

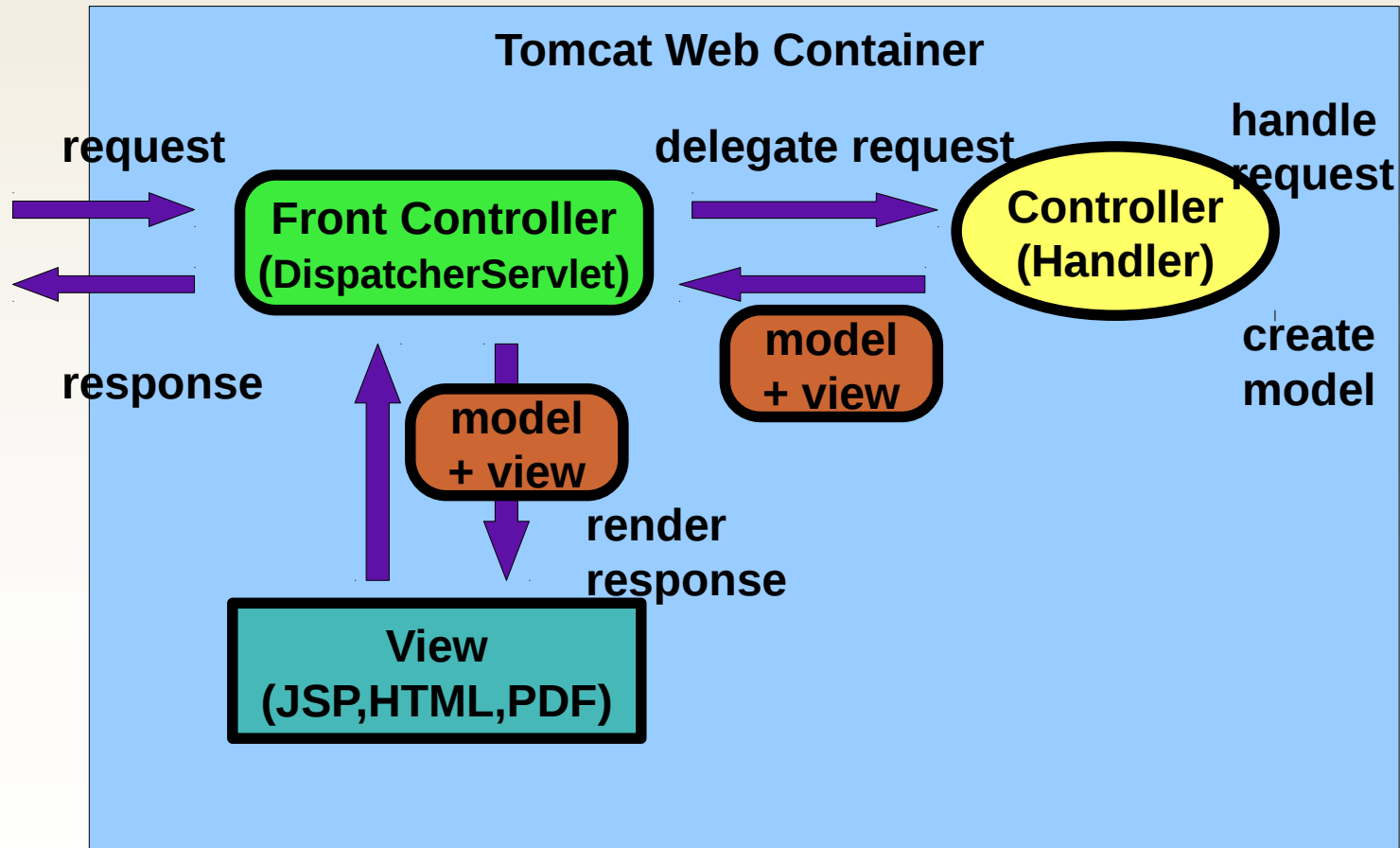
Spring MVC Configuration



Spring MVC ve DispatcherServlet

- Spring MVC de diğer pek çok MVC framework gibi **request ve action tabanlı** bir frameworktür
- **DispatcherServlet** etrafında kuruludur
- Bu servlet, Spring IoC Container ile **tam bir entegrasyona** sahiptir
- DispatcherServlet, **Front Controller** örüntüsünü implement eder

Front Controller Java EE Pattern



DispatcherServlet Konfigürasyonu

- DispatcherServlet'in aktivasyonu için **web.xml** içerisinde tanım yapılarak, burada **hangi requestleri ele alacağı** belirtilir

web.xml

```
<web-app>
  <servlet>
    <servlet-name>DispatcherServlet</servlet-name>
    <servlet-class>
org.springframework.web.servlet.DispatcherServlet
    </servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>DispatcherServlet</servlet-name>
    <url-pattern>/mvc/*</url-pattern>
  </servlet-mapping>
</web-app>
```

DispatcherServlet Konfigürasyonu

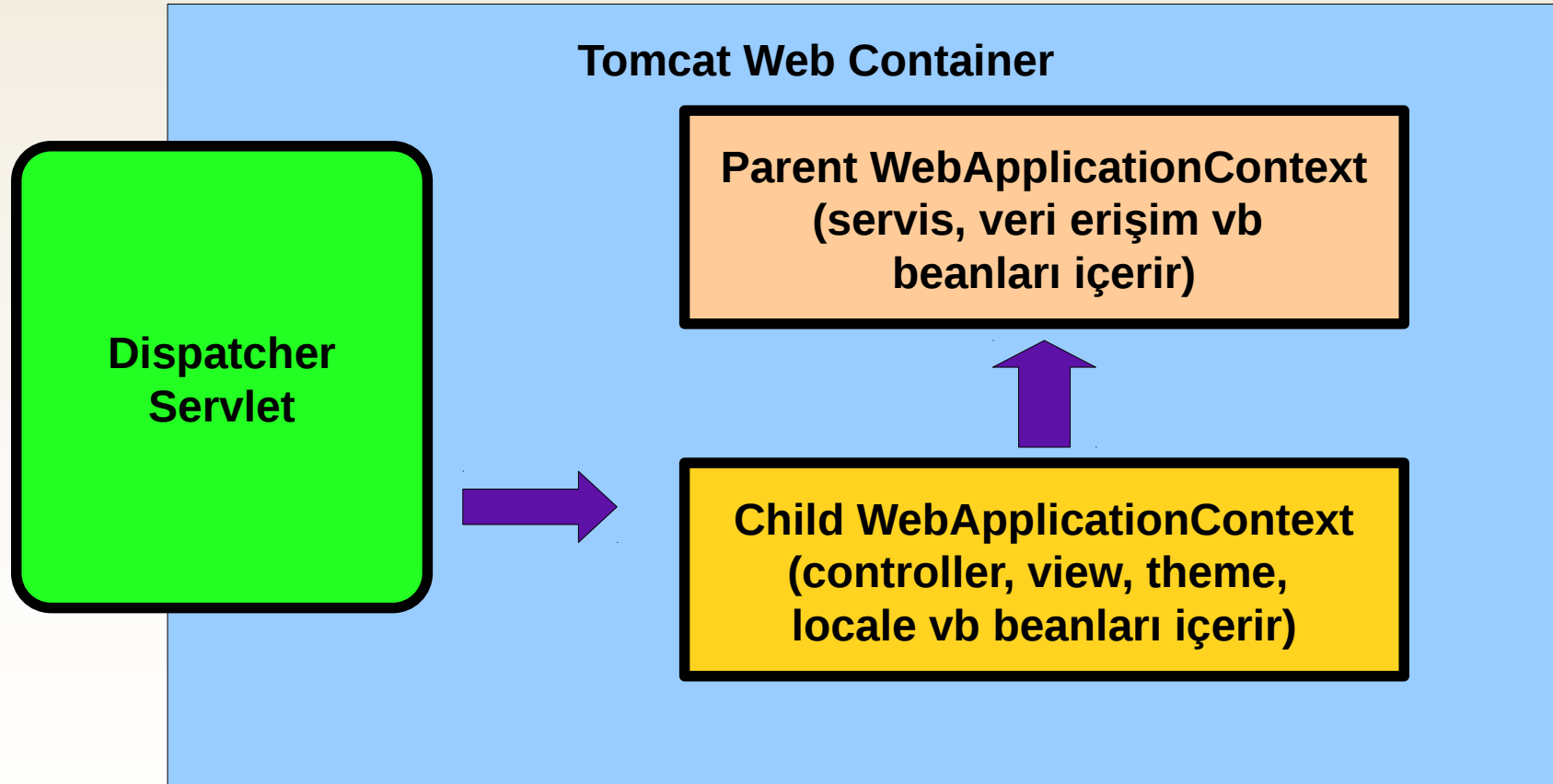
- Default olarak **WEB-INF** dizini altında **[servlet-name]-servlet.xml** isminde bir Spring bean konfigürasyon dosyası arar
- Bu dosyayı yükleyerek **kendine ait bir WebApplicationContext** yaratır
- Bu WebApplicationContext, eğer tanımlı ise **ContextLoaderListener** ile yaratılan WebApplicationContext'in çocuğudur

DispatcherServlet Konfigürasyonu

- WEB-INF dizini altındaki dosyanın lokasyonu veya adı istenirse **contextConfigLocation** servlet init parametresi ile değiştirilebilir, ya da **birden fazla dosya** lokasyonu belirtilebilir

```
<servlet>
  <servlet-name>DispatcherServlet</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:/appcontext/mvc-servlet.xml</param-value>
  </init-param>
</servlet>
```

DispatcherServlet ve ApplicationContext Hiyerarşisi



DispatcherServlet ve ApplicationContext Hiyerarşisi

- DispatcherServlet'a ait çocuk **WebApplicationContext** içerisinde, **ContextLoaderListener** ile yaratılan **root WebApplicationContext** içerisindeki bütün bean'lere erişilebilir
- Ancak parent **ApplicationContext** içerisinde **çocuk ApplicationContext'deki bean'lara erişim mümkün değildir**

DispatcherServlet ve ApplicationContext Hiyerarşisi

- Eğer DispatcherServlet'ın kendine ait child WebApplicationContext'inin olması istenmiyor ise **contextConfigLocation** init parametre değeri boş bırakılmalıdır

```
<servlet>
  <servlet-name>DispatcherServlet</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value></param-value>
  </init-param>
</servlet>
```

Spring MVC Özelliklerinin Aktivasyonu

- `<mvc:annotation-driven />` elemanı ise aşağıdaki kabiliyetleri devreye sokar:
 - Built-in **HttpMessageConverter** nesnelerini register eder
 - **Formatter** ve **Conversion** servislerini devreye alır
 - Controller handler metot parametrelerindeki **@NumberFormat**, **@DateTimeFormat** anotasyonları devreye sokar

Spring MVC Özelliklerinin Aktivasyonu

- **@ControllerAdvice**'in devreye girmesini sağlar
- Controller metotlarında **validasyon kabiliyetini** devreye sokar
- Controller handler metotlarının input argümanlarında veya return değerinde **@Valid** anotasyonu kullanılabilir

İletişim

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

