

Vaadin UI Framework'e Giriş

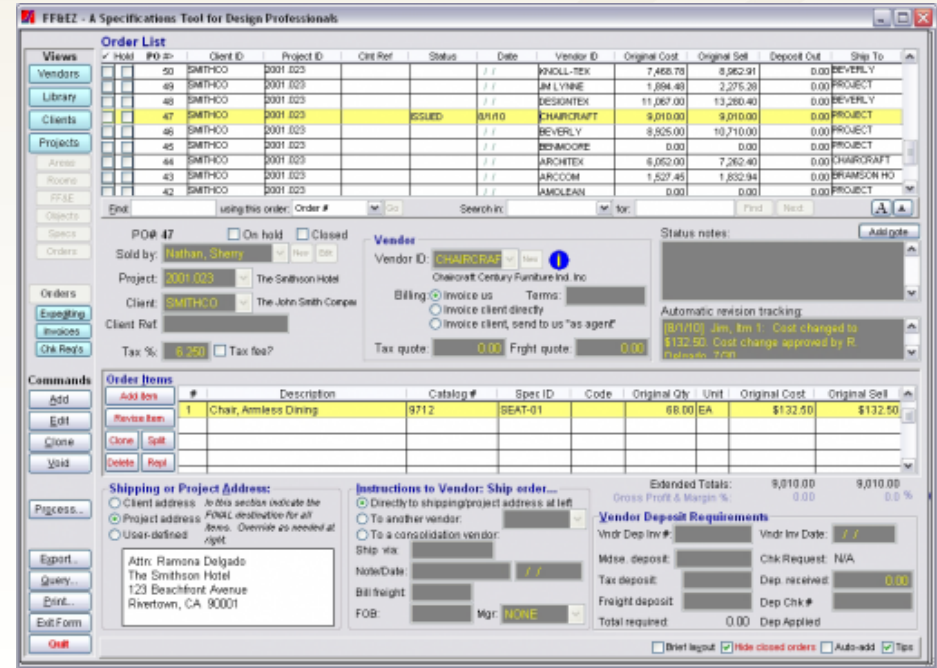
Vaadin Nedir?

- **Zengin içerikli web uygulamaları için bir UI framework'tür**
- **Java tabanlıdır**
- UI geliştirme **sadece Java kodu ile** gerçekleştirilir
- İstemci tarafında **HTML + CSS + Javascript** çalışır
- İstemci tarafında **tarayıcıdan başka birşeye** gerek duyulmaz

Web Siteleri vs Web Uygulamaları

Vaadin zengin içerikli kurumsal web uygulamaları geliştirmek için tasarlanmış bir UI Framework'tür.

Web sitesi geliştirmek için **başka** çözümlere yönelmelisiniz!



Vaadin'e Giriş

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

Vaadin uygulamasının bileşen hiyerarşisindeki en üst yapıdır. Her bir tarayıcı penceresi veya tab'ı bir UI instance'ına karşılık gelir.

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

UI'nın init metodu override edilir

```
    Button button = new Button("Click Me");
    button.addClickListener(new Button.ClickListener() {
        public void buttonClick(ClickEvent event) {
            layout.addComponent(new Label("Thank you for clicking"));
        }
    });
});
```

```
setContent(button);
```

UI'nın içeriği set edilir

```
}
```

Vaadin web.xml Ayarları

```
<web-app id="WebApp_ID" version="2.4" xmlns="...">
  <servlet>
    <servlet-name>PetClinicApplication</servlet-name>
    <servlet-class>com.vaadin.server.VaadinServlet</servlet-class>
    <init-param>
      <param-name>UI</param-name>
      <param-
value>com.javaegitimleri.petclinic.view.PetClinicUI</param-value>
    </init-param>
  </servlet>

  <servlet-mapping>
    <servlet-name>PetClinicApplication</servlet-name>
    <url-pattern>/app/*</url-pattern>
  </servlet-mapping>

  <servlet-mapping>
    <servlet-name>PetClinicApplication</servlet-name>
    <url-pattern>/VAADIN/*</url-pattern>
  </servlet-mapping>
</web-app>
```

Debug Modu & Penceresi

- Vaadin uygulamaları **iki modda** çalıştırılabilir
 - Debug
 - Production
- Default mod debug'tır.
- Production modunu aktive etmek için aşağıdaki **context parametresinin** **web.xml**'de tanımlı olması gerekir

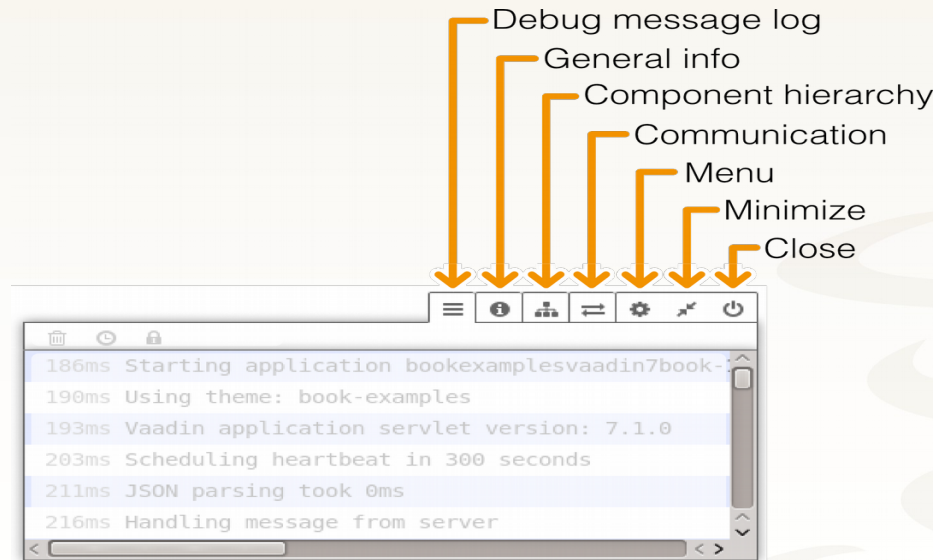
```
<context-param>
  <param-name>productionMode</param-name>
  <param-value>true</param-value>
</context-param>
```

Debug Modu & Penceresi

- Debug penceresini açmak için uygulama URL'ine "**?debug**" request parametresinin eklenmesi yeterlidir
- Örnek:
<http://localhost:8080/petclinic/app?debug>

Debug Modu & Penceresi

- Açılan pencereden **debug log mesajları**, kullanılan Vaadin sürümü ile ilgili **genel bilgiler**, render edilen **UI bileşen hiyerarşisi**, client-server arasındaki **haberleşme**, seçilen bir **bileşenle ilgili detaylı bilgi** elde edilebilir



Application Restart

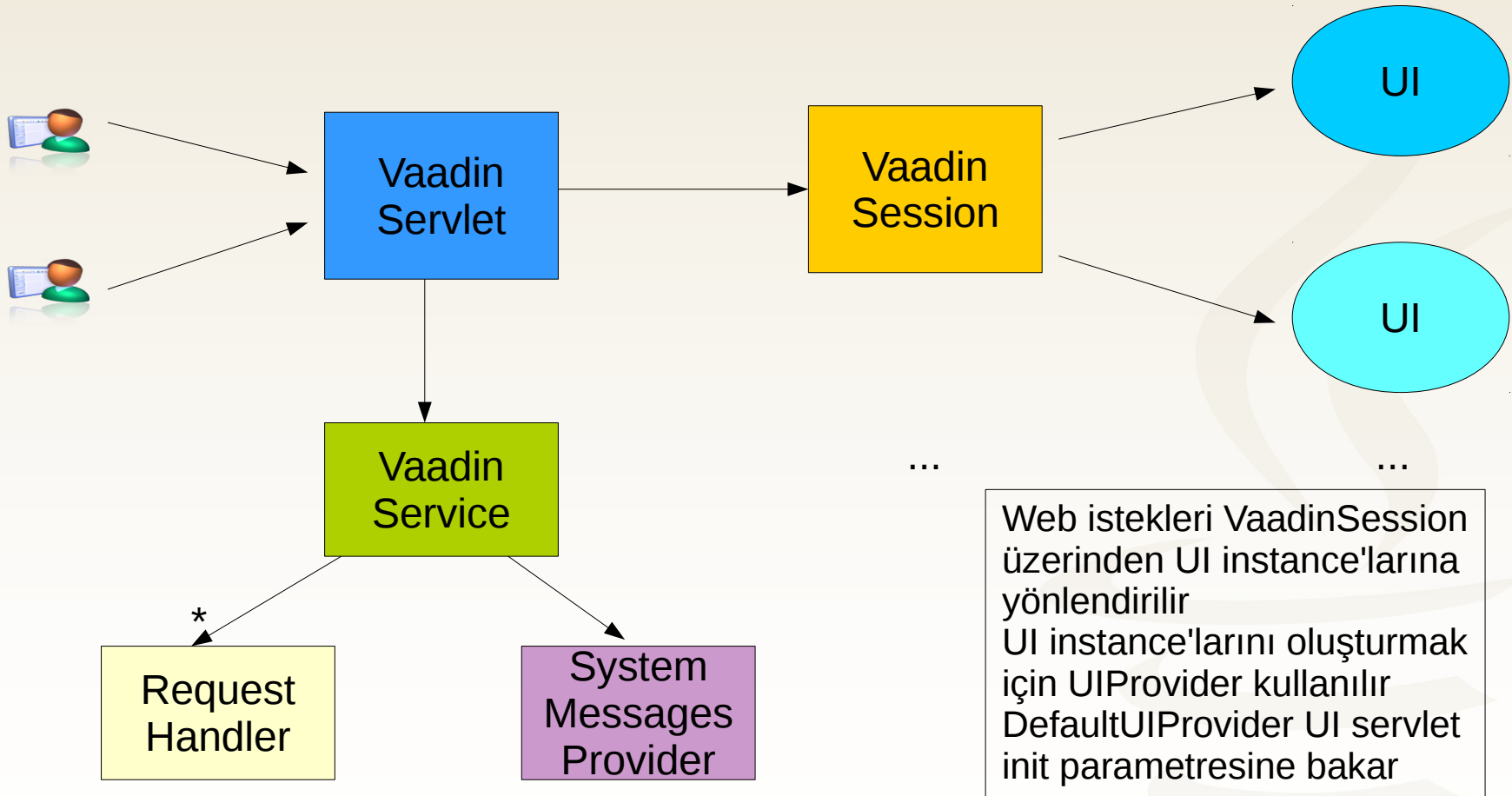
- Vaadin bir **server side framework**'tür
- Kod **sunucuda** çalıştırılır
- **UI state**'de sunucuda tutulur
- Dolayısı ile **tarayıcı** sadece **kendisine dönülen state**'i sunmaktadır
- Eğer sunucu **uygulama state**'ini **cache'lemiş ise** tarayıcının **refresh(F5)** yapılması kod değişikliklerinin ekrana yansımaya yetmeyebilir

Application Restart

- Vaadin bileşenleri **Serializable**'dir
- Sunucu **UI state'i save** edebilir
- Dolayısı ile **sunucu restart'ları da** ekran değişikliklerinin görüntülenmesi için yeterli olmayabilir
- Uygulama “**restartApplication**” request parametresi ile çağırılır ise yeniden başlatılır
- Yeni bir Vaadin **uygulama instance** yaratılır ve **UI.init()** metodu çalıştırılır

Vaadin UI – Servlet

– Service – Session İlişkisi



Application UI

- Tüm Vaadin uygulamaları **com.vaadin.ui.UI** sınıfından türetilen bir UI içerir
- Uygulamanın giriş noktası olarak görülebilir
- Genellikle uygulama düzeyinde bir tanedir
- Ancak uygulamalar **birden çok** UI'dan oluşabilir
- Current UI instance'ına **UI.getCurrent()** ile erişilebilir

VaadinServlet

- Sunucu tarafı **VaadinServlet** üzerinden çalışır
- İstemcilerden gelen **istekleri karşılar** ve bunları ait oldukları **kullanıcı oturumlarına** yönlendirir
- Servlet instance'ına **VaadinServlet.getCurrent()** ile erişilebilir
- VaadinServlet'in davranışı, **extend** edilerek özelleştirilebilir

VaadinSession

- Uygulamada **kullanıcıya ait herhangi bir veriyi tutan yapıdır**
- **HttpSession'a benzer**
- Aslında HttpSession içerisinde yönetilen bir yapıdır
- Gelen web isteklerinin VaadinSession üzerinden **UI instance'larına erişimi** sağlanır

VaadinService

- Uygulamaya ait deployment spesifik bilgilerin yönetildiği yapıdır
- **SystemMessagesProvider** üzerinden sistem mesajlarına erişilebilir
- Bu mesajlar uygulamaya özel hale getirilebilir
- **RequestHandler**'ları yönetir
- RequestHandler'lar üzerinden web request'inin ele alınması sağlanır

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

