        // Do the default error handling (optional)
        doDefault(event);
    }
});
```

Sistem Mesajları

- Sistem mesajları genellikle uygulamanın state'i ile ilgili olumsuz bir durum hakkında bilgi verirler
- Çoğu zaman bu durumdan kurtulmak için uygulamanın yeniden başlatılması gerekir
- Sistem mesajları **SystemMessages** sınıfı içerisinde tanımlıdır
- **SystemMessagesProvider** ile bu mesajlar uygulamaya göre özelleştirilebilir

Sistem Mesajları

- **sessionExpired**: Vaadin session'ın sona erdiğini anlatır. Session timeout web.xml içerisinde yönetilebilir.
- **communicationError**: Vaadin client engine ile uygulama sunucusu arasında tanımsız bir iletişim hatası olduğunu anlatır. Sunucu hizmet vermiyor olabilir.
- **authenticationError**: Sunucu tarafından 401(Unauthorized) cevabı döndüldüğünde gösterilir.

Sistem Mesajları

- **internalError**: Vaadin client engine veya custom client-side kodda bir hata olduğunu anlatır.
- **outOfSync**: Client ve server side state'lerin birbirleri ile uyumsuz olduğunu anlatır.
- **cookiesDisabled**: Cookie'lerin aktif olmadığını, uygulamanın bunlarsız çalışmayacağını anlatır.

Sistem Mesajlarının Özelleştirilmesi

```
vaadinRequest.getService().setSystemMessagesProvider(  
    new SystemMessagesProvider() {  
        @Override  
        public SystemMessages getSystemMessages(SystemMessagesInfo  
            systemMessagesInfo) {  
  
            CustomizedSystemMessages messages =  
                new CustomizedSystemMessages();  
  
            messages.setCommunicationErrorCaption("Comm Err");  
            messages.setCommunicationErrorMessage("This is bad.");  
            messages.setCommunicationErrorNotificationEnabled(true);  
            messages.setCommunicationErrorURL("http://www.example.org");  
  
            return messages;  
        }  
    });
```

SystemMessages sınıfından türer. Belirtilen bütün mesajlar bu nesne içerisinde özelleştirilebilir.

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

