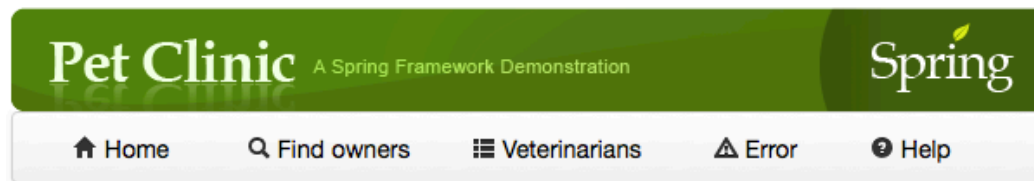


# Spring-Petclinic



# Spring Petclinic

- <https://github.com/SpringSource/spring-petclinic>
- <http://spring-petclinic.cloudfoundry.com/>



## Welcome



Sponsored by SpringSource 

## *Views*

JSP with  
custom tags

||

Thymeleaf

+

Bootstrap (CSS)

&&

webjars

&&

Dandelion

## *Controller*

Spring @MVC annotations

Bean Validation

## *Service*

@Cacheable

@Transactional

## *Repository*

3 profiles

Spring Data JPA

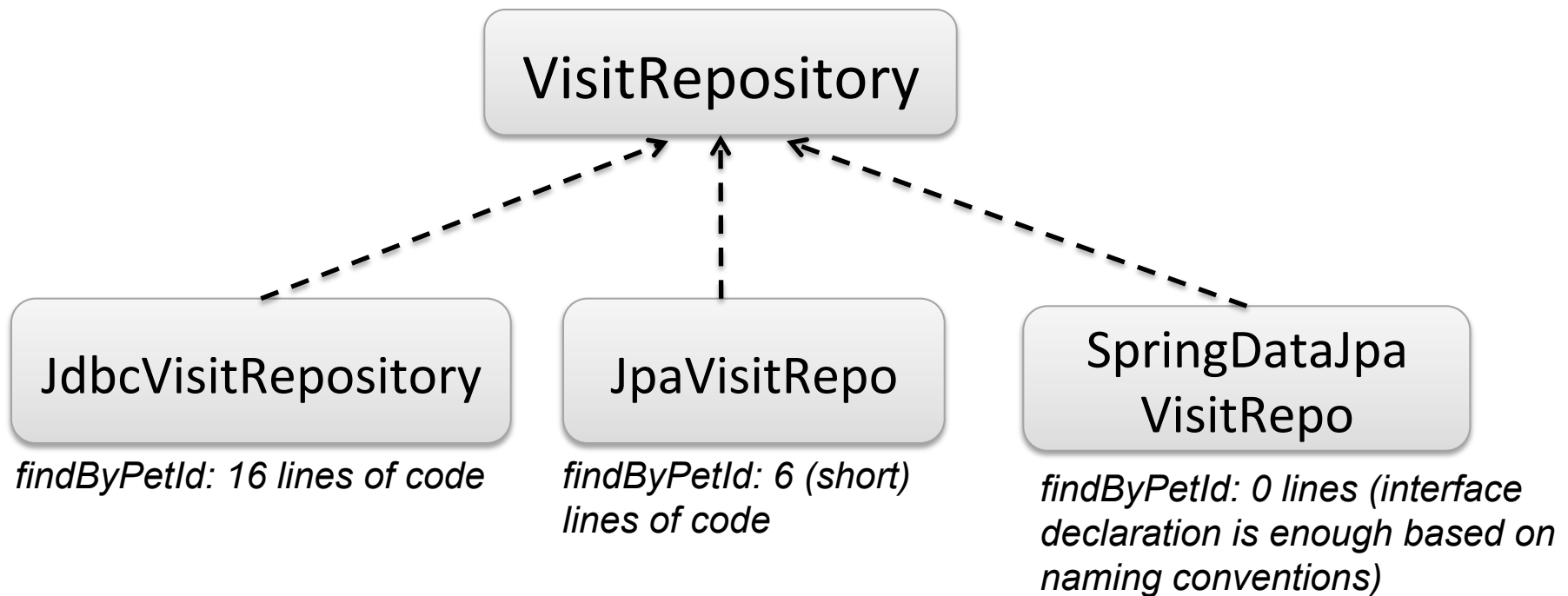
default (JPA)

jdbc

# Topics

- Core Spring
- The Web layer

# Data Access



In order to select which implementation should be used, select the Appropriate bean profile inside web.xml (jdbc, jpa or spring-data-jpa)

# Bean profiles



Inside JUnit tests

```
@ContextConfiguration(locations = ...)
@RunWith(SpringJUnit4ClassRunner.class)

@ActiveProfiles("jdbc")
public class JdbcOwnerRepositoryTests
...{}
```

Inside web.xml

```
<context-param>
  <param-name>
    spring.profiles.active
  </param-name>
  <param-value>
    jdbc
  </param-value>
</context-param>
```

No configuration needed in case you wish to use the default profile (JPA)

# Caching

- The list of Veterinarians is cached using ehcache

```
@Cacheable(value = "vets")  
public Collection<Vet> findVets() throws  
DataAccessException { ... }
```

*ClinicServiceImpl*

```
<cache name="vets"  
timeToLiveSeconds="60"  
maxElementsInMemory="100" ... />
```

*ehcache.xml*

```
<!-- enables scanning for @Cacheable annotation -->  
<cache:annotation-driven/>  
  
<bean id="cacheManager" class="org...EhCacheCacheManager">  
  <property name="cacheManager" ref="ehcache"/>  
</bean>  
  
<bean id="ehcache" class="org...EhCacheManagerFactoryBean">  
  <property name="configLocation" value="classpath:ehcache.xml"/>  
</bean>
```

*tools-config.xml*

# Exception Handling

## SimpleMapping ExceptionHandler

*Declared in mvc-core-config.xml*

*Based on the configuration used in petclinic:*

- *Logs the exception stacktrace*
- *Forwards to WEB-INF/jsp/exception.jsp*
- *Exception logged as a comment inside exception.jsp*

PetController

*Exception is **not handled there**  
It is propagated.*

ClinicService

*Transaction is rolled back  
in case of a RuntimeException  
(exception is still propagated to PetController)*

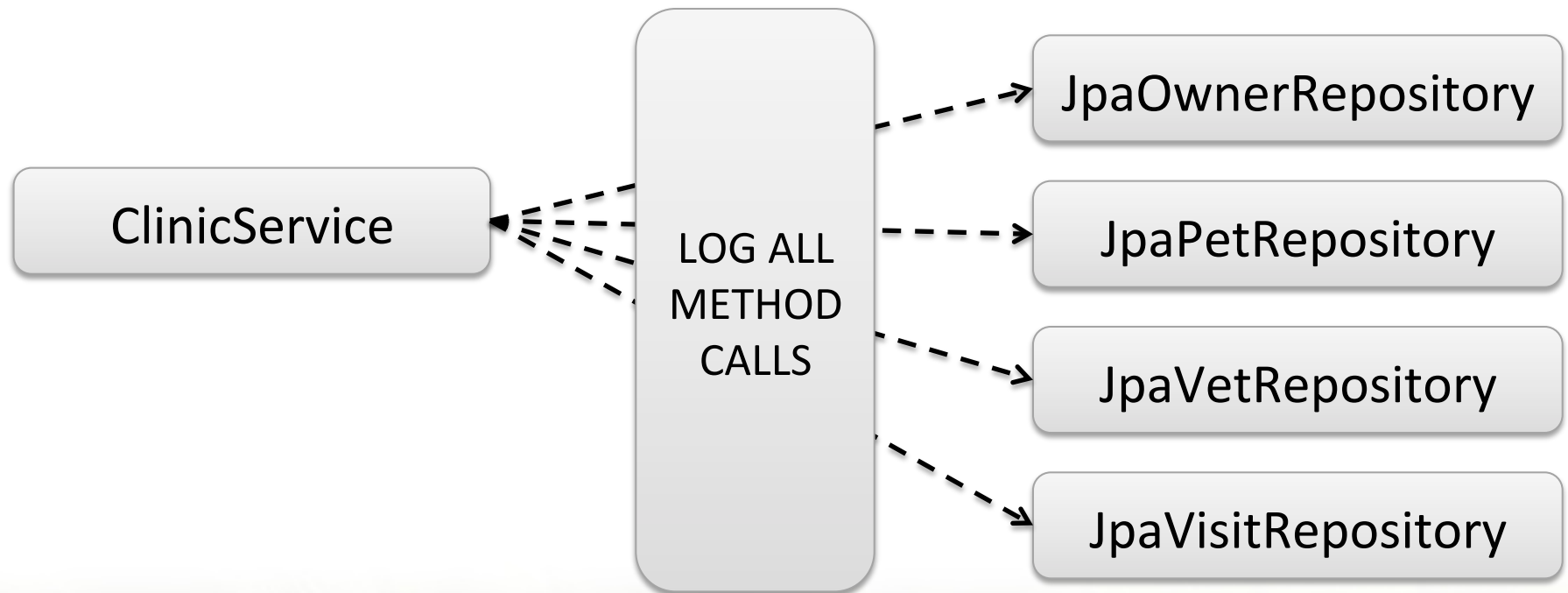
PetRepository

*May throw a RuntimeException  
(typically DataAccessException)*



# Aspect Oriented Programming (1/2)

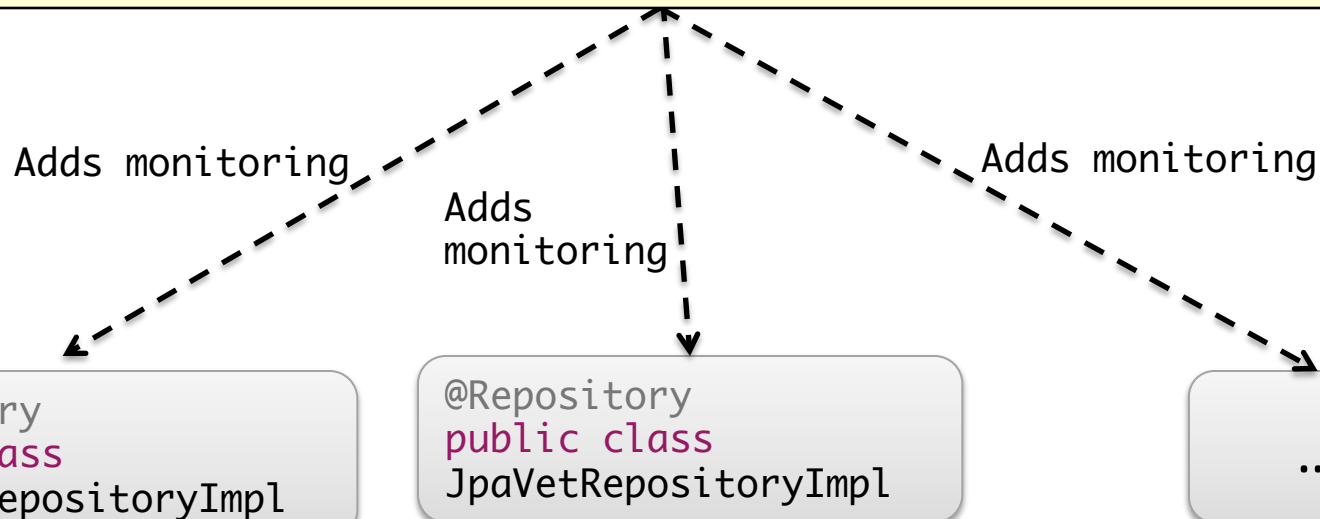
- How to add behavior in all methods of all Repository classes?



# Aspect Oriented Programming (2/2)

- CallMonitoringAspect

```
@Aspect
public class CallMonitoringAspect {
    @Around("within(@org.springframework.stereotype.Repository *)")
    public Object invoke(...) { ... }
}
```



To understand further how AOP works in Spring:

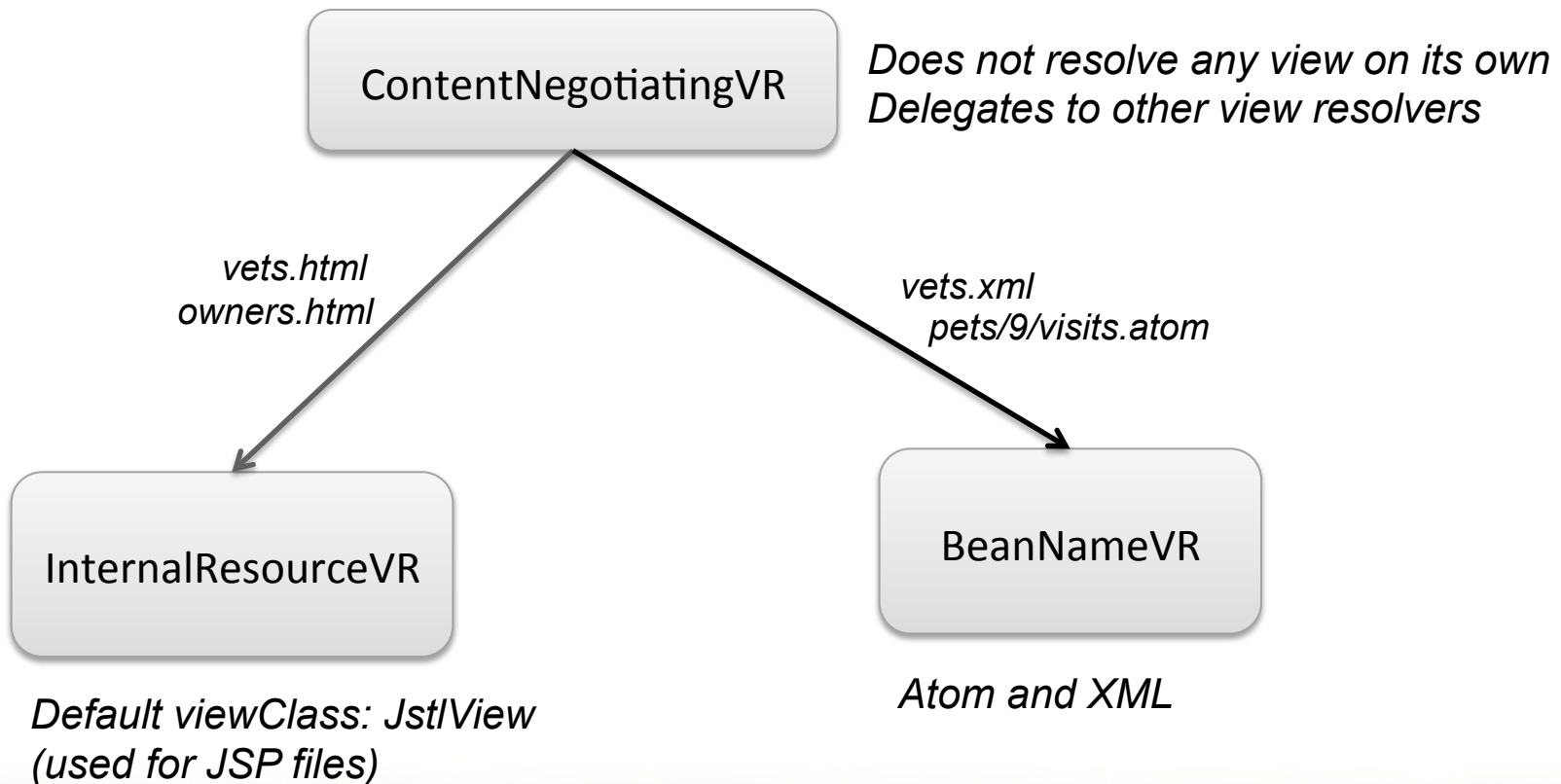
<http://blog.springsource.org/2012/05/23/understanding-proxy-usage-in-spring/>

# Topics

- Core Spring
- The Web layer
  - Spring MVC
  - Third-party web libraries

# View Resolvers in spring-petclinic

mvc-view-config.xml



# Topics

- Core Spring
- The Web layer
  - Spring MVC
  - Third-party web libraries
    - Dandelion for datatables
    - Webjars

# Datatables in Spring MVC

- Based on project Dandelion
  - <http://dandelion.github.com/>

```
<datatables:table data="${userList}" id="dataTable">
  <datatables:column title="First Name"
    property="firstName" sortable="true" />

  <datatables:column title="Last Name"
    property="lastName" sortable="true" />

</datatables:table>
```

*JSP file*

# Dandelion is based on jQuery Datatables and Bootstrap

- Click, sort, scroll, next/previous...
- Bootstrap theme
- PDF export...

Search:

PDF

Name ▲	Address ◆	City ◆	Telephone ◆	Pets ◆
Betty Davis	638 Cardinal Ave.	Sun Prairie	6085551749	Basil
Carlos Estaban	2335 Independence La.	Waunakee	6085555487	Lucky Sly
David Schroeder	2749 Blackhawk Trail	Madison	6085559435	Freddy
Eduardo Rodriguez	2693 Commerce St.	McFarland	6085558763	Jewel Rosy

# Topics







- Core Spring
- The Web layer
  - Spring MVC
  - Third-party web libraries
    - Dandelion
    - WebJars



# Webjars

- Allow CSS and JS libraries to be imported as Maven libraries
  - Used in Petclinic for jQuery, jQuery-ui, datatables, Bootstrap
  - <http://www.webjars.org/>

## List of WebJars

Name	Build Tool: SBT / Play 2 / Maven
<a href="#">jquery-colorhelpers</a> 1.1 	"org.webjars" % "jquery"
<a href="#">explorercanvas</a> r3 	"org.webjars" % "explor"
<a href="#">highstock</a> 1.2.4-1 1.2.4 	"org.webjars" % "highst"
<a href="#">jquery-history</a> 0.1 	"org.webjars" % "jquery"
<a href="#">wiki</a> 0.95 	"org.webjars" % "wiki"
<a href="#">highcharts</a> 2.3.3 	"org.webjars" % "highct"

# Webjars

```
<dependency>
  <groupId>org.webjars</groupId>
  <artifactId>jquery-ui</artifactId>
  <version>1.9.1</version>
</dependency>
```

*pom.xml*

- ▶  jquery-ui-1.9.1.jar - M2\_REPO/org/webjars/
- ▶  junit-4.10.jar - M2\_REPO/junit/junit/4.10 - ,
- ▶  junit-dep-4.10.jar - M2\_REPO/junit/junit-de
- ▶  log4j-1.2.14.jar - M2\_REPO/log4j/log4j/1.2.
- ▶  mysql-connector-java-5.1.18.jar - M2\_REPO

# Using Webjars

- Inside pom.xml

```
<dependency>  
  <groupId>org.webjars</groupId>  
  <artifactId>jquery-ui</artifactId>  
  <version>1.9.1</version>  
</dependency>
```

## Spring configuration

```
<mvc:resources mapping="/webjars/**"  
  location="classpath:/META-INF/  
  resources/webjars/" />
```

- Inside JSP

```
<link rel="stylesheet" href="/webjars/jquery-ui/1.9.1/js/jquery-  
ui-1.9.1.custom.js">
```

◦ js file is inside a jar file!

# References

- Series of 5 blog entries from Julien Dubois on how to “improve performance of the Spring-Petclinic application”
  - <http://blog.ippon.fr/2013/03/11/>
- Dandelion: clickable/sortable datatables inside Spring-Petclinic
  - <http://dandelion.github.io/blog/2013/04/24/Integrating-Dandelion-DataTables-in-the-Spring-Petclinic-app>