

GeoServer in action

Fortgeschrittene Möglichkeiten beim Einsatz des Geoservers

Nils Bühner

buehner@terrestris.de

terrestris GmbH & Co KG

Über uns

Nils Bühner



✉ buehner@terrestris.de  github.com/buehner

- Informatiker
- Java, Spring, Hibernate, Maven, Webtechnologien
- Entwickler bei [terrestris](#)

Über uns

terrestris.de



 [@terrestrisde](https://twitter.com/terrestrisde)  github.com/terrestris

- OpenSource GIS aus Bonn
- Projekte, Support, Schulung
- Beratung, Planung, Implementierung & Wartung

GeoServer



🐙 github.com/geoserver

- Java-basierter Server für Geodaten
- Standards des **OGC**
(z.B. WMS, WFS(-T), WPS)
- flexibel und erweiterbar
- gute Dokumentation (**user/dev**)



Server

-  Serverstatus
-  Protokollierung
-  Kontaktangaben
-  Über GeoServer

Daten

-  Layer-Vorschau
-  Arbeitsbereiche
-  Datenspeicher
-  Layer
-  Gruppenlayer
-  Stile

Dienste

-  WCS
-  WFS
-  WMS
-  WPS

Einstellungen

-  Global
-  JAI
-  Raster

Kartenkacheln-Cache

-  Gecachte Layer
-  Caching Standards
-  GridSets

- Konfiguration über Weboberfläche
- Arbeitsbereiche, Datenquellen, Layer, Stile
- OGC-Dienste
- Monitoring
(Serverstatus, Logs)
- erweiterte Features
(z.B. GeoWebCache)

Fragestellungen

Wie gehe ich mit dem GeoServer-Quellcode um?

Wie kann ich den GeoServer um Funktionalität erweitern?

Lässt sich der GeoServer auch programmatisch konfigurieren?

Wie kann der GeoServer für den Produktivbetrieb optimiert werden?

Technologien

maven.apache.org



- standardisierte Verwaltung von (Java-)Programmen
- Lebenszyklus der Software
- Validierung, Kompilierung, Paketierung, Installation und mehr

Technologien

git-scm.com



- verteilte Versionsverwaltung von Dateien
- nicht zwingend benötigt, aber sicher sinnvoll

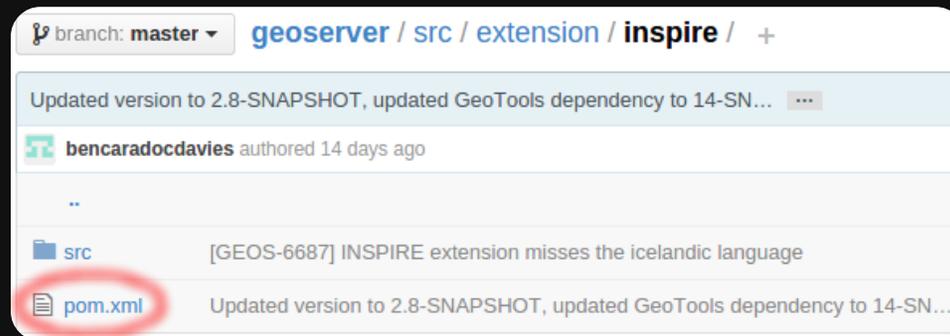
Quellcode auschecken

Mit *git*

```
git clone https://github.com/geoserver/geoserver.git
```

In ein Verzeichnis mit einer **pom.xml**
(Maven-Konfiguration) wechseln, z.B. zur *INSPIRE*-Erweiterung

```
cd geoserver/src/extension/inspire
```



Maven benutzen

```
mvn package
```

paketiert den Code zu einer `.jar`-Datei im Unterordner `target/`

```
user@osgeolive:~/src/extension/inspire/target$ ls -la *.jar
-rw-rw-r-- 1 user user 38077 Mär  6 13:20 gs-inspire-2.6.1.jar
-rw-rw-r-- 1 user user 24365 Mär  6 13:20 gs-inspire-2.6.1-sources.jar
-rw-rw-r-- 1 user user 13783 Mär  6 13:20 gs-inspire-2.6.1-tests.jar
-rw-rw-r-- 1 user user  6686 Mär  6 13:20 gs-inspire-2.6.1-test-sources.jar
```

Beinhaltet die *Validierung*, *Kompilierung* und das *Testen* des Codes.

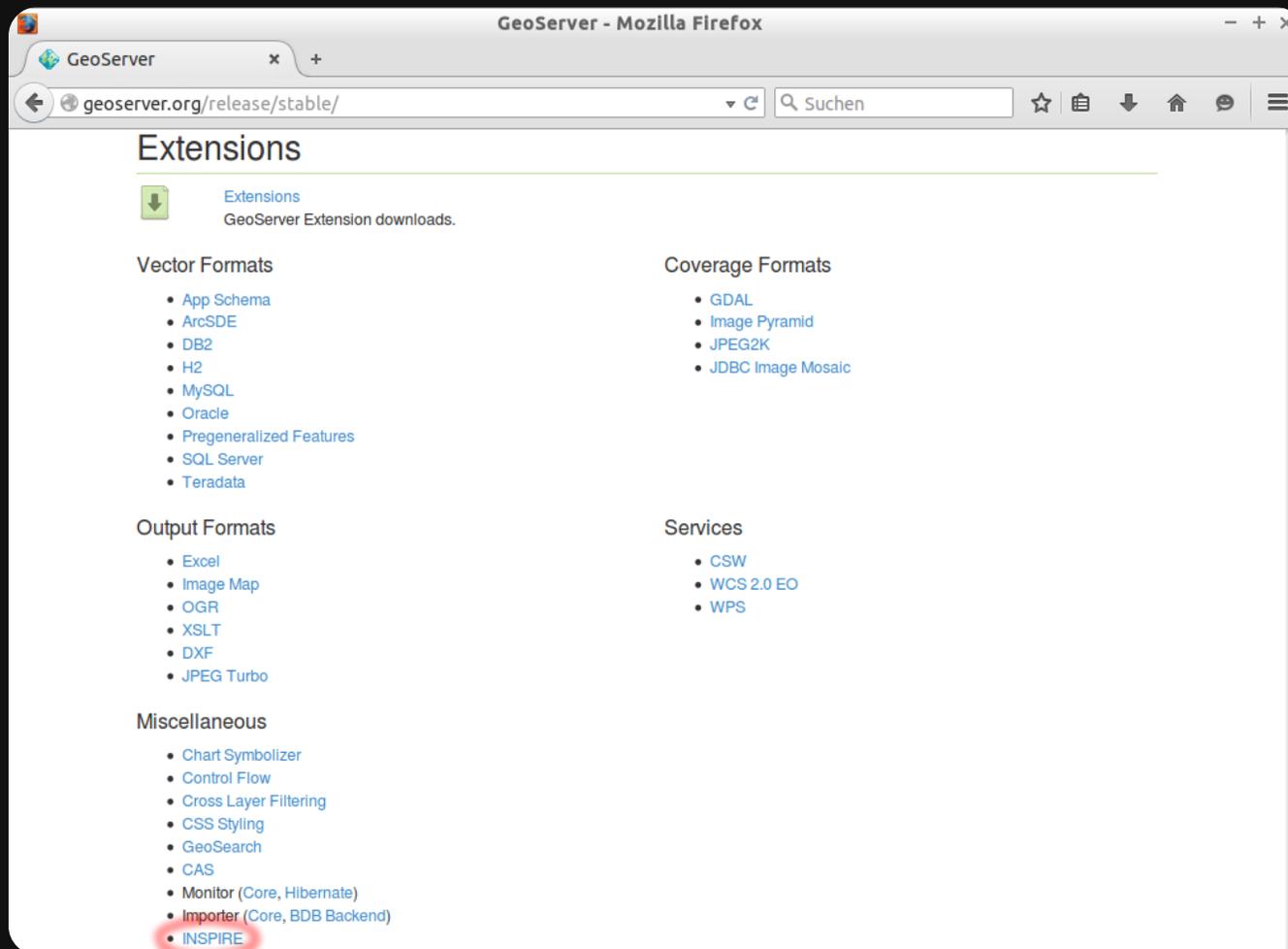
Schritte können auch einzeln ausgeführt werden, z.B.

```
mvn compile
```

Der Verzicht auf Tests beschleunigt den Prozess:

```
mvn package -DskipTests
```

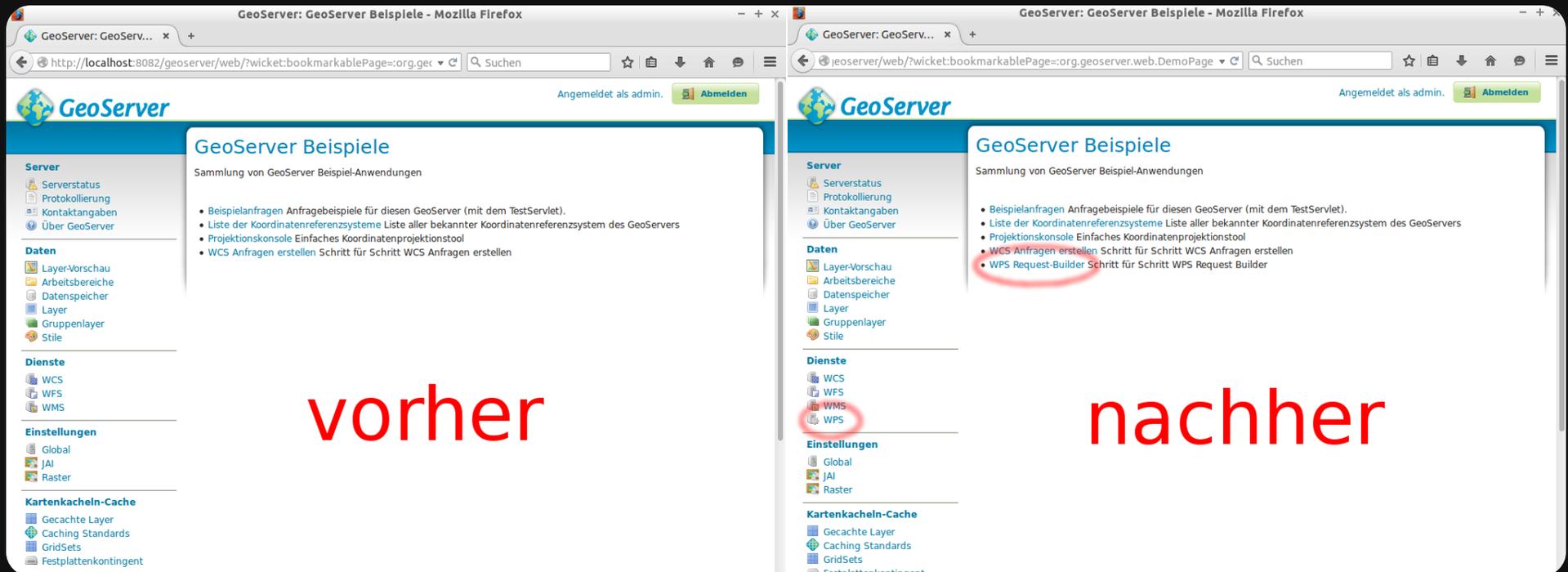
Erweiterungen



geoserver.org/release/stable

Erweiterungen installieren

1. Servlet-Container (z.B. *Tomcat*) stoppen
2. `.jar`-Artefakt(e) in das `WEB-INF/lib`-Verzeichnis des GeoServers kopieren
3. Servlet-Container starten
4. Prüfen, ob Erweiterung vorhanden



Zwischenbilanz

Wie gehe ich mit dem GeoServer-Quellcode um?

Wie kann ich den GeoServer um Funktionalität erweitern?

Lässt sich der GeoServer auch programmatisch konfigurieren?

Wie kann der GeoServer für den Produktivbetrieb optimiert werden?

REST

(REpresentational State Transfer)

- **Adressierbarkeit**
Jede Ressource hat eine eindeutige URI
- **Zustandslosigkeit**
keine Zustandsinformationen; jede REST-Anfrage enthält alle Informationen, die zum Verständnis von Client/Server nötig sind
- **Repräsentation**
Jede Ressource kann in unterschiedlichen Repräsentationen (z.B. im HTML-, JSON- und XML-Format) existieren
- **Standardmethoden**
Der Zugriff auf jede Ressource muss über standardisierte Methoden erlangt werden können

REST mit HTTP

METHODE	BESCHREIBUNG	BEISPIEL
POST	Erstelle eine Ressource (CREATE)	Layer anlegen
GET	Lese eine Ressource (READ)	Arbeitsbereiche auslesen
PUT	Aktualisiere eine Ressource (UPDATE)	Layer aktualisieren
DELETE	Entferne eine Ressource (DELETE)	Layer entfernen

REST beim GeoServer

Umsetzung per HTTP mit Basis-URL:

```
http://my/geoserver/rest
```

- Abruf in verschiedenen Formaten (Beispiel *Arbeitsbereiche*)
 - HTML (Standard):

```
http://my/geoserver/rest/workspaces.html
```

- JSON:

```
http://my/geoserver/rest/workspaces.json
```

- XML:

```
http://my/geoserver/rest/workspaces.xml
```

Dokumentation: <http://docs.geoserver.org/stable/en/user/rest/>

Layer anlegen

Request:

```
curl \
  -v \
  -u admin:geoserver \
  -XPOST \
  -H "Content-type: text/xml" \
  -d "<featuretype>
      <name>myft</name>
      <nativeName>myftsource</nativeName>
      <nativeCrs>EPSG:4326</nativeCrs>
      <enabled>true</enabled>
    </featuretype>" \
  http://my/geoserver/rest/workspaces/myws/datastores/mydata/featuretypes
```

Response:

```
HTTP/1.1 201 Created
```

Arbeitsbereiche auslesen

Request:

```
curl \  
-v \  
-u admin:geoserver \  
-XGET \  
-H "Accept: text/xml" \  
http://my/geoserver/rest/workspaces
```

Response:

```
<workspaces>  
  <workspace>  
    <name>myws</name>  
    <atom:link xmlns:atom="http://www.w3.org/2005/Atom" rel="alternate" href="http://my  
  </workspace>  
</workspaces>
```

Layer aktualisieren

Request:

```
curl \
  -v \
  -u admin:geoserver \
  -XPUT \
  -H "Content-type: text/xml" \
  -d "<featuretype>
    <enabled>true</enabled>
    <srs>EPSG:900913</srs>
    <projectionpolicy>REPROJECT_TO_DECLARED</projectionpolicy>
  </featuretype>" \
  http://my/geoserver/rest/workspaces/myws/datastores/myds/featuretypes/myft
```

Response:

```
HTTP/1.1 200 OK
```

Layer entfernen

Request:

```
curl \  
-v \  
-u admin:geoserver \  
-XDELETE \  
http://my/geoserver/rest/workspaces/myws/datastores/myds/featuretypes/myft?recurse=tr
```

Response:

```
HTTP/1.1 200 OK
```

GeoServer optimieren

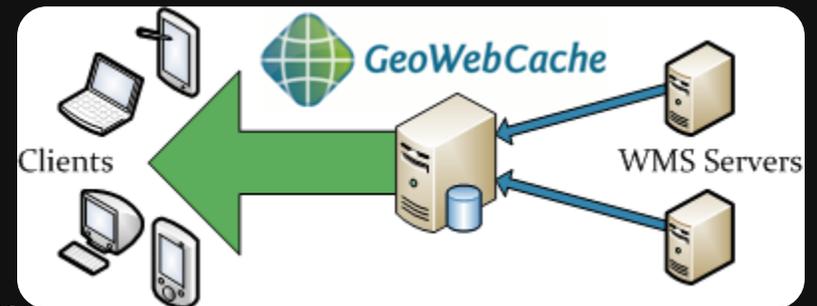
Wie kann der GeoServer für den Produktivbetrieb optimiert werden?

- Integrierter *GeoWebCache (GWC)*
- Einstellungen in der GeoServer-GUI
- *Java Virtual Machine (JVM)* tunen
- GeoServer-Datenverzeichnis auslagern

Schönes Whitepaper:

<http://boundlessgeo.com/whitepaper/geoserver-production-2/>

GeoWebCache (GWC)



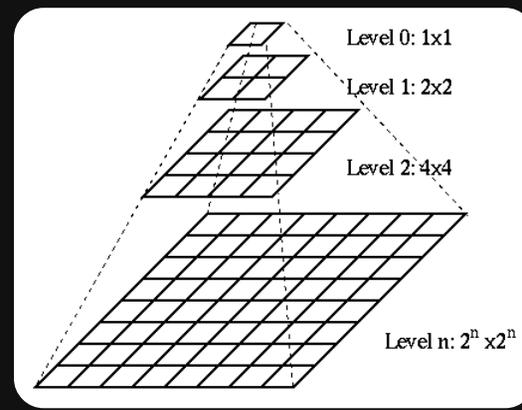
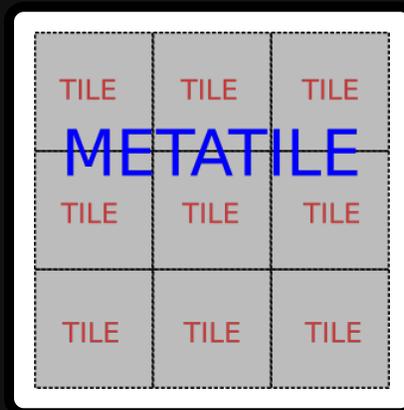
- Proxy zwischen Client und GeoServer
- Kacheln einmalig berechnen, um Prozessierungszeit sparen
- Verfügt ebenfalls über ReST-Schnittstelle

<http://docs.geoserver.org/stable/en/user/geowebcache/>

GeoWebCache (GWC)

Zwei Möglichkeiten zur Berechnung von Kartenkacheln:

- *On-The-Fly-Prozessierung*
 - Berechnung (und Ablage im Cache) nur beim ersten Aufruf
- *Vorberechnung von Kartenkacheln*
 - Die Kacheln eines Layers werden in definierten Zoomstufen und BBOX entlang eines Gridsets vorberechnet und abgelegt.



Quelle: <http://geowebcache.org>

GeoWebCache (GWC)

The screenshot displays a network inspector window with a list of requests on the left and a detailed view of the selected request on the right.

✓	Methode	Datei	Host	Kopfzeilen	Cookies	Parameter	Antwort	Zeit
●	200 GET	wms?LAYERS=topp:sta...	localhost:8082	Angefragte Adresse: http://localhost:8082/geoserver/gwc/service/wms?LAYERS...				
●	200 GET	wms?LAYERS=topp:sta...	localhost:8082	Anfragemethode: GET				
●	200 GET	wms?LAYERS=topp:sta...	localhost:8082	Status-Code: ● 200 OK			Bearbeiten und erneut senden	
●	200 GET	wms?LAYERS=topp:sta...	localhost:8082	Cache-Control: "max-age=3600, must-revalidate"				
●	200 GET	wms?LAYERS=topp:sta...	localhost:8082	Content-Disposition: "inline; filename=geoserver-dispatch.image"				
●	200 GET	wms?LAYERS=topp:sta...	localhost:8082	Content-Type: "image/png"				
●	200 GET	wms?LAYERS=topp:sta...	localhost:8082	Expires: "Tue, 03 Mar 2015 10:24:43 GMT"				
●	200 GET	wms?LAYERS=topp:sta...	localhost:8082	Last-Modified: "Tue, 03 Mar 2015 09:24:43 GMT"				
●	200 GET	wms?LAYERS=topp:sta...	localhost:8082	Server: "Jetty(6.1.8)"				
●	200 GET	wms?LAYERS=topp:sta...	localhost:8082	Transfer-Encoding: "chunked"				
●	200 GET	wms?LAYERS=topp:sta...	localhost:8082	geowebcache-cache-result: "MISS"				
●	200 GET	wms?LAYERS=topp:sta...	localhost:8082	geowebcache-crs: "EPSG:900913"				
●	200 GET	wms?LAYERS=topp:sta...	localhost:8082	geowebcache-gridset: "EPSG:900913"				
●	200 GET	wms?LAYERS=topp:sta...	localhost:8082	geowebcache-tile-bounds: "-10331840.2378125,46...7,5009377.085000001"				
●	200 GET	wms?LAYERS=topp:sta...	localhost:8082	geowebcache-tile-index: "[31, 79, 7], [31, 79, 7]"				
●	200 GET	blank.gif	localhost:8082	Host: "localhost:8082"				
				User-Agent: "Mozilla/5.0 (X11; Ubuntu; Lin...) Gecko/20100101 Firefox/34.0"				
				Accept: "image/png,image/*;q=0.8,*/*;q=0.5"				

Navigation: Alles | HTML | CSS | JS | XHR | Schriften | Grafiken | Medien | Flash | Sonstiges

Summary: 16 Anfragen, 95,66 KB, 42,53 s | Leeren

Einstellungen in der GUI

GeoServer: Vektordatenquelle bearbeiten - Mozilla Firefox

4. Tipps, Tricks & Troub... x GeoServer: Vektord... x +

http://localhost:8082/geoserver/web/?wicket:bookmarkablePage=:org.ge... Suchen

WFS
WMS

Einstellungen

- Global
- JAI
- Raster

Kartenkacheln-Cache

- Gecachte Layer
- Caching Standards
- GridSets
- Festplattenkontingent

Sicherheit

- Einstellungen
- Authentifizierung
- Kennwörter
- Benutzer, Gruppen, Rollen
- Daten
- Services

Demos

Werkzeuge

Verbindungsparameter

host *
localhost

port *
5432

database
cartaro

schema
public

user *
cartaro

passwd

Namensraum *

Expose primary keys

max connections
10

min connections
1

fetch size
1000

Connection timeout
20

validate connections

Test while idle

Evictor run periodicity

Java Virtual Machine (JVM)

`JAVA_OPTS` in Abhängigkeit der Systemumgebung setzen, z.B.

```
JAVA_OPTS="-server -Xms2g -Xmx2g -XX:MaxPermSize=256m -XX:+UseParallelGC -Dfile.encoding=...
```

PARAMETER	BESCHREIBUNG	BEISPIEL
-server	Für Server optimierte JVM	
-Xms	Anfangsgröße des Java-Heap-Speichers. Empfehlung: 2-4GB	-Xms2g
-Xmx	Maximale Größe des Java-Heap-Speichers. Evtl: <i>Xms=Xmx</i>	-Xmx2g
-XX:PermSize	Anfangsgröße des Speichers für Objektgenerierung (256m reichen)	-XX:PermSize=256m
-XX:MaxPermSize	Maximalgröße des Speichers für Objektgenerierung (256m reichen)	-XX:MaxPermSize=256m

Java Virtual Machine (JVM)

PARAMETER	BESCHREIBUNG	BEISPIEL
-Djavax.servlet.request.encoding	Encoding eingehender Anfragen (Standard: ISO 8559-1)	-Djavax.servlet.request.encoding=UTF-8
-Djavax.servlet.response.encoding	Kodierung ausgehender Antworten (Standard: ISO 8559-1)	-Djavax.servlet.response.encoding=UTF-8
-Dfile.encoding	Zeichenkodierung beim Umgang mit statischen Dateien (Standard: Default des Betriebssystems)	-Dfile.encoding=UTF-8
-XX:+UseParallelGC	Garbage Collection für Mehrkern-Systeme (siehe hier)	
-XX:+UseParallelOldGC	s.o.	

GeoServer-Datenverzeichnis

- Das Datenverzeichnis liegt standardmäßig "im" GeoServer.
- Es ist sinnvoll das Verzeichnis auszulagern, etwa für GS-Updates.
- Umgebungsvariable `GEOSERVER_DATA_DIR` muss gesetzt werden.
- Für den GWC kann auch `GEOWEBCACHE_CACHE_DIR` gesetzt werden.

The image displays two side-by-side screenshots of the GeoServer Server Status page, illustrating a configuration change. Both screenshots show the 'Serverstatus' page with the following details:

- Serverstatus:** Zusammenfassung der Konfiguration und Statusinformationen des Servers
- Datenverzeichnis:** `/usr/local/lib/geoserver-2.6.1/data_dir` (left) / `/home/user/gis_data_dir` (right)
- Sperrungen (Locks):** 0
- Verbindungen:** 5
- verwendeter Arbeitsspeicher:** 66 MB (left) / 63 MB (right)
- JVM Version:** Oracle Corporation: 1.7.0_75 (Open)DK Client VM
- Verfügbare Schriftarten:** GeoServer hat Zugriff auf 175 Schriftarten. Komplette Liste der verfügbaren Schriftarten
- Natives JAI:** true
- Natives JAI ImageIO:** true
- JAI maximaler Arbeitsspeicher:** 247 MB
- JAI verwendeter Arbeitsspeicher:** 0 KB
- JAI Grenzbereich Arbeitsspeicher:** 75.0

The left screenshot is labeled 'vorher' and the right one 'nachher'. Red circles highlight the 'Datenverzeichnis' field in both. The left sidebar menu has 'Serverstatus' circled in red. The bottom of both screenshots shows a Firefox notification: 'Firefox sendet automatisch einige Daten an Mozilla, damit die Benutzerzufriedenheit verbessert werden kann. Zu übermittelnde Daten festlegen'.

Fazit

Wie gehe ich mit dem GeoServer-Quellcode um?

Maven **und** git

Wie kann ich den GeoServer um Funktionalität erweitern?

JAR-Artefakte ins WEB-INF/lib des GS

Lässt sich der GeoServer auch programmatisch konfigurieren?

ReST-API

Wie kann der GeoServer für den Produktivbetrieb optimiert werden?

GUI, GWC, JVM...

Vielen Dank für die Aufmerksamkeit!

Fragen?

Workshop: <http://workshops.terrestris.de>

Vortrag: <http://rawgit.com/buehner/geoserver-in-action/master/index.html>

Impressum

- Autoren:
 - Nils Bühner, terrestris GmbH & Co. KG, buehner@terrestris.de
 - Daniel Koch, terrestris GmbH & Co. KG, koch@terrestris.de
- Lizenz: [CC BY-SA 3.0](https://creativecommons.org/licenses/by-sa/3.0/)
- [Vortragsfolien](#)