



Menschen. Innovationen. Lösungen.

Grails - Die Suche ist vorbei

DOAG Konferenz 2011
Stefan Glase & Michael Stähler
am 17.11.2011



OPITZ CONSULTING

Wer sind wir?



Michael Stähler

Softwareentwickler
Java, Spring, Grails, Groovy

Trainer und Coach

Sprecher und Autor

twitter.com/fred4jupiter
github.com/fred4jupiter



Stefan Glase

Softwareentwickler
Java, Spring, Grails, Groovy

Trainer und Coach

Sprecher und Autor

twitter.com/stefanglase
github.com/codescape



Ihr ORACLE Center of Excellence

Leistungsangebot

- Java
- SOA/BPM
- ORACLE
- BI/DWH
- Outtasking
- Exadata
- Exalogic



Kunden / Kollegen

- Über 600 Kunden
- Branchenübergreifend
- Über 450 Kollegen
- An 8 Standorten

Tätigkeitsfelder

- IT-Strategie
- Beratung
- Implementierung
- Betrieb
- Training



Specialized
Oracle Database



Specialized
Service-Oriented Architecture



Specialized
Oracle Enterprise Linux



Specialized
Oracle Real Application
Clusters



Specialized
Oracle Business Intelligence
Foundation

Besuchen Sie auch die anderen Vorträge von OPITZ CONSULTING und unseren Stand (Nr. 236)!

Dienstag, 15. November 2011	Mittwoch, 16. November 2011	Donnerstag, 17. November 2011
MySQL in an Oracle driven datacenter 10:00 bis 10:45 Uhr, Raum Singapur	Das ungleiche Paar – Koexistenz von OWB und ODI 09:00 bis 09:45 Uhr, Raum Kopenhagen	Grails – Die Suche ist vorbei 09:00 bis 09:45 Uhr, Raum Riga
Oracle Forms meets BI 10:00 bis 10:45 Uhr, Raum Kiew	Praxis Knowhow: Skalierung von SOA Suite 11g Cluster 09:00 bis 09:45 Uhr, Raum Budapest	Enterprise Architecture Deliverables – Let's talk about results! 09:00 bis 09:45 Uhr, Raum Prag
Minimale Latenz – Bedarfsgerechte Bereitstellung von Daten im DWH 10:00 bis 10:45 Uhr, Raum Kopenhagen	RAC ONE Node 11.2.0.2. – Wo ist meine Instanz? 13:00 bis 13:45 Uhr, Raum St. Petersburg	Brückentechnologie – Min. Downtime Plattform-Migration / Upgrade von 9 nach 11.2 10:00 bis 10:45 Uhr, Raum St. Petersburg
ESSBASE und die OBIEE 11g – Aufbruch zu „echten“ OLAP-Analysen 12:00 bis 12:45 Uhr, Raum Helsinki	Oracle BAM – Die unentdeckten Möglichkeiten 13:00 bis 13:45 Uhr, Raum Oslo	Versteckte Schätze in BPM&SOA Suite 11g – gesammelte Projekterfahrungen 10:00 bis 10:45 Uhr, Raum Oslo
SOA Continuous Integration 12:00 bis 12:45 Uhr, Raum Riga	Neues zur Oracle Lizenzierung 15:00 bis 15:45 Uhr, Raum Kopenhagen	Unterbrechungsfreies Reporting: Hochverfügbarkeit von OWB bis BIEE 11g 12:00 bis 12:45 Uhr, Raum Stockholm
Agile BI mit OBIEE 11g 14:00 bis 14:45 Uhr, Raum Helsinki		Oracle Resource Management 13:00 bis 13:45 Uhr, Raum St. Petersburg
Die Crux mit dem Delta – vom Fullload zum Incremental Load 16:00 bis 16:45 Uhr, Raum Kopenhagen		Forms Legacy – ein ADF Panorama 14:00 bis 14:45 Uhr, Konferenzraum EG
Automatisiertes Konfigurationsmanagement mit Puppet 16:00 bis 16:45 Uhr, Konferenzraum EG		Disaster Recovery bei Grid Infrastructure 11.2 mit zwei Rechenzentren 15:00 bis 15:45 Uhr, Raum Hongkong
		Effizientere ETL mit Table Function 16:00 bis 16:45 Uhr, Raum Stockholm



youtube.com/opitzconsulting



slideshare.net/opitzconsulting



twitter.com/OC_WIRE



xing.com/group-51062.460375

Agenda

- Was ist Groovy?
- Was ist Grails?
- Live Coding

1

Was ist Groovy?

Was ist Groovy?

- **Dynamische Sprache für die Java Virtual Machine (JVM)**
- **Nahtlose Integration existierender Java Klassen und Bibliotheken**



- **Vereinfachtes Testen dank Power Asserts und Mocking**
- **Ausdrucksstarker Code durch kompaktere Syntax, Support für domänenspezifische Sprachen (DSLs), Closures**

Hello World mit Groovy

```
class Greeter {  
    def name  
    def greet() { "Hello $name!" }  
}
```

```
helloGroovy = new Greeter(name: 'Groovy')  
println helloGroovy.greet()
```


Groovy im Web ausprobieren



<http://groovyconsole.appspot.com/>

Objekte erstellen mit Groovy

```
class Person {  
    Long id  
    String firstName  
    String lastName  
}  
  
def person = new Person(  
    id: 1,  
    firstName: 'Fred',  
    lastName: 'Feuerstein'  
)  
  
assert person.id == 1  
assert person.firstName == 'Fred'  
assert person.lastName == 'Feuerstein'
```

AST-Transformationen mit Groovy

```
@groovy.transform.ToString
```

```
class Person {  
    Long id  
    String firstName  
    String lastName  
}
```

```
def person = new Person().with {  
    id = 1  
    firstName = 'Fred'  
    lastName = 'Feuerstein'  
    delegate  
}
```

```
assert 'Person(1, Fred, Feuerstein)' == person.toString()
```

Operationen auf Collections mit Groovy

```
class Person {
    Long id
    String firstName
    String lastName
}

def people = [
    new Person(id: 1, firstName: 'Fred', lastName: 'Feuerstein'),
    new Person(id: 2, firstName: 'Wilma', lastName: 'Feuerstein'),
    new Person(id: 3, firstName: 'Betty', lastName: 'Geröllheimer'),
    new Person(id: 4, firstName: 'Barney', lastName: 'Geröllheimer'),
    new Person(id: 5, firstName: 'Bam-Bam', lastName: 'Geröllheimer')]

assert ['Fred', 'Wilma'] ==
    people.findAll{it.lastName == 'Feuerstein'}.firstName
assert ['Feuerstein':2, 'Geröllheimer':3] == people.countBy{it.lastName}

people.findAll{it.lastName == 'Geröllheimer'}
    .each{ println "Hello $it.firstName!"}
```

Vereinfachtes File-Handling mit Groovy

```
def file = new File('myTemp.file')  
  
file.text = """Good day  
Guten Tag  
Buenos Dias"""  
  
file.eachLine { line, i -> println "$i: $line" }  
  
println file.text
```



2

Was ist Grails?

Was ist Grails?

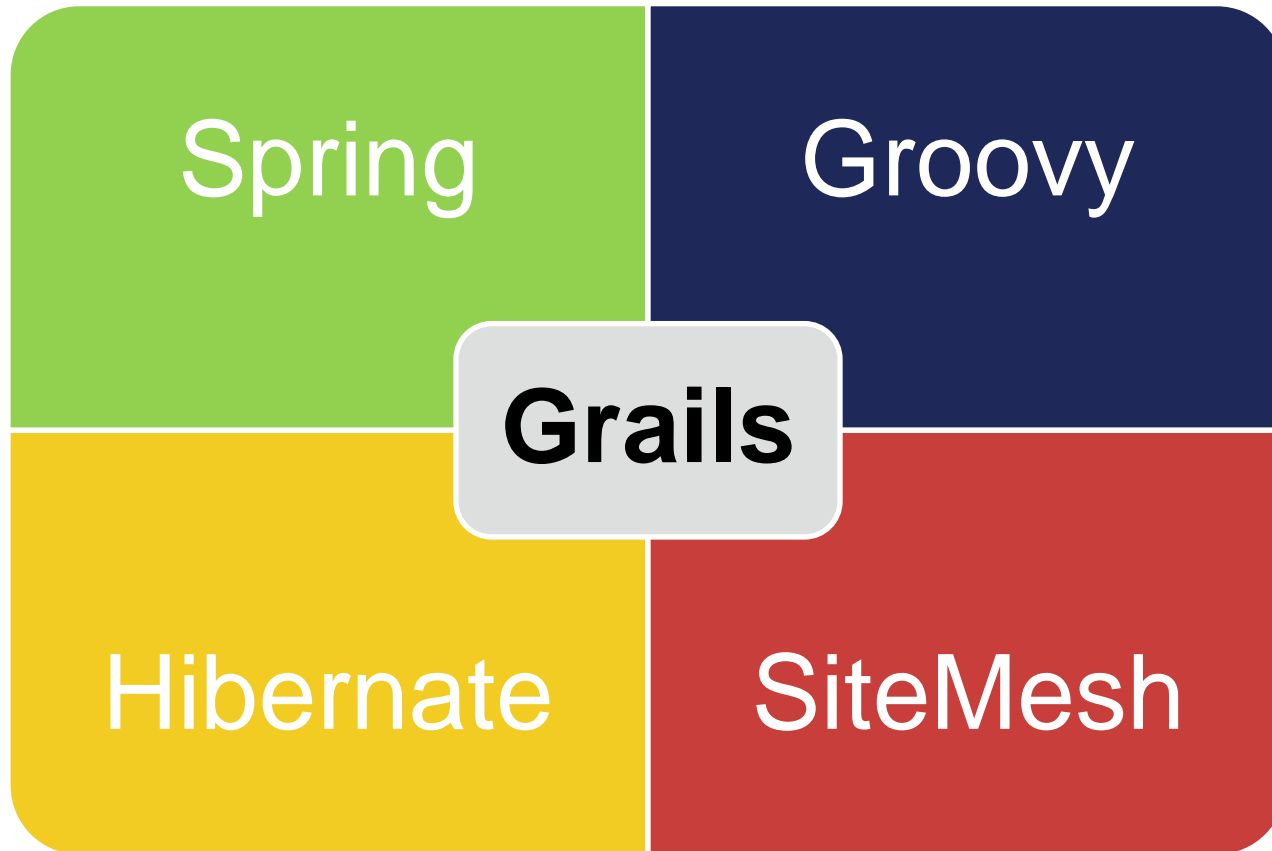


SpringSource über Grails...

Grails is an advanced and innovative **open source web application platform** that delivers new levels of developer productivity by applying principles like **Convention over Configuration**. Grails helps development teams embrace agile methodologies, deliver quality applications in reduced amounts of time, and focus on what really matters: creating **high quality, easy to use** applications that delight users. Grails naturally complements Java application development since it **is built on Spring** and **based on Groovy**, the leading dynamic language for the Java platform.

<http://www.springsource.com/developer/grails>








Ein solides Fundament



Konventionen in der Verzeichnisstruktur

```
grails-app
- conf
- controllers
- domain
- i18n
- services
- taglib
- utils
- views
lib
scripts
src
- groovy
- java
test
- integration
- unit
web-app
```

Konventionen bei der Konfiguration

 hibernate	06.10.2011 10:15	Dateiordner	
 spring	07.09.2011 10:12	Dateiordner	
 BootStrap.groovy	07.09.2011 10:12	GROOVY-Datei	1 KB
 BuildConfig.groovy	06.10.2011 10:31	GROOVY-Datei	2 KB
 Config.groovy	07.09.2011 10:12	GROOVY-Datei	4 KB
 DataSource.groovy	07.09.2011 10:12	GROOVY-Datei	2 KB
 UrlMappings.groovy	07.09.2011 10:12	GROOVY-Datei	1 KB

grails-app/conf

Fachklassen-Modellierung mit JPA

@Entity

```
public class Customer {
```

```
    private int id;  
    private String name;  
    private Collection<Order> orders;
```

@Id

```
    public int getId() { return id; }  
    public void setId(int id) { this.id = id; }
```

```
    public String getName() { return name; }  
    public void setName(String name) { this.name = name; }
```

@OneToMany(cascade=ALL, mappedBy="customer")

```
    public Collection<Order> getOrders() { return orders; }  
    public void setOrders(Collection<Order> orders) { this.orders = orders; }
```

```
}
```

<http://glassfish.java.net/javaee5/persistence/persistence-example.html>

Fachklassen-Modellierung mit Grails

```
class Customer {  
    String name  
    String email  
  
    static hasMany = [orders: Order]  
  
    static constraints = {  
        name(blank: false)  
        email(unique: true, email: true)  
    }  
  
    String toString() {  
        "$name ($email)"  
    }  
}
```

GORM = Grails Objekt Relational Mapping

- Objektrelationales Mapping als DSL
- **Dynamische Finder-Methoden**
- Dynamische Persistenz-Methoden
- Hibernate Criteria Builder als DSL
- Basierend auf Hibernate



Dynamische Finder-Methoden

```
// Erste Person mit dem Vornamen „Fred“  
Person.findByFirstName('Fred')
```

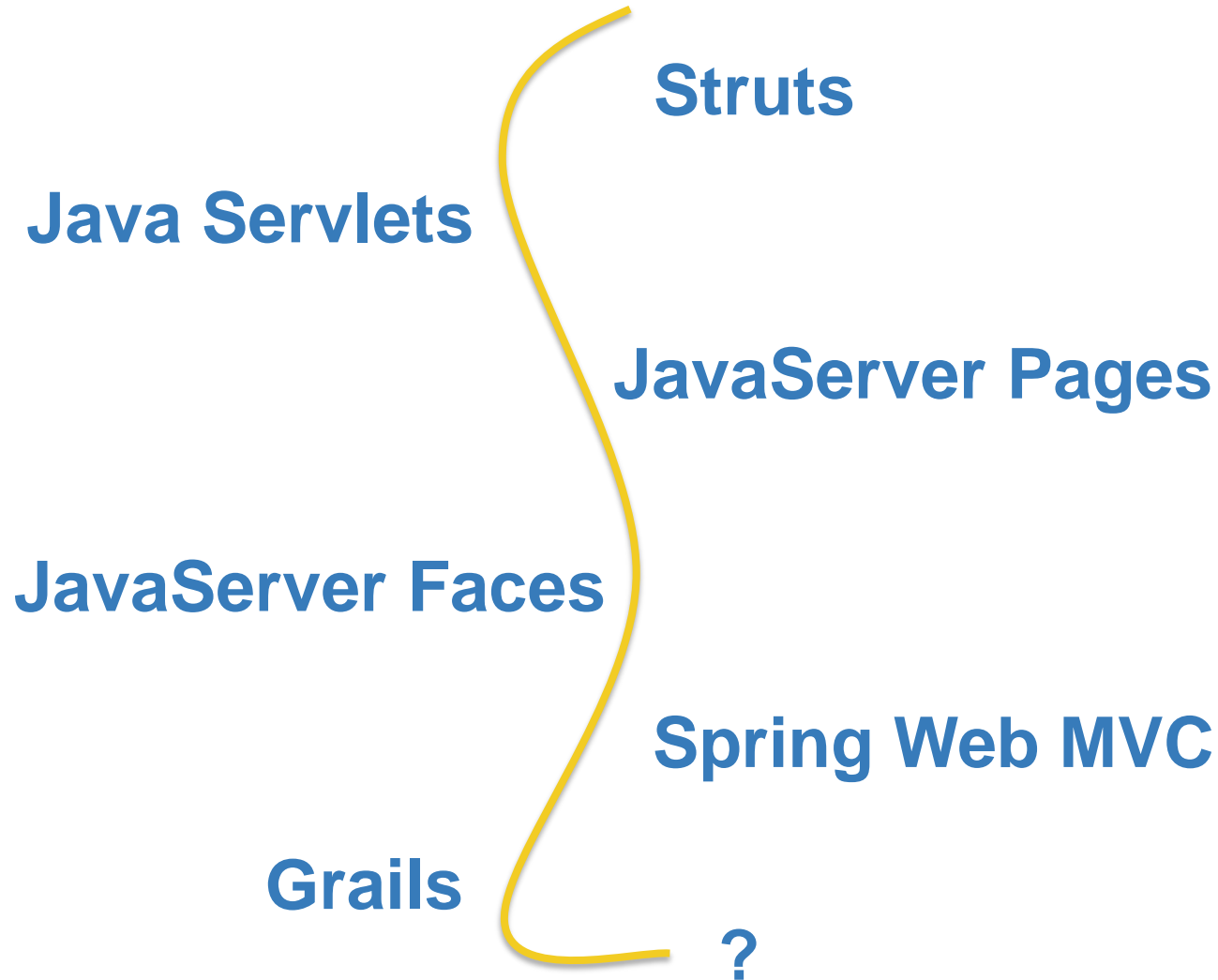
```
// Erste Person mit dem Vornamen „Fred“ und Nachnamen „Feuerstein“  
Person.findByFirstNameAndLastName('Fred', 'Feuerstein')
```

```
// Alle Personen im Alter zwischen 2 und 10 Jahren  
Person.findAllByAgeBetween(2, 10)
```

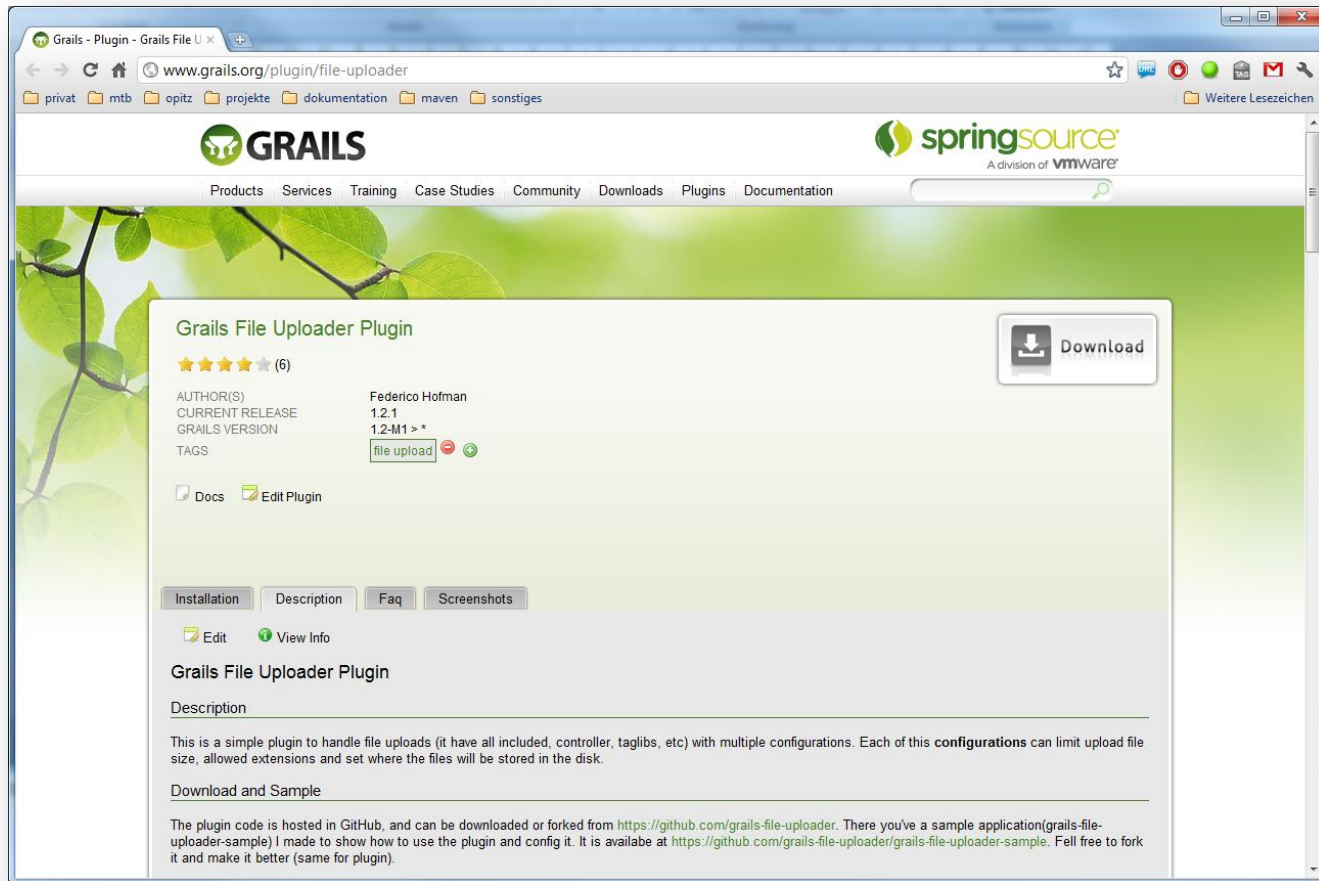
```
// Alle Personen mit einem hinterlegten Geburtstag  
Person.findAllByBirthdayIsNotNull()
```

```
// Alle Personen mit einem „B“ als ersten Buchstaben im Vornamen  
Person.findAllByFirstNameLike('B%')
```

Plug-In Beispiel: Datei-Upload



Plug-In Beispiel: Grails File Uploader Plugin



Plug-In-Mechanismus

All Plugins (690 Total)

A

- [acegi](#) - Acegi Plugin
- [activemq](#) - ActiveMQ Plugin
- [activiti](#) - Grails Activiti Plugin - Enabled Activiti BPM Suite...
- [activiti-spring-security](#) - Activiti Spring Security Integration
- [acts-as-taggable](#) - Acts As Taggable Plugin
- [address-lookup-zpfour](#) - Address Lookup via ZP4
- [ajax-uploader](#) - Ajax Uploader Plugin
- [ajaxflow](#) - Ajaxflow enables Ajaxified Webflows
- [akismet](#) - Akismet Anti-Spam Plugin
- [alfresco](#) - Alfresco DMS Integration
- [alfresco34](#) - Alfresco 3.4 integration
- [amazon-s3](#) - Amazon S3 Plugin
- [amazonfps](#) - Amazon Flexible Payments
- [aop-reloading-fix](#) - AOP Reloading Fix
- [apns](#) - Apple Push Notification Service Integration
- [app-engine](#) - Grails AppEngine plugin
- [app-info](#) - Application Info
- [appengine-gorm](#) - AppEngine GORM - Batch insert, update and delete op...
- [applet](#) - Applet Plugin
- [artisteerplugin](#) - Artisteer plugin
- [asynchronous-mail](#) - Asynchronous mail
- [atac](#) - Another Tag Cloud Plugin
- [atmosphere](#) - Grails Atmosphere Plugin
- [attachementable](#) - Attachmentable

MVC mit Grails

- Command-Objekte
- URL-Mappings mittels DSL
- Groovy Server Pages (GSPs)
- **Groovy Tag-Libraries**
- Scaffolding von CRUD-Anwendungen
- Internationalisierung (i18n)
- **Content Negotiation**
- Basierend auf Spring MVC

Content Negotiation

```
class TwitterController {  
    def status = {  
        withFormat {  
            json { render Status.list() as JSON }  
            xml { render Status.list() as XML }  
        }  
    }  
}
```

twitter/status.xml

```
<list>  
  <status id="1">  
    <dateCreated>2011-11-04 ...</dateCreated>  
    <message>My first post!</message>  
    <poster id="1"/>  
  </status>  
  <status id="2">  
    <dateCreated>2011-11-04 ...</dateCreated>  
    <message>Second message!</message>  
    <poster id="1"/>  
  </status>  
</list>
```

twitter/status.json

```
[{  
  "class": "grails4taglib.Status",  
  "id": 1,  
  "dateCreated": "2011-11-04T12:51:30Z",  
  "message": "My first post!",  
  "poster": {"class": "Person", "id": 1}  
}, {  
  "class": "grails4taglib.Status",  
  "id": 2,  
  "dateCreated": "2011-11-04T12:51:42Z",  
  "message": "Second message!",  
  "poster": {"class": "Person", "id": 1}  
}]
```

Groovy Tag Libraries

```
class TwitterTagLib {  
  
    static namespace = "twitter"  
  
    def displayStatus = { attrs, body ->  
        def status = attrs.status as Status  
        out << ""  
            <div class="status">  
                <span>${status.poster.username} posted on  
${g.formatDate(date: status.dateCreated)}:</span>  
                <p>${status.message}</p>  
            </div>""  
        }  
    }  
}
```

Definition

Verwendung

```
<twitter:displayStatus status="${status}"/>
```

3

Live Coding

Live Coding



Bitte Daumen drücken! 😊

Fazit



Im Netz...

Groovy & Grails:

<http://grails.org/>

<http://groovy.codehaus.org/>

Beispiele:

<https://github.com/codescape/presentations>

Twitter:

[@stefanglase](#)

Fragen und Antworten



Ihre Ansprechpartner

Stefan Glase, Senior Consultant

OPITZ CONSULTING Gummersbach GmbH

stefan.glase@opitz-consulting.com

Telefon +49 2261 60 01-1093



Michael Stähler, Senior Consultant

OPITZ CONSULTING Gummersbach GmbH

michael.steahler@opitz-consulting.com

Telefon +49 2261 60 01-1180



youtube.com/opitzconsulting



slideshare.net/opitzconsulting



xing.com/group-51062.460375