

[subscribe to entries \(http://stsmmedia.net/feed/\)](http://stsmmedia.net/feed/) or [subscribe to comments \(http://stsmmedia.net/comments/feed/\)](http://stsmmedia.net/comments/feed/)

[StSMedia \(http://stsmmedia.net/\)](http://stsmmedia.net/) Programming, Photography, Life

• [Home](#)



[\(http://stsmmedia.net/\)](http://stsmmedia.net/)

- [Life \(http://stsmmedia.net/category/life/\)](http://stsmmedia.net/category/life/)
- [Photography \(http://stsmmedia.net/category/photography/\)](http://stsmmedia.net/category/photography/)
 - [Animals \(http://stsmmedia.net/category/photography/animals/\)](http://stsmmedia.net/category/photography/animals/)
 - [City \(http://stsmmedia.net/category/photography/city/\)](http://stsmmedia.net/category/photography/city/)
 - [Equipment \(http://stsmmedia.net/category/photography/equipment/\)](http://stsmmedia.net/category/photography/equipment/)
 - [Nature \(http://stsmmedia.net/category/photography/nature/\)](http://stsmmedia.net/category/photography/nature/)
 - [Sports \(http://stsmmedia.net/category/photography/sports/\)](http://stsmmedia.net/category/photography/sports/)
- [Programming \(http://stsmmedia.net/category/programming/\)](http://stsmmedia.net/category/programming/)
 - [Dojo \(http://stsmmedia.net/category/programming/dojo/\)](http://stsmmedia.net/category/programming/dojo/)
 - [Java \(http://stsmmedia.net/category/programming/java/\)](http://stsmmedia.net/category/programming/java/)
 - [Roo \(http://stsmmedia.net/category/programming/roo/\)](http://stsmmedia.net/category/programming/roo/)
 - [Spring \(http://stsmmedia.net/category/programming/spring/\)](http://stsmmedia.net/category/programming/spring/)
- [About Me \(http://stsmmedia.net/about-me/\)](http://stsmmedia.net/about-me/)
- [Contact \(http://stsmmedia.net/contact/\)](http://stsmmedia.net/contact/)

Search

Spring Finance > Part 2: Spring @MVC & Spring 3.0 REST integration

This entry is part 2 of 6 in the series [Spring Finance \(http://stsmmedia.net/series/spring-finance/\)](http://stsmmedia.net/series/spring-finance/)

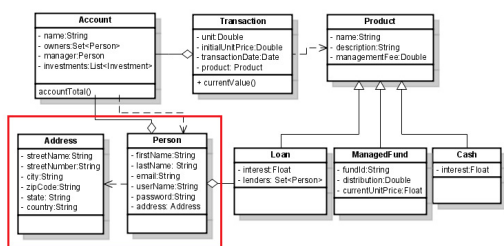
[Spring Finance \(http://stsmmedia.net/series/spring-finance/\)](http://stsmmedia.net/series/spring-finance/)

- [Spring Finance > Part 1: Introduction & Architecture \(http://stsmmedia.net/spring-finance-part-1-introduction-architecture/\)](http://stsmmedia.net/spring-finance-part-1-introduction-architecture/)
- Spring Finance > Part 2: Spring @MVC & Spring 3.0 REST integration
- [Spring Finance > Part 3: DDD, JPA & Transaction Support \(http://stsmmedia.net/spring-finance-part-3-ddd-jpa-transaction-support/\)](http://stsmmedia.net/spring-finance-part-3-ddd-jpa-transaction-support/)
- [Spring Finance > Part 4: Spring JS, Dojo & Bean Validation \(http://stsmmedia.net/spring-finance-part-4-spring-js-dojo-bean-validation/\)](http://stsmmedia.net/spring-finance-part-4-spring-js-dojo-bean-validation/)
- [Spring Finance > Part 5: Spring Web Flow Integration \(http://stsmmedia.net/spring-finance-part-5-spring-web-flow-integration/\)](http://stsmmedia.net/spring-finance-part-5-spring-web-flow-integration/)
- [Spring Finance > Part 6: Spring Security 3.0 Integration \(http://stsmmedia.net/spring-finance-part-6-spring-security-3-integration/\)](http://stsmmedia.net/spring-finance-part-6-spring-security-3-integration/)
- [Spring Finance > Part 7: Adding support for JSON and XML views \(http://stsmmedia.net/spring-finance-part-7-adding-support-for-json-and-xml-views/\)](http://stsmmedia.net/spring-finance-part-7-adding-support-for-json-and-xml-views/)

After a brief introduction of the Spring Finance Manager application in [part 1 \(http://stsmmedia.net/spring-finance-part-1-introduction-architecture/\)](http://stsmmedia.net/spring-finance-part-1-introduction-architecture/) of this series, I am now discussing the technical aspects of the sample application. The main focus of this part is on the new REST features introduced since the first milestone release of Spring 3.0.

•

[\(/media/spring-finance/class_diagram_part_1.png\)](/media/spring-finance/class_diagram_part_1.png)



To keep things clear I am not providing code for the complete application (as laid out in the initial class diagram) in this article but rather just a cut-down version of the application. I decided to create a Spring MVC application with only one controller, one service, and two domain objects, Person and Address.

Project Code Repository & Build Instructions

I have created a project over at Google code to host my SVN repository. The project can be found [here \(http://code.google.com/p/spring-finance-manager/\)](http://code.google.com/p/spring-finance-manager/) and the SVN repository is available via:

svn checkout <http://spring-finance-manager.googlecode.com/svn/FinanceManager> -0.1/

Detailed build instructions are available on [this \(http://code.google.com/p/spring-finance-manager/wiki/BuildAndInstall\)](http://code.google.com/p/spring-finance-manager/wiki/BuildAndInstall) Wiki page.

Spring @MVC

As mentioned before, the application uses Spring @MVC with annotations. Taking a look at PersonController.java we can see that a controller in Spring MVC is actually nothing more than a POJO with a few annotations which register this controller in the Spring application context as a controller:

```
01. @Controller
02. @RequestMapping("/person/**")
03. public class PersonController {
04.
05.     @Autowired
06.     private PersonService personService;
```

```
08. @RequestMapping(value = "person/{id}", method = RequestMethod.GET)
09. public String show(@PathVariable("id") Long id, ModelMap modelMap) {
10.     Assert.notNull(id, "Identifier must be provided.");
11.     modelMap.addAttribute("person", personService.find(id));
12.     return "person/show";
13. }
14. //further methods omitted
15. }
```

Most likely, you will be familiar with the [@Controller](http://static.springframework.org/spring/docs/2.5.x/reference/mvc.html#mvc-annotation) annotation. This annotation registers the controller class in the Spring application context as an MVC controller. The only configuration required to make Spring aware of Pojo's annotated with @Controller is captured in the FinanceManager-servlet.xml configuration:

```
1. <context:component-scan base-package="net.stsmidia.financemanager"
2.   use-default-filters="false">
3.   <context:include-filter expression="org.springframework.stereotype.Controller"
4.     type="annotation"/>
5. </context:component-scan>
```

Apart from the @Controller annotation, I have used the @RequestMapping annotation to define a URI-to-class mapping. This maps this controller to the /FinanceManager/person/ path and all URIs under it. If you are unfamiliar with Spring's @MVC model you can take a look at Juergen Hoellers blog post entitled ['Annotated Web MVC Controllers in Spring 2.5'](http://blog.springsource.com/2007/11/14/annotated-web-mvc-controllers-in-spring-25/) (<http://blog.springsource.com/2007/11/14/annotated-web-mvc-controllers-in-spring-25/>).

After injecting the PersonService using the [@Autowired](http://static.springframework.org/spring/docs/2.5.x/reference/beans.html#beans-autowired-annotation) annotation, we are ready to take a look at the show() method shown above. This method is used to retrieve details of a specific person in REST style.

Spring 3.0 REST

Again, we use the @RequestMapping annotation to define a URI-to-method mapping. Further, a RequestMethod is defined as part of the @RequestMapping annotation to specify the HTTP request method the show method should apply to. So in this case a HTTP GET request to /FinanceManager/person/1 would return the desired data provided a person with id=1 exists. The Spring Dispatcher servlet supports the following HTTP request methods: GET, HEAD, POST, PUT and DELETE.

Spring 3.0 introduces the @PathVariable annotation allows convenient binding of a variable supplied in the request URI. The @PathVariable is the heart of the new Spring 3.0 REST features. In the code shown above the path variable designated as {id} is automatically mapped to a Long object (@PathVariable("id") Long id) thanks to the default Spring property editor support.

The only thing left is to query the personService.find(id) and to insert the result into a ModelMap object. As you have noticed, the show() method returns only a String which indicates the location of the view to which the results are to be rendered. An InternalResourceViewResolver is used to complete the mapping from "person/show" to "/WEB-INF/jsp/person/show.jsp":

```
1. <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver"
2.   p:prefix="/WEB-INF/jsp/" p:suffix=".jsp"/>
```

Let's take a look at the update.jsp page:

```
1. <form:form action="/FinanceManager/person/${person.id}"
2.   method="PUT" modelAttribute="person">
3.   [...]
4. </form:form>
```

You notice that the HTTP method is defined as method="PUT". Since this is not directly supported by the Web browser (most Web browsers only support GET and POST), we need to register an additional listener in web.xml:

```
01. <filter>
02.   <filter-name>httpMethodFilter</filter-name>
03.   <filter-class>org.springframework.web.filter.HiddenHttpMethodFilter
04. </filter-class>
05. </filter>
06.
07. <filter-mapping>
08.   <filter-name>httpMethodFilter</filter-name>
09.   <servlet-name>FinanceManager</servlet-name>
10. </filter-mapping>
```

REST Mappings

In this section I wanted to elaborate a little bit on the [REST](http://en.wikipedia.org/wiki/Representational_State_Transfer) mappings I am following for my PersonController. In a typical Web application such as the Spring Finance Manager, four types of functionalities are exposed to the user. These are: **create** a new person (resource), **read** (or show) details of an existing person (resource), **update** a person (resource), and **delete** (remove) a person (resource). This is what we call [CRUD](http://en.wikipedia.org/wiki/Create,_read,_update_and_delete) functions for reading and manipulating the Person entity. When creating REST applications it is recommended to adhere to the following pattern which maps the CRUD functions for an entity to HTTP methods.

HTTP	CRUD
POST	CREATE
GET	READ
PUT	UPDATE
DELETE	DELETE

Apart from the standard CRUD functions an MVC controller also needs to supply form backing objects for the create.jsp and update.jsp forms for our Person entity. So for our PersonController the following REST mappings apply:

Resource	GET	PUT	POST	DELETE
Collection URI such as http://domain.com/financemanager/person/	List the members of the collection. For example list all the persons available in the system.	Not used.	Create a new person in the collection where the ID is assigned automatically by the collection.	Not used.
Member URI such as http://domain.com/financemanager/person/5	Retrieve the addressed person with id=5	Update the addressed person with id=5.	Not used.	Delete the addressed person with id=5.

Resource	GET	PUT	POST	DELETE
Member URI such as http://domain.com/financemanager/person/form	Create Form - returns an initialized, but empty person for form binding.	Not used.	Not used.	Not used.
Member URI such as http://domain.com/financemanager/person/5/form	Update Form returns the person resource which is pre-populated for form binding	Not used.	Not used.	Not used.

This table provides an overview of the REST mappings used in the Spring Finance Manager sample application. As you can see, the first two rows deal solely with the manipulation of resources while the last two rows of this table add some form mapping required when using Spring MVC. It would be possible to combine the forms for the create.jsp and update.jsp as they differ only in minor aspects. However, the REST mappings for the create form submits a HTTP POST message and the update form submits a HTTP PUT message. Hence, merging both forms require some more hacking in the JSPs and the controller.

Conclusion & Further Pointers

This article has demonstrated the general use of the Spring MVC framework with the (relatively) new Spring @MVC annotation style as well as the (brand new) REST capabilities of Spring @MVC. I have not shown how content negotiation works, such as delivering a XML, ATOM, RSS, or even a JSON document, however if someone presents a good use case how this fits into the Spring Finance Manager application I am happy to include a brief article as part of this series.

In the meantime, I would like to refer you to the excellent posts by my colleagues Arjen Poutsma and Alef Arendsen:

- **REST in Spring 3: @MVC** (<http://blog.springsource.com/2009/03/08/rest-in-spring-3-mvc/>): Arjen introduces main features of the Spring 3.0 REST support
- **Adding an Atom view to an application using Spring's REST support** (<http://blog.springsource.com/2009/03/16/adding-an-atom-view-to-an-application-using-springs-rest-support/>): Alef shows how to create an ATOM feed using Spring 3.0 REST content negotiation

Also, along with the upcoming release of Spring 3.0 M3 documentation for these features should be available.

In the next part of this series I am planning to explain the backend implementation of the Spring Finance Manager application. In particular, I am planning to express some brief thoughts on the current domain model and it's relationship to DDD. Further, I would like to discuss the integration of the sample application with the JPA API, Hibernate and the HSQL database.

Series Navigation

«**Spring Finance > Part 1: Introduction & Architecture** (<http://stsmmedia.net/spring-finance-part-1-introduction-architecture/>) **Spring Finance > Part 3: DDD, JPA & Transaction Support**» (<http://stsmmedia.net/spring-finance-part-3-ddd-jpa-transaction-support/>)

 (http://www.addtoany.com/share_save?sitename=StSMedia&siteurl=http%3A%2F%2Fstsmmedia.net%2F&linkname=Spring%20Finance%20%3E%20Part%20%3A%20Spring%20%40MVC%20%26%23038%3B%20Spring%203.0%20REST%20integration&linkurl=http%3A%2F%2Fstsmmedia.net%2Fspring-finance-part-2-spring-mvc-spring-30-rest-integration%2F)

Mar 22, 2009 | Filed under [Java](http://stsmmedia.net/category/programming/java/) (<http://stsmmedia.net/category/programming/java/>), [Programming](http://stsmmedia.net/category/programming/) (<http://stsmmedia.net/category/programming/>), [Spring](http://stsmmedia.net/category/programming/spring/) (<http://stsmmedia.net/category/programming/spring/>) and tagged with [Java](http://stsmmedia.net/tag/java/) (<http://stsmmedia.net/tag/java/>), [REST](http://stsmmedia.net/tag/rest/) (<http://stsmmedia.net/tag/rest/>), [Spring](http://stsmmedia.net/tag/spring/) (<http://stsmmedia.net/tag/spring/>), [Spring 3.0](http://stsmmedia.net/tag/spring-30/) (<http://stsmmedia.net/tag/spring-30/>), [Spring @MVC](http://stsmmedia.net/tag/spring-mvc/) (<http://stsmmedia.net/tag/spring-mvc/>).

Tags: [Java](http://stsmmedia.net/tag/java/) (<http://stsmmedia.net/tag/java/>), [REST](http://stsmmedia.net/tag/rest/) (<http://stsmmedia.net/tag/rest/>), [Spring](http://stsmmedia.net/tag/spring/) (<http://stsmmedia.net/tag/spring/>), [Spring 3.0](http://stsmmedia.net/tag/spring-30/) (<http://stsmmedia.net/tag/spring-30/>), [Spring @MVC](http://stsmmedia.net/tag/spring-mvc/) (<http://stsmmedia.net/tag/spring-mvc/>)

← (<http://stsmmedia.net/spring-finance-part-3-ddd-jpa-transaction-support/>)

→ (<http://stsmmedia.net/spring-finance-part-1-introduction-architecture/>)

15 Responses [Subscribe to comments](http://stsmmedia.net/spring-finance-part-2-spring-mvc-spring-30-rest-integration/feed/) (<http://stsmmedia.net/spring-finance-part-2-spring-mvc-spring-30-rest-integration/feed/>)

1.
[Spring Finance > Part 3: DDD, JPA & Transaction Support | StSMedia](http://stsmmedia.net/spring-finance-part-3-ddd-jpa-transaction-support/) (<http://stsmmedia.net/spring-finance-part-3-ddd-jpa-transaction-support/>)

[...] article is the exception to the rule :-). The code for this article was already realeased with the previous article on the Google code project website. This was needed to get get the sample application running to [...]

[Mar 29, 2009 @ 12:17 \(#comment-7\)](#)
2.
cease

Hi I downloaded your example app. When I try to serve static files it goes to continuous redirect loop. I'm deploying on google app engine. any Ideas ?

[Apr 21, 2009 @ 20:16 \(#comment-17\)](#)
3.
[sschmidt](http://stsmmedia.net/) (<http://stsmmedia.net/>)

Well, I guess I would need to see some logs :-).

I was planning to try it out myself one of these days but I am currently quite busy with work. So I won't get to it very soon.

Obviously you changed the persistence to use GAEs JPA or JDO support. Can you possibly share the changed sources and configurations for the complete application so I can deploy it and take a look myself? You could attach them to a bug ticket here: <http://code.google.com/p/spring-finance-manager/issues/list> (<http://code.google.com/p/spring-finance-manager/issues/list>) ?

Thanks.

Stefan

[Apr 22, 2009 @ 20:37 \(#comment-18\)](#)

4.

Mahesh Yamsani

Hi,

This was very helpful, at the same time it will be better for us, if you could provide some info about how to integrate rest in Spring dm Server(OSGi bundle development) module with a small example.

[May 22, 2009 @ 00:29 \(#comment-63\)](#)

5.

[sschmidt \(http://stsmidia.net\)](#)

Hi Mahesh,

This is on my TODO list, but the next part of this series will be looking at Spring Security.

The REST features depend on Spring 3 which is not supported by the current dmServer release 1.0.

However, there are milestone releases for it available so I would encourage you to try out the Spring Finance application there and gradually convert it to a bundle based solution. Alternatively, you can wait until the smServer sample applications have been updated for release 2.0 or wait for my blog post :).

Best,
Stefan

[May 22, 2009 @ 11:04 \(#comment-64\)](#)

6.

[Mahesh Yamsani \(http://maheshyamsani.blogspot.com\)](#)

Hi Stefan,

I just written a sample program for HelloWorld Rest API Using Spring 3.0 as a Web Module(OSGi). Please find it at

<http://maheshyamsani.blogspot.com/2009/05/rest-in-spring-30-sample-application-in.html> (<http://maheshyamsani.blogspot.com/2009/05/rest-in-spring-30-sample-application-in.html>)

[May 27, 2009 @ 16:50 \(#comment-68\)](#)

7.

[Prashant \(http://www.oplm.org\)](#)

This is a very good initiative.

1. First it helps understand how to put spring framework technologies together
2. Since you are releasing the source code, it actually reduces the ramp up time.

I appreciate your effort and thank you. Soon your blog will be on top of every chart of enterprise application development

[Jun 04, 2009 @ 00:50 \(#comment-69\)](#)

8.

[Jettro \(http://www.gridshore.nl\)](#)

Very nice post. I am using your sample application to create something as well. Is helping me a lot to understand things better.

thanks

[Jul 09, 2009 @ 23:47 \(#comment-81\)](#)

9.

[Arokia Raj \(http://www.anadocs.com\)](#)

hi mate,

Its great article to kick start the application..

Thanks for your service...

Hope to see more contribution from you...

Thanks,
Arokia

[Aug 07, 2009 @ 15:54 \(#comment-87\)](#)

10.

Peepod

Hi Mate,

Very nice tutorial series. I am working my way through from the beginning.

I have a question for you. Building and deploying version 0.1, I am seeing strange x and y values appearing in the links for show and update.

When I click on the show person link, the RESTful url appears as <http://localhost:8080/FinanceManager/person/4?x=8&y=7> (<http://localhost:8080/FinanceManager/person/4?x=8&y=7>).

When I click on the update person link, the RESTful url appears as <http://localhost:8080/FinanceManager/person/4/form?x=12&y=6> (<http://localhost:8080/FinanceManager/person/4/form?x=12&y=6>).

The actual values of x and y appear to be random, and moreover, they don't appear as input form parameters.

Therefore, my question for you is, where do these parameters come from? Some sort of debug/test code in spring M2?

[Aug 27, 2009 @ 19:04 \(#comment-106\)](#)

11.

[Stefan Schmidt \(http://twitter.com/schmidtstefan\)](#)

@Peepod,

Yes, these params appear to be added by dojo. I am not 100% sure but I think it has to do with back button support. I'll take a look if this can be disabled by any means.

-Stefan

[Aug 30, 2009 @ 10:31 \(#comment-108\)](#)

12.

dddavid

Hi Stefan,

I've imported your sample (Part 2) into SprinSource Tool Suite. It runs fine.

However, I didn't figure out how the hsqldb in-memory DB is started and how can I load some test data into the tables.

Thanks,
Dave

[Oct 02, 2009 @ 07:06 \(#comment-112\)](#)

13.

[Stefan Schmidt \(http://twitter.com/schmidtstefan\)](#)

@dddavid,

The in-memory DB is started with your application (when you deploy your war file in tomcat). Since it is in memory all data you entered will be lost as soon as you undeploy your application or shut down the container.

As an alternative you can use a DB which saves your data persistently (HSQLDB persistent, MySQL, ...). The connector setups for these are documented in the Spring framework documentation and fairly easy to setup since all the bits and pieces are already configured in this sample application.

Hope this helps..

-Stefan

[Oct 17, 2009 @ 12:25 \(#comment-118\)](#)

14.

Dennis

Hello Stefan,

First of all, happy new year!

After installing the application via maven, the created FinanceManager-0.5.war file worked fine after deploying it in Tomcat. However, when I imported the project into eclipse (I used STS), I keep on getting this error after building and deploying the project to STS to server...

org.springframework.beans.factory.BeanNotOfRequiredTypeException: Bean named 'conversionService' must be of type [org.springframework.core.convert.ConversionService], but was actually of type [net.stsmedia.financemanager.web.flow.ApplicationConversionService]

I haven't even modified the codes yet.

Thanks,
Dennis

[Dec 31, 2009 @ 22:21 \(#comment-141\)](#)

15.

dragos

hi Stefan,

in the controller you used a some requestmappings path I'm not used to.

First, the class annotation is @RequestMapping("/person/**") and then the methods @RequestMapping(value = "person", method =). Can you give me some details about this kind of mappings?

I'm using "/person" for class and "/create" or "/list" for methods.

Thanks

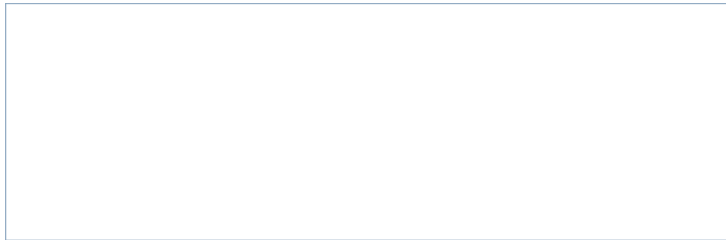
[Jan 04, 2010 @ 10:54 \(#comment-142\)](#)

Reply

Name (required)

Email (will not be published) (required)

Website



Archives

- [January 2010 \(http://stsmmedia.net/2010/01/\)](http://stsmmedia.net/2010/01/)
- [September 2009 \(http://stsmmedia.net/2009/09/\)](http://stsmmedia.net/2009/09/)
- [August 2009 \(http://stsmmedia.net/2009/08/\)](http://stsmmedia.net/2009/08/)
- [July 2009 \(http://stsmmedia.net/2009/07/\)](http://stsmmedia.net/2009/07/)
- [June 2009 \(http://stsmmedia.net/2009/06/\)](http://stsmmedia.net/2009/06/)
- [May 2009 \(http://stsmmedia.net/2009/05/\)](http://stsmmedia.net/2009/05/)
- [April 2009 \(http://stsmmedia.net/2009/04/\)](http://stsmmedia.net/2009/04/)
- [March 2009 \(http://stsmmedia.net/2009/03/\)](http://stsmmedia.net/2009/03/)
- [February 2009 \(http://stsmmedia.net/2009/02/\)](http://stsmmedia.net/2009/02/)
- [January 2009 \(http://stsmmedia.net/2009/01/\)](http://stsmmedia.net/2009/01/)
- [December 2008 \(http://stsmmedia.net/2008/12/\)](http://stsmmedia.net/2008/12/)
- [January 2008 \(http://stsmmedia.net/2008/01/\)](http://stsmmedia.net/2008/01/)

Tags

[Architecture \(http://stsmmedia.net/tag/architecture/\)](http://stsmmedia.net/tag/architecture/) [Australia \(http://stsmmedia.net/tag/australia/\)](http://stsmmedia.net/tag/australia/) [bush walking \(http://stsmmedia.net/tag/bush-walking/\)](http://stsmmedia.net/tag/bush-walking/) [DDD \(http://stsmmedia.net/tag/ddd/\)](http://stsmmedia.net/tag/ddd/) [Dojo \(http://stsmmedia.net/tag/dojo/\)](http://stsmmedia.net/tag/dojo/) [HDRI \(http://stsmmedia.net/tag/hdri/\)](http://stsmmedia.net/tag/hdri/) [Java \(http://stsmmedia.net/tag/java/\)](http://stsmmedia.net/tag/java/) [JavaScript \(http://stsmmedia.net/tag/javascript/\)](http://stsmmedia.net/tag/javascript/) [JMS \(http://stsmmedia.net/tag/jms/\)](http://stsmmedia.net/tag/jms/) [JPA \(http://stsmmedia.net/tag/jpa/\)](http://stsmmedia.net/tag/jpa/) [json \(http://stsmmedia.net/tag/json/\)](http://stsmmedia.net/tag/json/) [Kayak \(http://stsmmedia.net/tag/kayak/\)](http://stsmmedia.net/tag/kayak/) [Koala \(http://stsmmedia.net/tag/koala/\)](http://stsmmedia.net/tag/koala/) [Lens Review \(http://stsmmedia.net/tag/lens-review/\)](http://stsmmedia.net/tag/lens-review/) [ORM \(http://stsmmedia.net/tag/orm/\)](http://stsmmedia.net/tag/orm/) [OXM \(http://stsmmedia.net/tag/oxm/\)](http://stsmmedia.net/tag/oxm/) [Photography \(http://stsmmedia.net/tag/photography/\)](http://stsmmedia.net/tag/photography/) [REST \(http://stsmmedia.net/tag/rest/\)](http://stsmmedia.net/tag/rest/) [Roo \(http://stsmmedia.net/tag/roo/\)](http://stsmmedia.net/tag/roo/) [Skyline \(http://stsmmedia.net/tag/skyline/\)](http://stsmmedia.net/tag/skyline/) [Spring \(http://stsmmedia.net/tag/spring/\)](http://stsmmedia.net/tag/spring/) [Spring 3.0 \(http://stsmmedia.net/tag/spring-30/\)](http://stsmmedia.net/tag/spring-30/) [Spring @MVC \(http://stsmmedia.net/tag/spring-mvc/\)](http://stsmmedia.net/tag/spring-mvc/) [Spring Security \(http://stsmmedia.net/tag/spring-security/\)](http://stsmmedia.net/tag/spring-security/) [Spring Security 3.0 \(http://stsmmedia.net/tag/spring-security-30/\)](http://stsmmedia.net/tag/spring-security-30/) [Spring Web Flow \(http://stsmmedia.net/tag/spring-web-flow/\)](http://stsmmedia.net/tag/spring-web-flow/) [Surfing \(http://stsmmedia.net/tag/surfing/\)](http://stsmmedia.net/tag/surfing/) [Sydney \(http://stsmmedia.net/tag/sydney/\)](http://stsmmedia.net/tag/sydney/) [Tennis \(http://stsmmedia.net/tag/tennis/\)](http://stsmmedia.net/tag/tennis/) [Transactions \(http://stsmmedia.net/tag/transactions/\)](http://stsmmedia.net/tag/transactions/) [Travel \(http://stsmmedia.net/tag/travel/\)](http://stsmmedia.net/tag/travel/) [USA \(http://stsmmedia.net/tag/usa/\)](http://stsmmedia.net/tag/usa/) [Validation \(http://stsmmedia.net/tag/validation/\)](http://stsmmedia.net/tag/validation/)

My Links

- [LinkedIn Profile \(http://www.linkedin.com/in/stefanschmidt\)](http://www.linkedin.com/in/stefanschmidt)
- [My Employer \(http://www.springsource.com/\)](http://www.springsource.com/)
- [My Picasa Web Albums \(http://picasaweb.google.com/stsmmedia\)](http://picasaweb.google.com/stsmmedia)
- [Twitter \(http://twitter.com/schmidtstefan\)](http://twitter.com/schmidtstefan)
- [Xing Profile \(http://www.xing.com/profile/Stefan_Schmidt118\)](http://www.xing.com/profile/Stefan_Schmidt118)

Recent Posts

- [USA Trip 2009 \(http://stsmmedia.net/usa-trip-2009/\)](http://stsmmedia.net/usa-trip-2009/)
- [Sydney Royal Botanical Gardens '09 \(http://stsmmedia.net/sydney-royal-botanical-gardens-09/\)](http://stsmmedia.net/sydney-royal-botanical-gardens-09/)
- [Spring Finance > Part 7: Adding support for JSON and XML views \(http://stsmmedia.net/spring-finance-part-7-adding-support-for-json-and-xml-views/\)](http://stsmmedia.net/spring-finance-part-7-adding-support-for-json-and-xml-views/)
- [Trip to La Perouse & Maroubra, Sydney \(http://stsmmedia.net/trip-to-la-perouse-maroubra-sydney/\)](http://stsmmedia.net/trip-to-la-perouse-maroubra-sydney/)
- [Sydney by Day \(http://stsmmedia.net/sydney-by-day/\)](http://stsmmedia.net/sydney-by-day/)

[What am I doing? \(http://twitter.com/schmidtstefan\)](http://twitter.com/schmidtstefan)

- No public Twitter messages.

Gallery



[\(http://stsmmedia.net/media/sydney-royal-botanical-gardens/_mg_8876.jpg\)](http://stsmmedia.net/media/sydney-royal-botanical-gardens/_mg_8876.jpg) [\(http://stsmmedia.net/media/medibank-international-sydney-2009/_mg_1709.jpg\)](http://stsmmedia.net/media/medibank-international-sydney-2009/_mg_1709.jpg) [\(http://stsmmedia.net/media/medibank-international-sydney-2009/_mg_2341.jpg\)](http://stsmmedia.net/media/medibank-international-sydney-2009/_mg_2341.jpg) [\(http://stsmmedia.net/media/sydney-harbour-qe2-and-qv/02240012.jpg\)](http://stsmmedia.net/media/sydney-harbour-qe2-and-qv/02240012.jpg)

[subscribe to entries \(http://stsmmedia.net/feed/\)](http://stsmmedia.net/feed/) or [subscribe to comments \(http://stsmmedia.net/comments/feed/\)](http://stsmmedia.net/comments/feed/)

All content © 2010 by StSMedia