

Aus dem Spring Nähkästchen

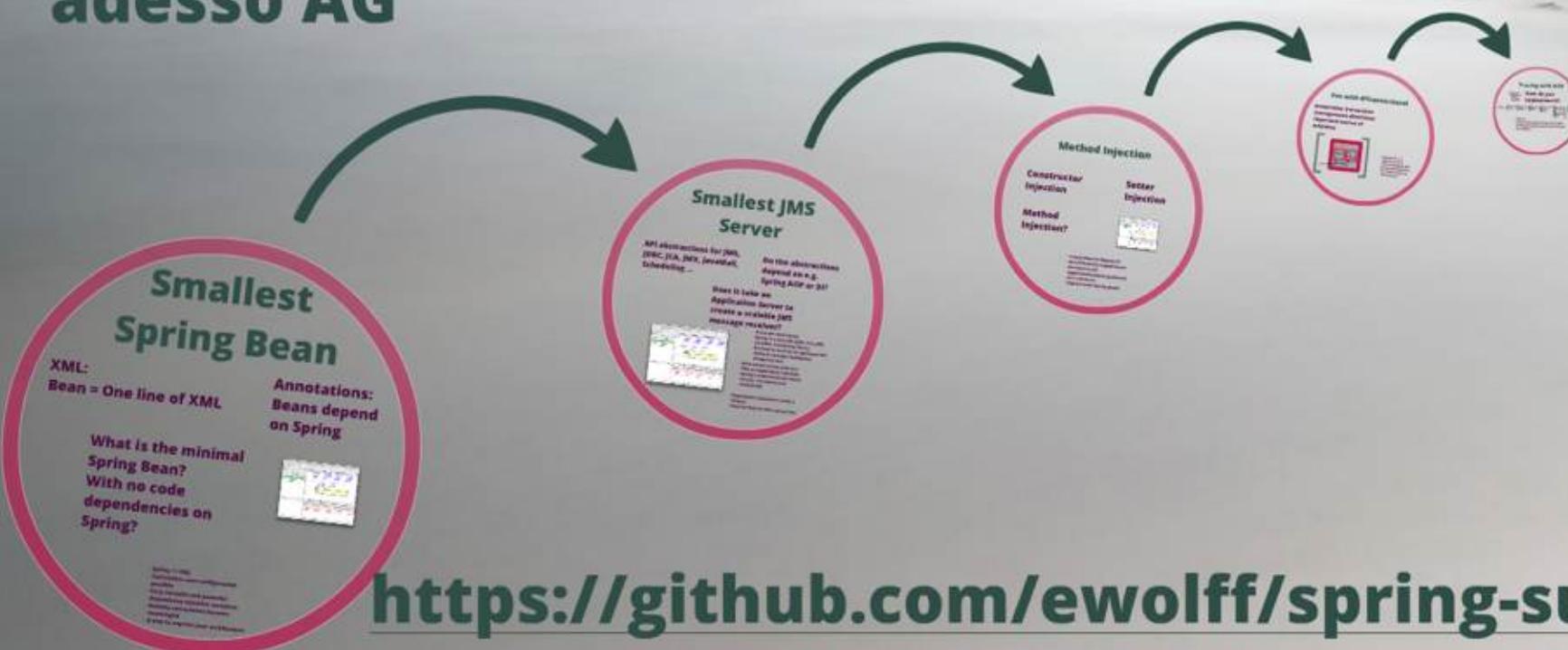
Eberhard Wolff

Twitter: @ewolff

Architecture & Technology

Manager

adesso AG



<https://github.com/ewolff/spring-surprises>

<https://github.com/ewolff/spring-jms-server>

Aus dem Spring Nähkästchen

Eberhard Wolff

Twitter: @ewolff

Architecture & Technology

Manager

adesso AG



<https://github.com/ewolff/spring-surprises>

<https://github.com/ewolff/spring-jms-server>

Aus dem Spring Nähk...

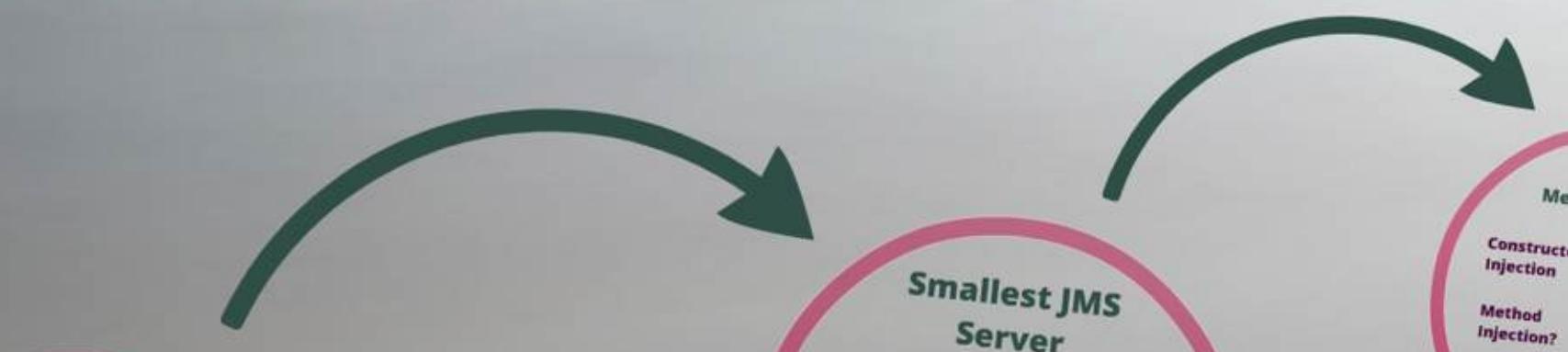
Eberhard Wolff

Twitter: @ewolff

Architecture & Technology

Manager

adesso AG





Me looking
at Spring

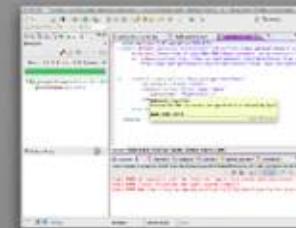
Smallest Spring Bean

XML:

Bean = One line of XML

Annotations:
Beans depend on Spring

**What is the minimal Spring Bean?
With no code dependencies on Spring?**



- Spring != XML
- Convention over configuration possible
- Very versatile and powerful Dependency Injection container
- Naming conventions become meaningful
- A way to express your architecture

http://lett

Spring |

XML:

Bean = One line of XML

**What is the minimum
Spring Bean?**

Annotations: Beans depend on Spring

**What is the minimal
Spring Bean?
With no code
dependencies on
Spring?**

Spring – spring-surprises/src/main/resources/smallbean.xml – Spring Tool Suite – /Users/wolff/Documents/wor...

Pa Ou Ty U J

Runs: 1/1 Errors: 0 Failures: 0

springjms.SpringJms [Runner: JUnit 4] (1) checkResponse (1,454 s)

spring-jms-server/po MyRepository.java smallbean.xml "7"

```
<?xml version="1.0" encoding="UTF-8"?>
<beans default-autowire="constructor" xmlns="http://www.springframework.org/
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context="htt
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://w
    http://www.springframework.org/schema/context http://www.springframe

<context:component-scan base-package="smallbean"
  use-default-filters="false">
  <context:include-filter type="regex"
    expression=". *Repository" />
  <con Attribute : expression
    Indicates the filter expression, the type of which is indicated by "type".
    Data Type : string
  </context:include-filter>
</context:component-scan>
</beans>
```

Source Namespaces Overview beans context Beans Graph

Console Markers Progress Servers Spring Explorer Problems

<terminated> SpringJms [JUnit] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/ji

log4j:WARN No appenders could be found for logger (org.apache.activemq.broker.jmx).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info

Writable Smart Insert 1:1

- **Spring != XML**
- **Convention over configuration possible**
- **Very versatile and powerful Dependency Injection container**
- **Naming conventions become meaningful**
- **A way to express your architecture**

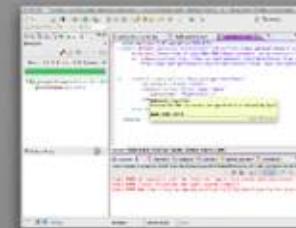
Smallest Spring Bean

XML:

Bean = One line of XML

Annotations:
Beans depend on Spring

**What is the minimal Spring Bean?
With no code dependencies on Spring?**



- Spring != XML
- Convention over configuration possible
- Very versatile and powerful Dependency Injection container
- Naming conventions become meaningful
- A way to express your architecture

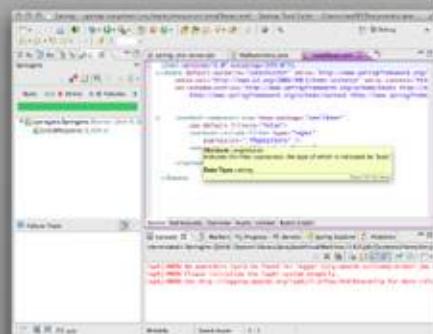
http://lett

Smallest JMS Server

**API abstractions for JMS,
JDBC, JCA, JMX, JavaMail,
Scheduling ...**

**Do the abstractions
depend on e.g.
Spring AOP or DI?**

**Does it take an
Application Server to
create a scalable JMS
message receiver?**



- Great API abstraction
- Spring is a nice JMS, JDBC, JCA, JMX, JavaMail, Scheduling library
- No need to wait for an optimized API
- Uniform concepts (templates, exceptions etc)

- Abstraction can be used w/o XML or Dependency Injection
- Spring's modularization works for you - no need to use everything!

- Dependency Injection is really a Pattern
- Hand-written DI: Not a great idea

Service

**API abstractions for JMS,
JDBC, JCA, JMX, JavaMail,
Scheduling ...**

**Does it take
Application**

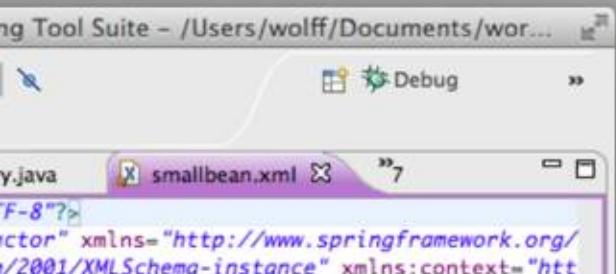
**Do the abstractions
depend on e.g.
Spring AOP or DI?**

**Take an
Application Server to**

Spring AOP

**Does it take an
Application Server to
create a scalable JMS
message receiver?**

- Great API abstraction
- Spring is a nice JMS, JDG, JavaMail, Scheduling library



Spring – spring-surprises/src/main/resources/smallbean.xml – Spring Tool Suite – /Users/wolff/Documents/wor...

Pa Ou Ty U J

Runs: 1/1 Errors: 0 Failures: 0

springjms.SpringJms [Runner: JUnit 4] (1)
checkResponse (1,454 s)

spring-jms-server/po MyRepository.java smallbean.xml "7"

```
<?xml version="1.0" encoding="UTF-8"?>
<beans default-autowire="constructor" xmlns="http://www.springframework.org/
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context="htt
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://w
    http://www.springframework.org/schema/context http://www.springframe

<context:component-scan base-package="smallbean"
  use-default-filters="false">
  <context:include-filter type="regex"
    expression=". *Repository" />
  <con Attribute : expression
    Indicates the filter expression, the type of which is indicated by "type".
    Data Type : string
  </context:include-filter>
</context:component-scan>
</beans>
```

Source Namespaces Overview beans context Beans Graph

Console Markers Progress Servers Spring Explorer Problems

<terminated> SpringJms [JUnit] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/ji

log4j:WARN No appenders could be found for logger (org.apache.activemq.broker.jmx).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info

Writable Smart Insert 1:1

scalable JMS

receiver?

- **Great API abstraction**
- **Spring is a nice JMS, JDBC, JCA, JMX, JavaMail, Scheduling library**
- **No need to wait for an optimized API**
- **Uniform concepts (templates, exceptions etc)**
- **Abstraction can be used w/o XML or Dependency Injection**
- **Spring's modularization works for you - no need to use**

- No need to wait for an optimizer
- Uniform concepts (templates, exceptions etc)
- Abstraction can be used w/o XML or Dependency Injection
- Spring's modularization works for you - no need to use everything!
- Dependency Injection is really a Pattern

TIME OF Dependency Injection

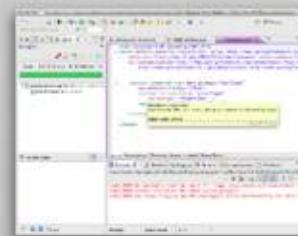
- **Spring's modularization works for you - no need to use everything!**
- **Dependency Injection is really a Pattern**
- **Hand-written DI: Not a great idea**

Method Injection

**Constructor
Injection**

**Setter
Injection**

**Method
Injection?**



- Create Domain Objects (?)
- Use differently scoped beans
- No need to call `ApplicationContext.getBean()`
- DI is universal
- Impress even Spring geeks

Constructor Injection

Setter Injection

Method Injection?

Spring – spring-surprises/src/main/resources/smallbean.xml – Spring Tool Suite – /Users/wolff/Documents/wor...

Pa Ou Ty U J

Runs: 1/1 Errors: 0 Failures: 0

springjms.SpringJms [Runner: JUnit 4] (1) checkResponse (1,454 s)

spring-jms-server/po MyRepository.java smallbean.xml "7"

```
<?xml version="1.0" encoding="UTF-8"?>
<beans default-autowire="constructor" xmlns="http://www.springframework.org/
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context="htt
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://w
    http://www.springframework.org/schema/context http://www.springframe

<context:component-scan base-package="smallbean"
  use-default-filters="false">
  <context:include-filter type="regex"
    expression=". *Repository" />
  <con Attribute : expression
    Indicates the filter expression, the type of which is indicated by "type".
    Data Type : string
  </context>
</beans>
```

Source Namespaces Overview beans context Beans Graph

Console Markers Progress Servers Spring Explorer Problems

<terminated> SpringJms [JUnit] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/ji

log4j:WARN No appenders could be found for logger (org.apache.activemq.broker.jmx).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info

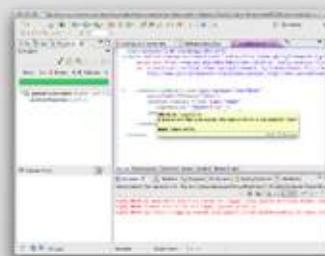
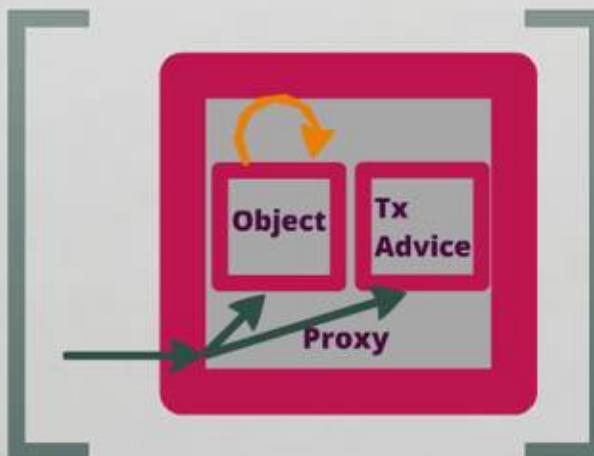
xml

Writable Smart Insert 1:1

- **Create Domain Objects (?)**
- **Use differently scoped beans**
- **No need to call
ApplicationContext.getBean()**
- **DI is universal**
- **Impress even Spring geeks**

Fun with @Transactional

Declarative transaction management eliminates important source of mistakes



- Spring AOP / Tx doesn't work on internal method calls
- @Transactional tests: No need to clean up environment

Fun with @Transactions

**Declarative transaction
management eliminates
important source of
mistakes**



Spring – spring-surprises/src/main/resources/smallbean.xml – Spring Tool Suite – /Users/wolff/Documents/wor...

Pa Ou Ty U J

Runs: 1/1 Errors: 0 Failures: 0

springjms.SpringJms [Runner: JUnit 4] (1) checkResponse (1,454 s)

spring-jms-server/po MyRepository.java smallbean.xml "7"

```
<?xml version="1.0" encoding="UTF-8"?>
<beans default-autowire="constructor" xmlns="http://www.springframework.org/
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context="htt
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://w
    http://www.springframework.org/schema/context http://www.springframe

<context:component-scan base-package="smallbean"
  use-default-filters="false">
  <context:include-filter type="regex"
    expression=". *Repository" />
  <con Attribute : expression
    Indicates the filter expression, the type of which is indicated by "type".
    Data Type : string
  </context>
</beans>
```

Source Namespaces Overview beans context Beans Graph

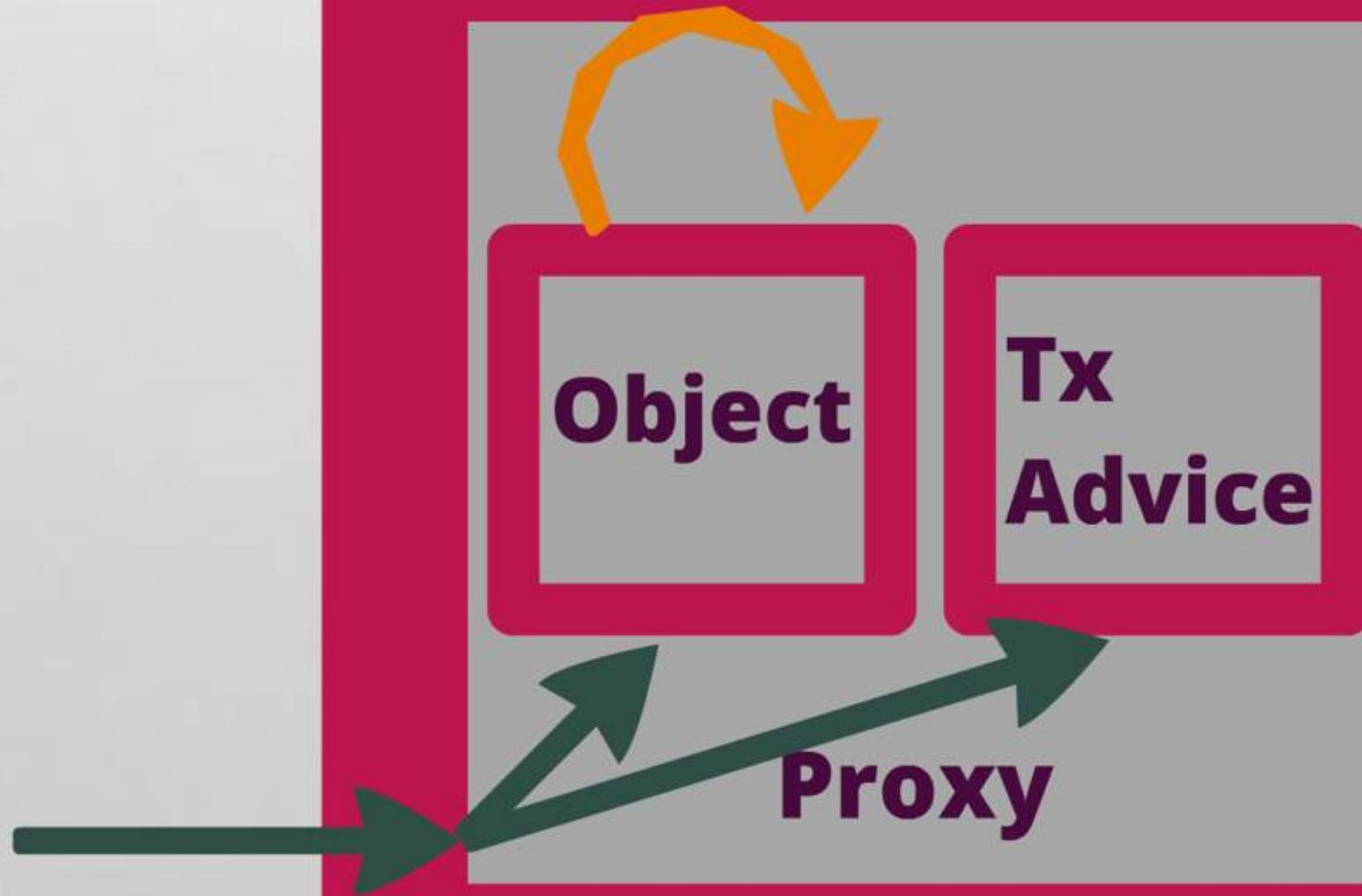
Console Markers Progress Servers Spring Explorer Problems

<terminated> SpringJms [JUnit] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/ji

log4j:WARN No appenders could be found for logger (org.apache.activemq.broker.jmx).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info

xml

Writable Smart Insert 1:1



- Spring AOP / Tx
doesn't work on
internal method calls
- @Transactional tests:
No need to clean up
environment

Tracing with AOP

How do you implement it?

Some Pointcuts

`execution(void hello())`
Execution of method hello
no parameters
void return type

`execution(* hello())`
Execution of method hello
no parameters
any return type

`execution(* hello(..))`
Execution of method hello
any parameters
any return type

`execution(* package.Class.*(..))`
Execution of any method in
package.Class
any parameters
any return type

`execution(* *.Repository.*(..))`
Execution of any method in any
package and all subpackages
Class name ends in Repository

`execution(* *.logging.*(..))`
Execution of any method in
package logging and all
subpackages
Any class name



- Look at
org.springframework.aop.interceptor
and **org.springframework.util** for some
nice helpers

Aspekt-Orientiertes
Programmieren
· Wie man diese Art
von Rat nutzt:
währenddessen,
davor, danach

Aspekt-Orientiertes Programmieren

- Wie man diese Art von Rat nutzt:
**währenddessen,
davor, danach**

racing with AOP

How do you implement it?

ution(* hello())
ution of method hello
parameters
return type

execution(* hello(..))
Execution of method hello
any parameters
any return type

execution(* package.Class.*(..))
Execution of any method in
package.Class
any parameters
any return type

execution(* *..*Repository)
Execution of any method
in package and all subpacka
Class name ends in Repos

execution(* logging..*.*(..))
Execution of any method

Some Pointcuts

execution(void hello())
Execution of method hello
no parameters
void return type

execution(* hello())

Execution of method hello

no parameters

any return type

execution(* hello(..))

**Execution of method hello
any parameters
any return type**

execution(* package.Class.*(..))

**Execution of any method in
package.Class**

any parameters

any return type

execution(* *..*Repository.*(..))

**Execution of any method in any
package and all subpackages
Class name ends in Repository**

execution(* logging..*.*(..))

Execution of any method in

execution(* logging.*.*(..))
Execution of any method in
package logging and all
subpackages
Any class name

Spring – spring-surprises/src/main/resources/smallbean.xml – Spring Tool Suite – /Users/wolff/Documents/wor...

Pa Ou Ty U J

Runs: 1/1 Errors: 0 Failures: 0

springjms.SpringJms [Runner: JUnit 4] (1) checkResponse (1,454 s)

spring-jms-server/po MyRepository.java smallbean.xml "7"

```
<?xml version="1.0" encoding="UTF-8"?>
<beans default-autowire="constructor" xmlns="http://www.springframework.org/
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context="htt
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://w
    http://www.springframework.org/schema/context http://www.springframe

<context:component-scan base-package="smallbean"
  use-default-filters="false">
  <context:include-filter type="regex"
    expression=".Repository" />
  <con Attribute : expression
    Indicates the filter expression, the type of which is indicated by "type".
    Data Type : string
  </context>
</beans>
```

Source Namespaces Overview beans context Beans Graph

Console Markers Progress Servers Spring Explorer Problems

<terminated> SpringJms [JUnit] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/ji

log4j:WARN No appenders could be found for logger (org.apache.activemq.broker.jmx).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info

xml

Writable Smart Insert 1:1

Implement it!

`execution(void hello())`
Execution of method hello
no parameters
void return type

`execution(* hello())`
Execution of method hello
no parameters
any return type

`execution(* hello(..))`
Execution of method hello
any parameters
any return type

`execution(* package.Class.*(..))`
Execution of any method in
package.Class
any parameters
any return type

`execution(* *..*Repository.*(..))`
Execution of any method in any
package and all subpackages
Class name ends in Repository

`execution(* logging..*.*(..))`
Execution of any method in
package logging and all
subpackages
Any class name

- Look at
org.springframework.aop.interceptor
and **org.springframework.util** for some
nice helpers

Aus dem Spring Nähkästchen

Eberhard Wolff

Twitter: @ewolff

Architecture & Technology

Manager

adesso AG



<https://github.com/ewolff/spring-surprises>

<https://github.com/ewolff/spring-jms-server>

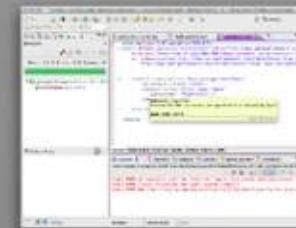
Smallest Spring Bean

XML:

Bean = One line of XML

Annotations:
Beans depend on Spring

**What is the minimal Spring Bean?
With no code dependencies on Spring?**



- Spring != XML
- Convention over configuration possible
- Very versatile and powerful Dependency Injection container
- Naming conventions become meaningful
- A way to express your architecture

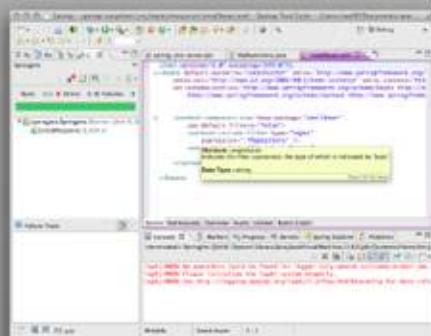
http://lett

Smallest JMS Server

**API abstractions for JMS,
JDBC, JCA, JMX, JavaMail,
Scheduling ...**

**Do the abstractions
depend on e.g.
Spring AOP or DI?**

**Does it take an
Application Server to
create a scalable JMS
message receiver?**



- Great API abstraction
- Spring is a nice JMS, JDBC, JCA, JMX, JavaMail, Scheduling library
- No need to wait for an optimized API
- Uniform concepts (templates, exceptions etc)

- Abstraction can be used w/o XML or Dependency Injection
- Spring's modularization works for you - no need to use everything!

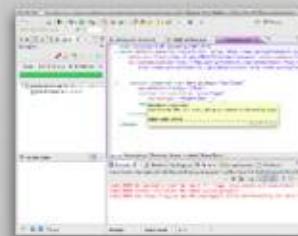
- Dependency Injection is really a Pattern
- Hand-written DI: Not a great idea

Method Injection

**Constructor
Injection**

**Setter
Injection**

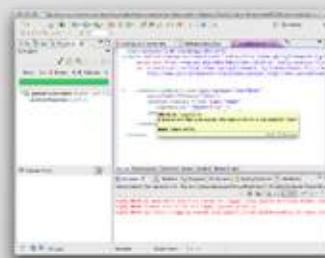
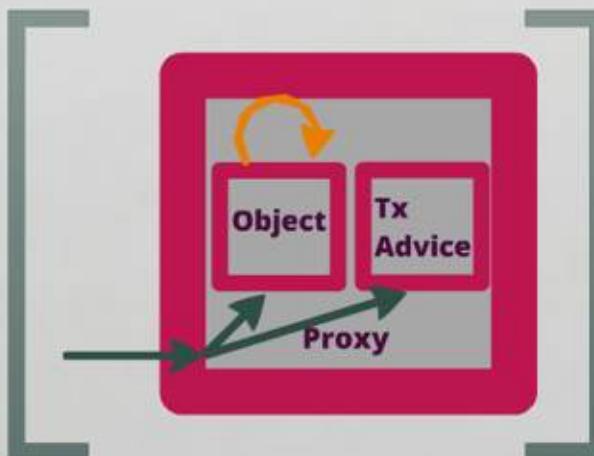
**Method
Injection?**



- Create Domain Objects (?)
- Use differently scoped beans
- No need to call `ApplicationContext.getBean()`
- DI is universal
- Impress even Spring geeks

Fun with @Transactional

Declarative transaction management eliminates important source of mistakes



- Spring AOP / Tx doesn't work on internal method calls
- @Transactional tests: No need to clean up environment

Tracing with AOP

How do you implement it?

Some Pointcuts

`execution(void hello())`
Execution of method hello
no parameters
void return type

`execution(* hello())`
Execution of method hello
no parameters
any return type

`execution(* hello(..))`
Execution of method hello
any parameters
any return type

`execution(* package.Class.*(..))`
Execution of any method in
package.Class
any parameters
any return type

`execution(* *.Repository.*(..))`
Execution of any method in any
package and all subpackages
Class name ends in Repository

`execution(* *.logging.*(..))`
Execution of any method in
package logging and all
subpackages
Any class name



- Look at
org.springframework.aop.interceptor
and **org.springframework.util** for some
nice helpers

Aspekt-Orientiertes
Programmieren
· Wie man diese Art
von Rat nutzt:
währenddessen,
davor, danach

Aus dem Spring Nähkästchen

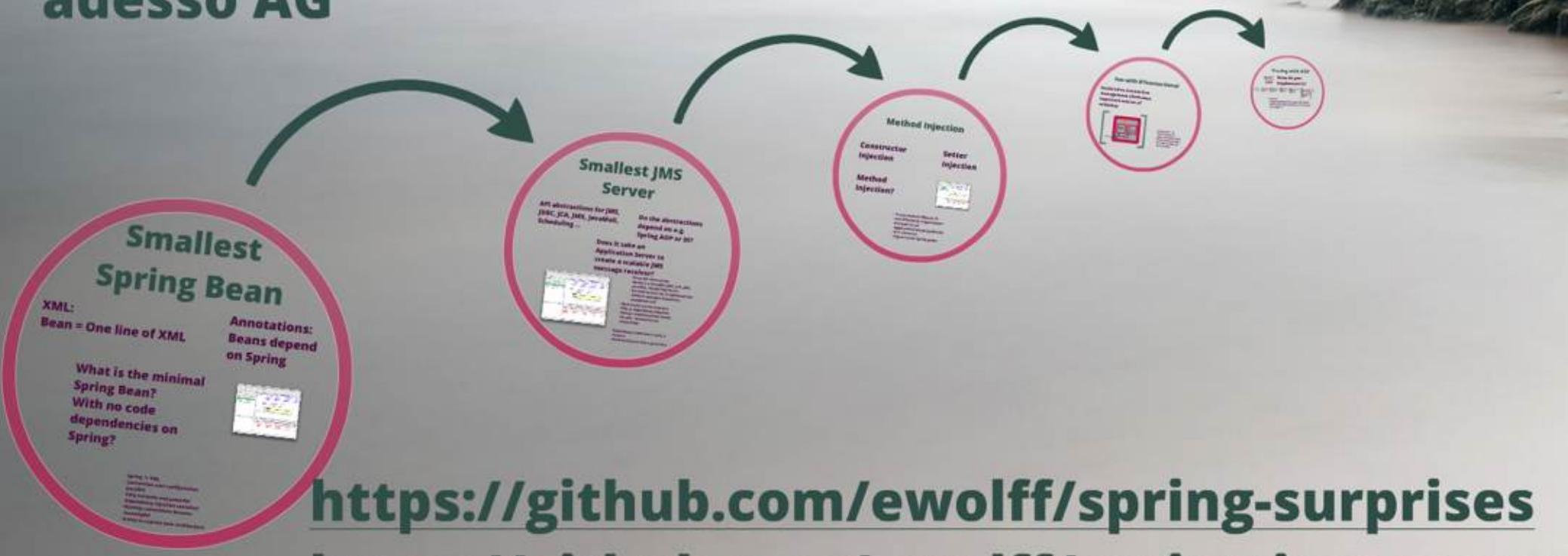
Eberhard Wolff

Twitter: @ewolff

Architecture & Technology

Manager

adesso AG



<https://github.com/ewolff/spring-surprises>

<https://github.com/ewolff/spring-jms-server>