



Spring MVC 3

Pregled značajki
Praktični primjeri

Neven Borojević, dipl.ing.
Beta Tau Beta d.o.o.

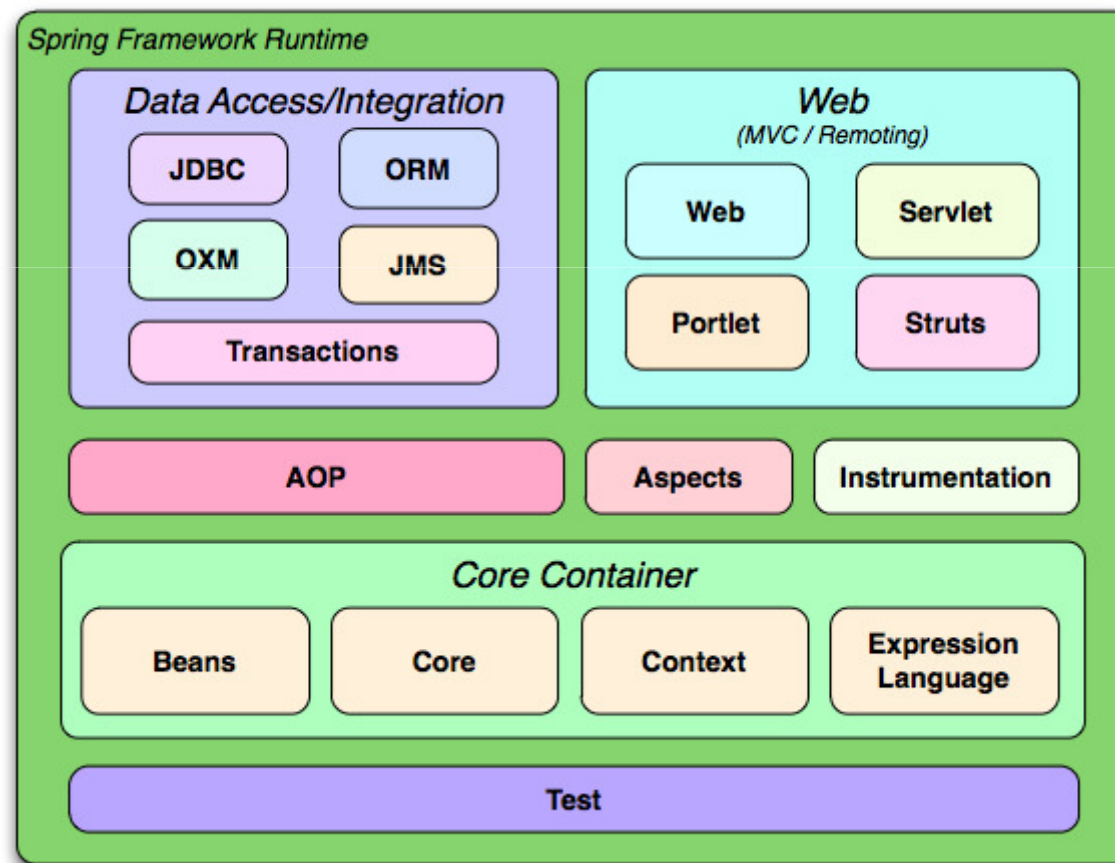


Spring

- Sveobuhvatan framework
- Spring implementira infrastrukturu – korisnik se koncentrira na aplikaciju
- POJO aplikativne komponente
- Primjena enterprise servisa na POJO komponente

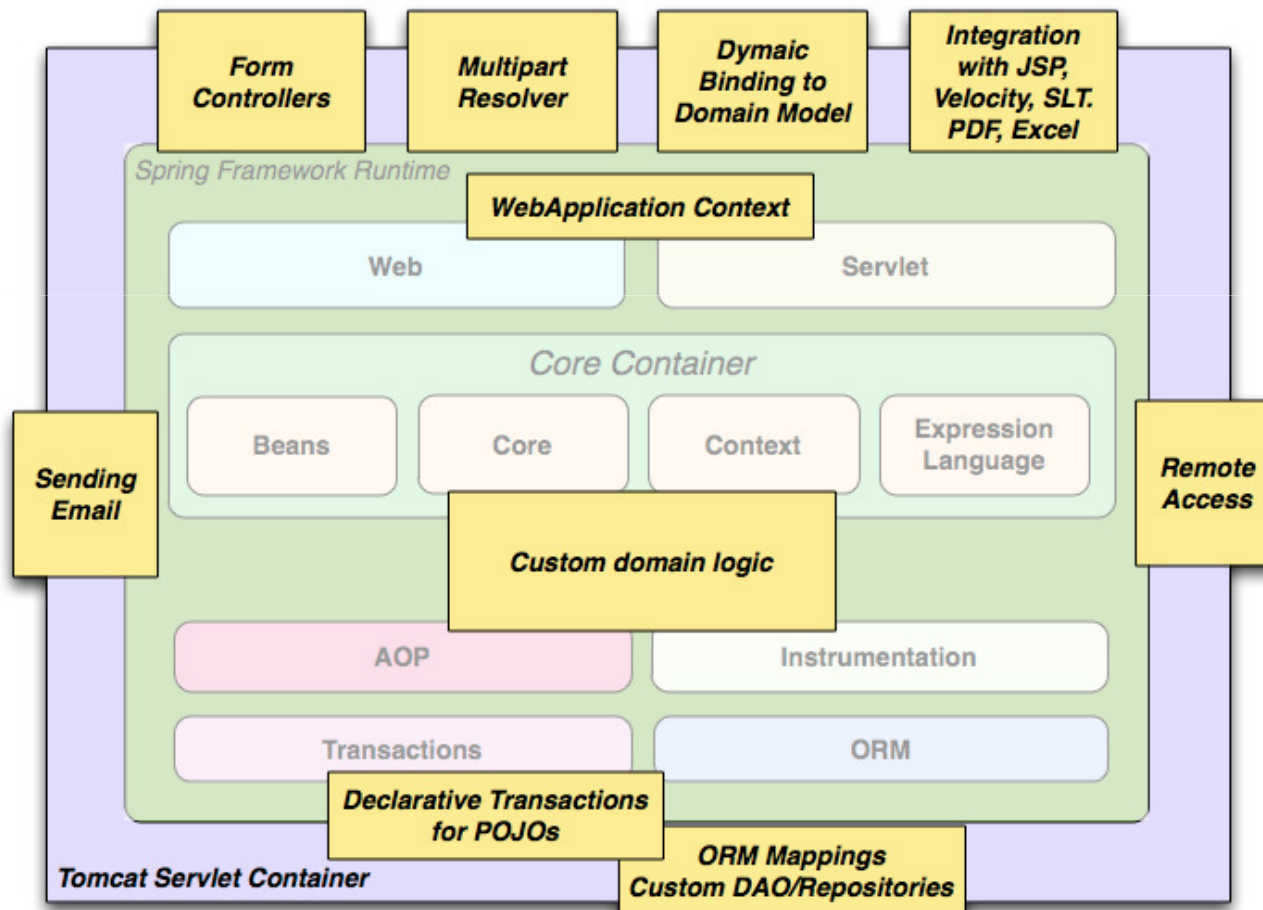


Spring - Moduli





Spring – Scenarij korištenja





Spring MVC – karakteristike

- Izgrađen oko *DispatcherServleta*
- Distribucija requesta na *handlera*
- Konfiguracija:
 - Mapiranje *handlera*
 - Rezolucija *view-ova*
 - Rezolucija *locale-a* i tema
 - Podrška za *upload*



Spring MVC – karakteristike

- Bilo koji objekt kao *command* ili *backing-bean*
- Ne zahtijeva implementaciju *interfacea* ili bazne klase specifične za framework
- Fleksibilan *data binding*



Spring MVC – karakteristike

- Rezolucija *view*-ova je krajnje fleksibilna
- *ModelAndView* – referenca na *view* + *Map* sa sadržajem modela
- Model nezavisan od *view* tehnologije



Spring MVC – karakteristike

- Separacija uloga po komponentama
- Komponente frameworka i aplikacije - POJO
- Adaptibilnost, fleksibilnost, neinvazivnost
- Transparentno korištenje business modela

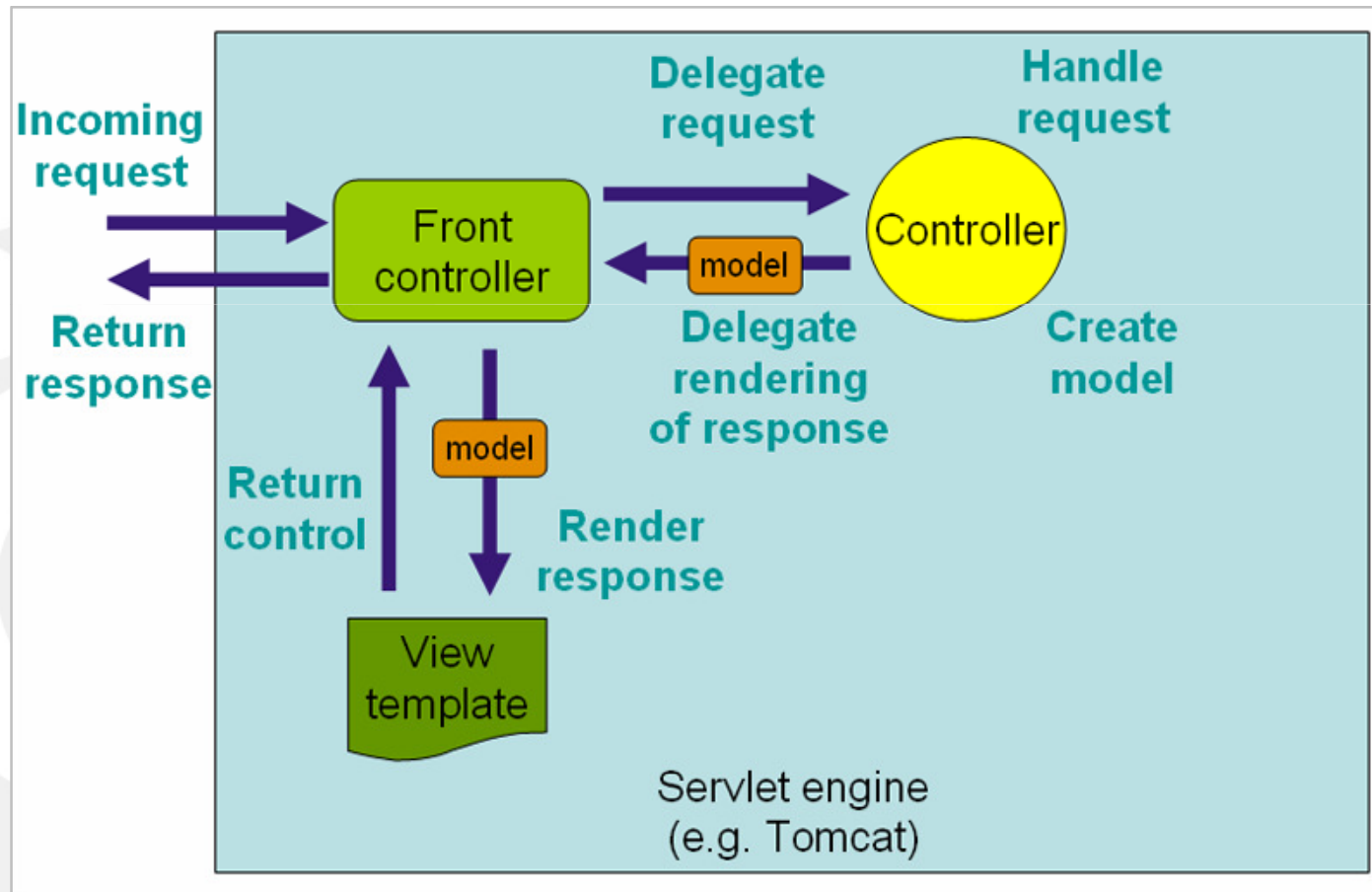


Spring MVC – karakteristike

- Fleksibilan *binding* i validacija
- Fleksibilno mapiranje *handlera* i *view-ova*
- Fleksibilan prijenos modela
- Fleksibilna rezolucija *localea* i tema
- Jednostavan i snažan *taglib*



DispatcherServlet



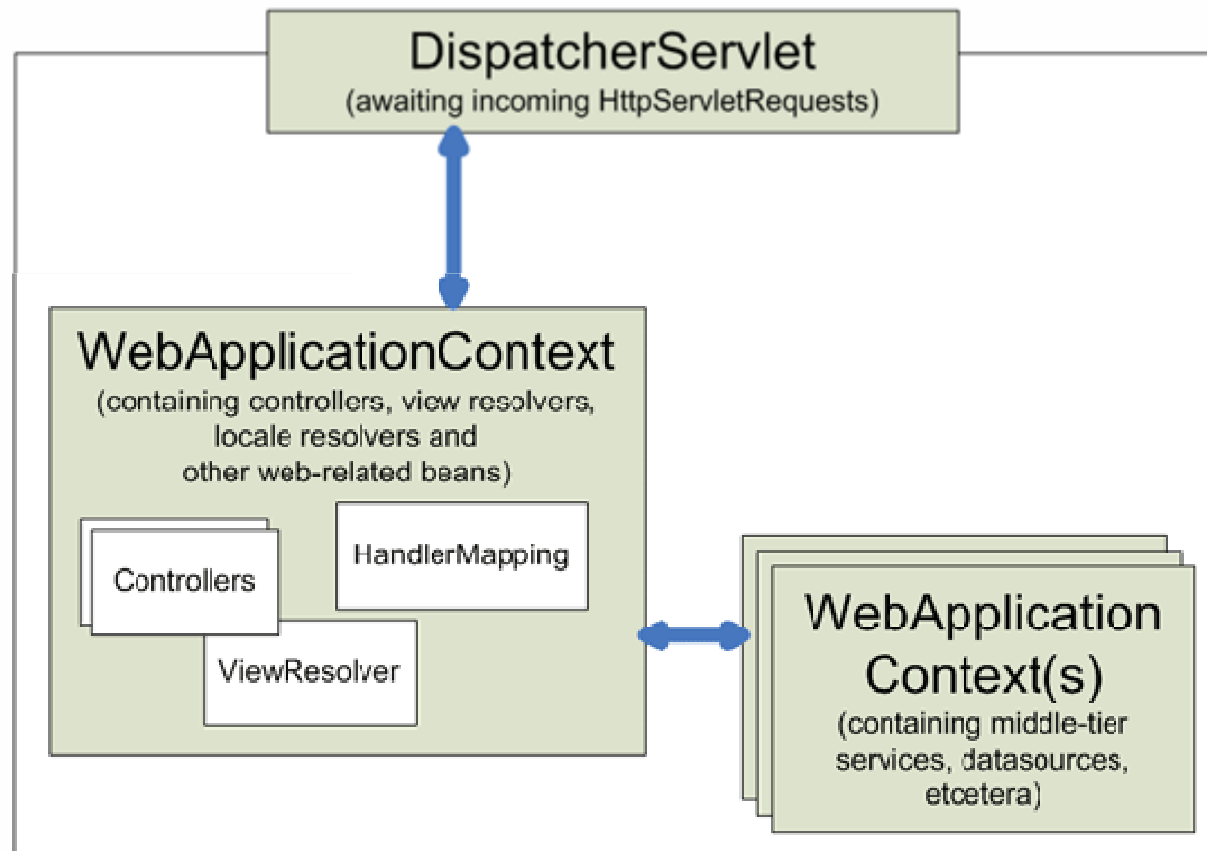


DispatcherServlet

```
<web-app>
  <servlet>
    <servlet-name>example</servlet-name>
    <servlet-class>
      org.springframework.web.servlet.DispatcherServlet
    </servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>example</servlet-name>
    <url-pattern>*.form</url-pattern>
  </servlet-mapping>
</web-app>
```



DispatcherServlet





WebApplicationContext

- [servlet-name]-servlet.xml
- datoteka sa deklaracijom Spring MVC spec. beanova
- ekstenzija *ApplicationContexta*
- konfiguracija beanova za rezoluciju requesta i viewova

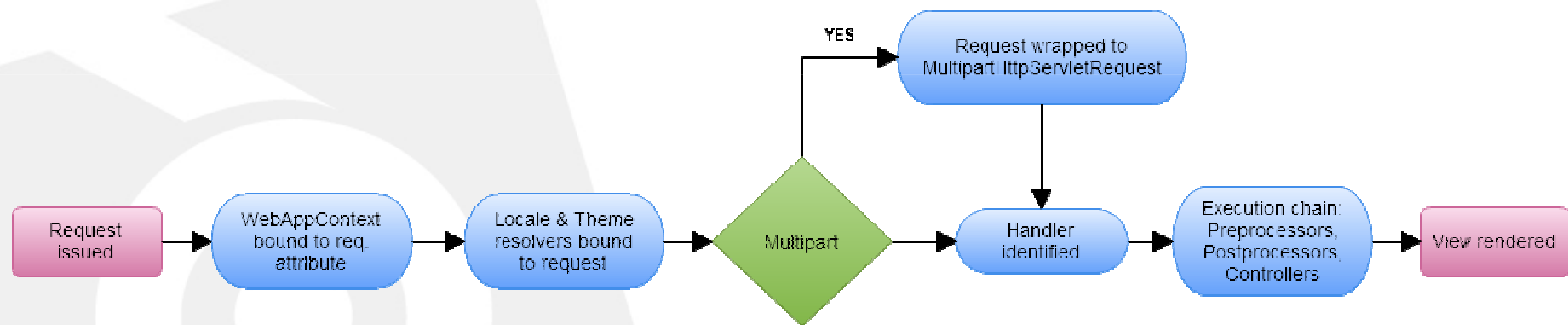


WebApplicationContext

- standardni beanovi
 - kontroleri (**controllers**)
 - mapiranje handlera (**handler mappings**)
 - rezolucija viewova (**view resolvers**)
 - locale i theme resolveri
 - multipart file resolveri
 - handler exception resolveri



DispServlet workflow





Kontroleri

- Kontroler
 - Omogućava pristup poslovnoj logici u servisnom sloju
 - Konvertiraju korisnički unos u model koji se prenosi u poslovnu logiku i natrag na prezentaciju



Kontroleri

- Spring 2.5 nadalje
 - Pristup upotrebom anotacija
 - @Controller
 - @RequestMapping
 - @RequestParam
 - @ModelAttribute...



Kontroleri

- @Controller
- Skeniranje anotiranih klasa, detekcija @RequestMapping anotacija
- Shema spring-context-3.0.xsd

```
<context:component-scan  
base-package="org.springframework.samples.petclinic.web"/>
```



Mapiranje requesta

- Mapiranje URL ↔ klasa / handler metoda
- `@RequestMapping`
- Value – primarno mapiranje
- Path URI: npr. `/myPath.do`
- Ant-style: `/myPath/*.do`
- Na razini klase i metode



Mapiranje requesta

- URI Template
- Dio URI-ja popunjava klijent prije upotrebe
- Predloženi RFC
- Dohvat dijelova URL-a u handler metodi
- `@PathVariable`
- Npr. `/owners/{ownerId}`



Mapiranje requesta

- `@PathVariable`
- Dekorirane varijable jednostavnih tipova:
 - `int`, `Date`, `long` itd.
 - Automatska konverzija
 - `TypeMismatchException`
- `WebDataBinder`



Mapiranje requesta

- @RequestParam
- @RequestBody
 - HttpMessageConverter
 - FormHttpMessageConverter
- @ResponseBody
 - HttpMessageConverter
 - StringHttpMessageConverter



Mapiranje requesta

- `@ModelAttribute`
- Prijenos modela u handler metodu
 - Razina argumenta handler metode
- Kreiranje referentnih objekata
 - Razina metode
 - Metoda prije bilo koje handler metode
 - `@SessionAttributes`



Validacija

- JSR-303 – JavaBean validation
- Hibernate Validator – referentna implementacija
- Metapodaci u samom beanu
- Npr. `@NotEmpty`, `@Email`, `@NotNull...`



Validacija

- @Valid
- BindingResult
- Prikaz validacijskih poruka na GUI-ju



File upload

- `Multipart request` → `MultipartResolver`
- `(Commons FileUpload)`
- `HttpServletRequest` →
`MultipartHttpServletRequest`
`(MultipartFile)`



Rad s localeom

- Podrška za i18n
- Korištenje lokaliziranih tekstova
- LocaleResolver
 - AcceptHeaderLocaleResolver
 - CookieLocaleResolver
 - SessionLocaleResolver
- LocaleChangeInterceptor



Rad s exceptionima

- DispatcherServlet registrira
DefaultHandlerExceptionHandler

Exception	HTTP Status Code
ConversionNotSupportedException	500 (Internal Server Error)
HttpMediaTypeNotAcceptableException	406 (Not Acceptable)
HttpMediaTypeNotSupportedException	415 (Unsupported Media Type)
HttpMessageNotReadableException	400 (Bad Request)
HttpMessageNotWritableException	500 (Internal Server Error)
HttpRequestMethodNotSupportedException	405 (Method Not Allowed)
MissingServletRequestParameterException	400 (Bad Request)
NoSuchRequestHandlingMethodException	404 (Not Found)
TypeMismatchException	400 (Bad Request)



Rad s exceptionima

- `@ExceptionHandler` anotacija na handler metodi
- Potpis handler metode fleksibilan



Konfiguracija

- `<mvc:annotation-driven>`
 - `ConversionService`
 - `@NumberFormat, @DateTimeFormat`
 - `@Valid`
 - `XML marshal/unmarshal`
 - `JSON marshal/unmarshal`



Konfiguracija

- `<mvc-interceptors>`
- **Npr.** `LocaleChangeInterceptor`
- `<mvc:view-controller>`
- **Npr.** `<mvc:view-controller path="/"
view-name="home" />`



POST-REDIRECT-GET

- Zaštita od nehotičnog submita forme
- Problem: REFRESH, BACK
- 1. POST handler
- 2. REDIRECT view
- 3. GET handler
- Rješenje: Nema problematičnog URL-a u history-ju