

# Einführung in GeoNode 4

Willkommen auf der Schulungsplattform der [CSGIS gbr](#).

Diese Dokumentation führt in die Benutzung von [GeoNode 4](#) ein.

GeoNode ist ein Content-Management-System für räumliche Daten.

Auf der linken Seite finden Sie Übungen und Informationen für

- Benutzer die eigene Ebene publizieren möchten
- Benutzer die über Administratoren-Rechte besitzen
- Benutzer die mit der Entwicklung mit GeoNode starten möchten
- System Administratoren die auf Server Ebene das Portal warten

[Download der Dokumentation als PDF](#)

Aus Gründen der besseren Lesbarkeit wird im Text verallgemeinernd das generische Maskulinum verwendet.

Diese Formulierungen umfassen gleichermaßen weibliche, männliche und diverse Personen; alle sind damit selbstverständlich gleichberechtigt angesprochen.

# 1. Einführung in GeoNode 4.0 für Anwender

---

## Inhalt:

In dieser Schulung werden wir folgende Themen mit GeoNode behandeln:

- Publikation von Geodaten, Karten und Dokumenten
- Analysieren, Filtern und Abfragen
- Karten und Daten teilen
- Einbinden und Bereitstellen eigener WMS Dienste
- Legendenerstellung
- Kartenerstellung
- Überblick: Geostories, Dashboards, Diagramme und Widgets

## Ziel:

In diesem Kurs arbeiten Sie mit GeoNode an Hand konkreter Beispiele und Fragestellungen aus der Praxis. Wir bauen gemeinsam ein Geoportal auf. Schritt für Schritt lernen wir GeoNode kennen.

Wir beschäftigen uns mit der Publikation, dem Management und der Analyse von Geodaten in GeoNode. Den Inhalt der Schulung runden wir mit weiteren Informationen über das GeoNode Projekt ab: Organisation des Open Source Projektes, Entwicklerteam, Code Repository, etc.

Nach dem Kurs besitzen Sie einen fundierten Überblick über GeoNode, dessen wichtigste Funktionen und Möglichkeiten.

# Registrieren

Im ersten Schritt erfolgt eine Registrierung auf der GeoNode Plattform. Ohne Registrierung können die Daten (Karten, Datensätze, Dokumente etc.), soweit sie öffentlich sind, angesehen aber nicht bearbeitet werden. Nachdem sich der Anwender erfolgreich registriert hat, kann er nach Freigabe des Accounts GeoNode in vollem Umfang nutzen.

## Anmeldung

Erstellen Sie ein neues lokales Konto

**E-Mail**

E-Mail-Adresse

**Benutzer**

Benutzer

**Kennwort**

Kennwort

**Passwort (wiederholen)**

Passwort (wiederholen)

Anmeldung

Die Option zum Registrieren kann nach Bedarf ausgeblendet werden, sodass ein Admin die Registrierung eigenständig durchführt. Standardmäßig läuft die Registrierung automatisch. Man kann aber diese Option ändern, sodass ein Admin die Registrierung bestätigen muss.

## User Profil

Im Profil stehen dem User u.a. folgende Funktionen zur Verfügung:

- Andere Anwender kontaktieren und Nachrichten senden
- Emailadresse der Registrierung ändern
- Passwort ändern
- Benachrichtigungseinstellungen
- Andere Anwender einladen, Ihre GeoNode Plattform zu benutzen

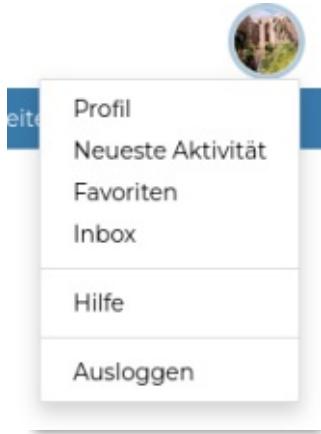
Es lohnt sich, diese Optionen in Ruhe zu prüfen und zu entscheiden, mit welchen Aktionen dem Anwender eine Email gesendet werden soll (standardmäßig sind alle Optionen ausgewählt. Mit einer GeoNode Instance, die von vielen Anwendern aktiv benutzt wird, werden Sie viele Emails erhalten!)

# Benachrichtigungseinstellungen

Notiz Typ	Email
<b>Herunterladen einer Ressource anfragen</b> Eine Anfrage für das Herunterladen einer Ressource wurde gesendet	<input checked="" type="checkbox"/>
<b>Änderung an Ressource anfragen</b> Ein Benutzer hat nach Zugang zu der Seite gefragt	<input checked="" type="checkbox"/>
<b>Dataset Created</b> A Dataset was created	<input type="checkbox"/>
<b>Dataset Updated</b> A Dataset was updated	<input type="checkbox"/>
<b>Dataset Approved</b> A Dataset was approved by a Manager	<input type="checkbox"/>
<b>Dataset Published</b> A Dataset was published	<input type="checkbox"/>
<b>Dataset Deleted</b> A Dataset was deleted	<input type="checkbox"/>

Außerdem können registrierte Benutzer:

- die letzten Aktivitäten von allen Benutzern sehen und auf die Daten zugreifen (Ebenen, Karten, Dokumente, Geostories und Dashboards)
- eine Liste aller favorisierten Inhalte sehen
- die Mailbox abfragen
- das Hilfe-Dokument (mit verlinkten Inhalten) lesen



Weitere empfehlenswerte GeoNode Dokumentationen sind:

- [GeoNode Handbuch](#)
- [Mapstore Handbuch\\*](#)

Auf der Startseite, im Menü “Über” kann man visualisieren, filtern und sehen, welche User in GeoNode registriert sind und welche Daten sie veröffentlicht haben.

## Benutzer entdecken

SUCHEN

Total: 3

Suche nach Name  

 <b>joseca</b> Keine Organisation	 <b>jose</b> Keine Organisation	 <b>joseca2</b> Keine Organisation
<input type="radio"/> 0 <input checked="" type="radio"/> 1 <input type="radio"/> 0	<input type="radio"/> 0 <input checked="" type="radio"/> 2 <input type="radio"/> 1	<input type="radio"/> 0 <input checked="" type="radio"/> 1 <input type="radio"/> 1

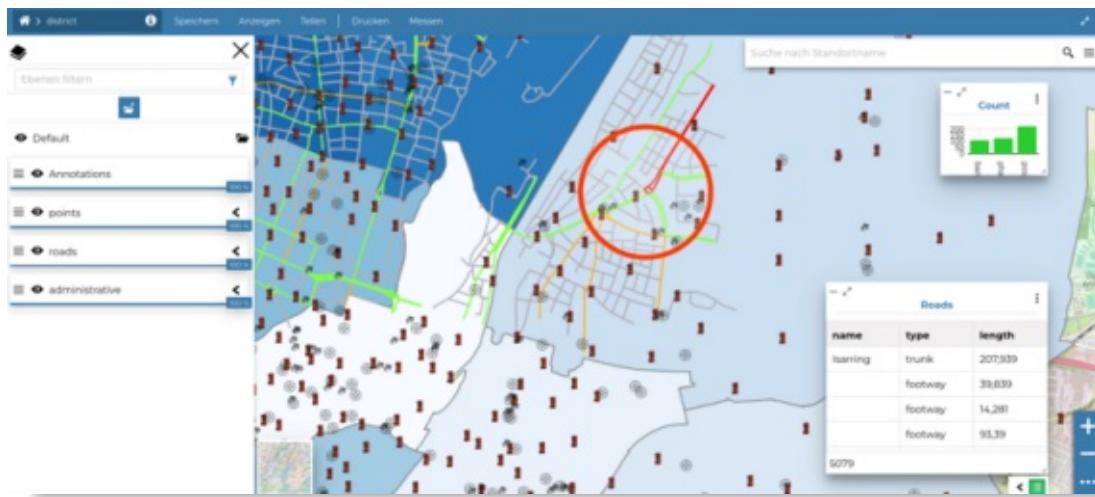
## Übung

1. Finden Sie die Email, die mit Ihrem Account verknüpft ist.
2. Erstellen Sie die gewünschten Aktionen unter Benachrichtigungen.

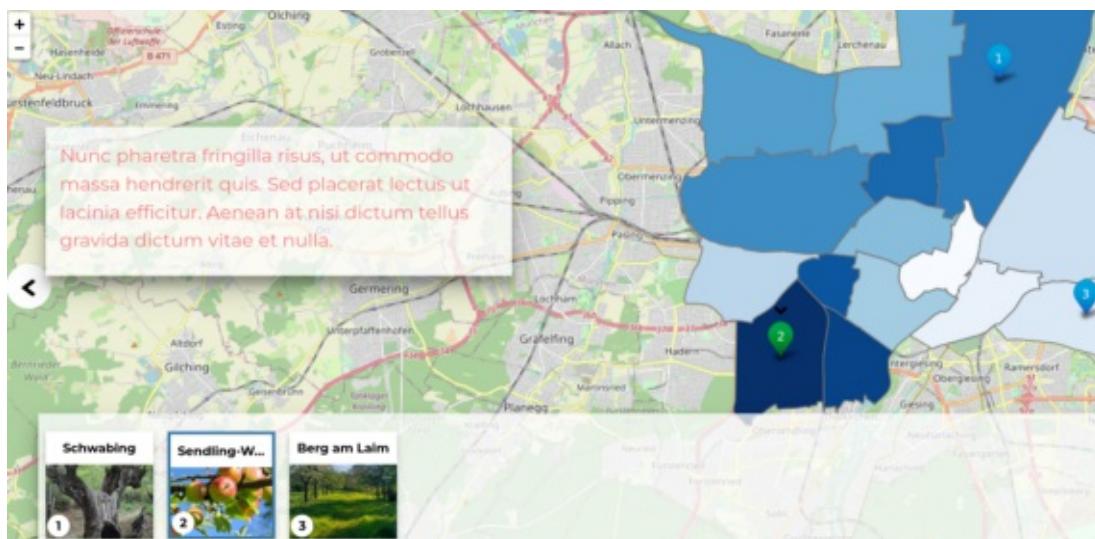
# Publikation von Ressourcen

Folgende Ressourcen können in GeoNode publiziert und bearbeitet werden:

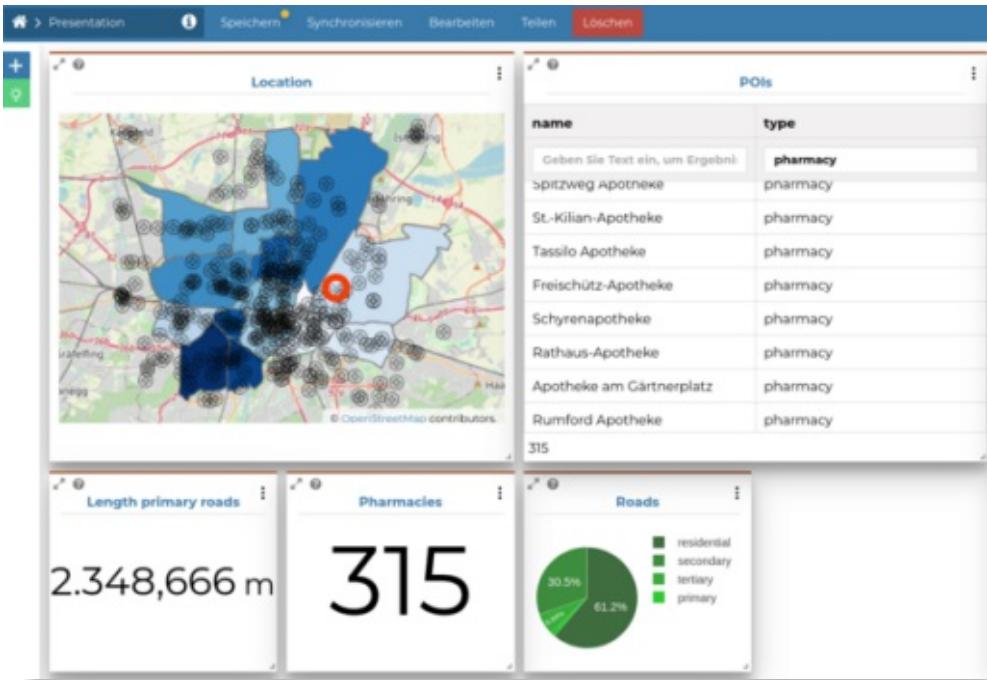
- DATENSÄTZE (Vektor, Raster, Remote oder Zeit Serien). Folgende Dateitypen können hochgeladen werden: ESRI Shapefile, GeoTIFF, Comma Separated Value (CSV), Zip Archive, XML Metadata File, Styled Layer Descriptor (SLD)
- DOKUMENTE ( Bilder, Textdateien, Videos, PDF Dokumente, Tabellen etc. ). Diese Dateitypen werden unterstützt: .txt, .log, .doc, .docx, .ods, .odt, .sld, .qml, .xls, .xlsx, .xml, .bm, .bmp, .dwg, .dx, .fif, .gif, .jpg, .jpe, .jpeg, .png, .tif, .tiff, .pbm, .odp, .ppt, .pptx, .pdf, .tar, .tgz, .rar, .gz, .7z, .zip, .aif, .aifc, .aiff, .au, .mp3, .mpga, .wav, .afl, .avi, .avs, .fli, .mp2, .mp4, .mpg, .ogg, .webm, .3gp, .flv, .vdo, .glb, .pcd, .gltf
- KARTEN (Publikation von bestehenden GeoNode Datensätzen). Der Anwender kann beliebige Datensätze hinzufügen und eigenständig eine Karte mit den verschiedenen Ebenen erstellen.



- GEOSTORIES sind Online Berichte, die mit hochgeladenen GeoNode Ressourcen oder externen Ressourcen (Videos, Webseiten etc.) erstellt werden können.



- DASHBOARDS sind Bereiche in GeoNode, in denen der Benutzer zusammenfassende Geoinformationen mit Widgets wie Diagramme, Karten, Tabellen, Texten etc. (die miteinander interaktiv verbunden sein können) präsentieren kann.



- Alle Ressourcen in GeoNode sind standardmäßig öffentlich. Der Benutzer, der die GeoNode Ressource erstellt hat, kann entscheiden, ob diese Daten für alle oder nur für registrierte Benutzer sichtbar sein sollen
- Nur die Person, der die GeoNode Ressource gehört (i.d.R. die Person, die die Ressource erzeugt hat), kann die Daten bearbeiten. Diese Person kann aber weitere Berechtigungen hinzufügen und erlauben, dass andere Benutzer die Daten ansehen, herunterladen, bearbeiten oder verwalten können

The screenshot shows the ownership and sharing settings for a GeoNode resource:

- Eigentümer:** Shows the owner as "joseca".
- Jeder:** Shows the sharing level as "Jeder".
- Registrierte Mitglieder:** Shows the sharing level as "Registrierte Mitglieder".
- + Benutzer / Gruppen hinzufügen:** Button to add users/groups.
- Nach Namen oder Berechtigungen filtern:** Search/filter bar.
- Name:** Column header for user names.
- Berechtigungen:** Column header for permissions.
- joseca2:** User entry with "Bearbeiten" (Edit) button.

- Der Besitzer der GeoNode Ressource kann entscheiden, ob einem anderen Benutzer die Daten gehören sollen (siehe optionale Metadaten):

## Übung

1. Laden Sie in GeoNode die Shape Dateien places, natural, administrative, railways und roads hoch.
2. Erstellen Sie einen Filter, damit nur Sie Ihre Daten ansehen können.

**T Filter**

Filter löschen

**x**

Suchen...

**Ressourcen**

Meine Ressourcen

# Geodaten publizieren und bearbeiten

Nach dem die Daten hochgeladen sind, stehen uns unter "Datensatz ansehen" folgende Funktionen zur Verfügung:



Speichern: Sollen Änderungen an den Daten, der Symbologie, den Berechtigungen etc. vorgenommen werden, so können Sie diese hier dauerhaft speichern. Mit "speichern als" erzeugen Sie ein neues Dataset.

Im Menü Bearbeiten können wir:

Informationen anzeigen lassen: (allgemeine Informationen zu der Ressource). Hier haben Sie auch die Möglichkeit, diese Ressource als Favorit zu speichern, den Link zu teilen oder die Datei downloaden.

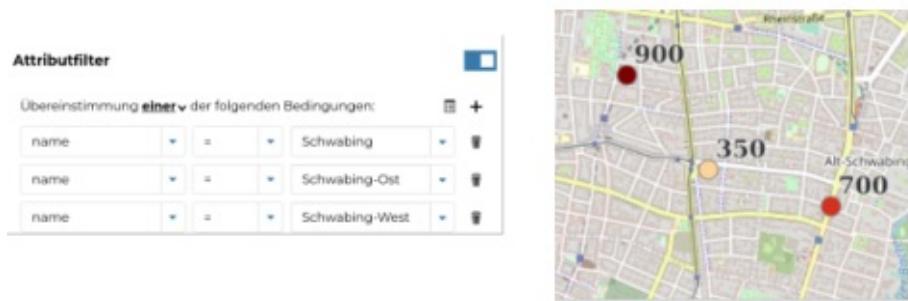
## Daten bearbeiten

Geometrien und Tabellen können in GeoNode bearbeitet werden. Die Optionen für die Digitalisierung der Geometrien sind begrenzt. Wir schauen die Editierungs-Optionen am besten mit einer Übung an (detaillierte Informationen finden Sie [hier](#)). Der erste Schritt ist, den Bearbeitungsmodus einzuschalten.

Übung:

Editierungen im Dataset places

1. Starten Sie die Editierung. Wählen Sie Schwabing aus und zoomen Sie zum Extent
2. Filtern Sie Schwabing, Schwabing-West und Schwabing-Ost, lassen Sie nur diese Orte sichtbar auf der Karte und verändern Sie die Einwohnerzahlen (Spalte population) dieser Orte.



3. Untersuchen Sie weitere Filter Optionen wie "Interessengebiet"
4. Erzeugen Sie einen Punkt und vergeben Sie die Attribute in der Tabelle
5. Verschieben Sie einen Punkt in der Karte
6. Löschen Sie einen beliebigen Punkt

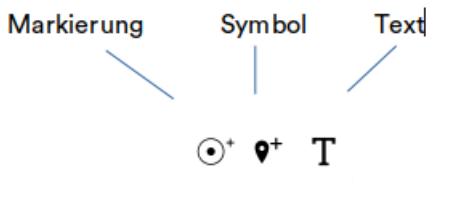
Editierungen im Dataset natural

1. Fügen Sie ein neues Polygon hinzu.
2. Digitalisieren Sie ein neues benachbartes Polygon (Snap Optionen)

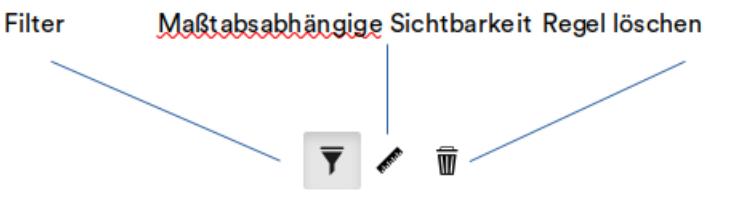
- Zum Selektieren nur ein Mal in die Karte klicken.
- Selektieren von mehreren Objekten können wir über Filter machen.

## Stil bearbeiten

Es gibt 3 verschiedene Gruppen (Regeln) von Stilen, die man benutzen kann (sie können miteinander kombiniert werden)

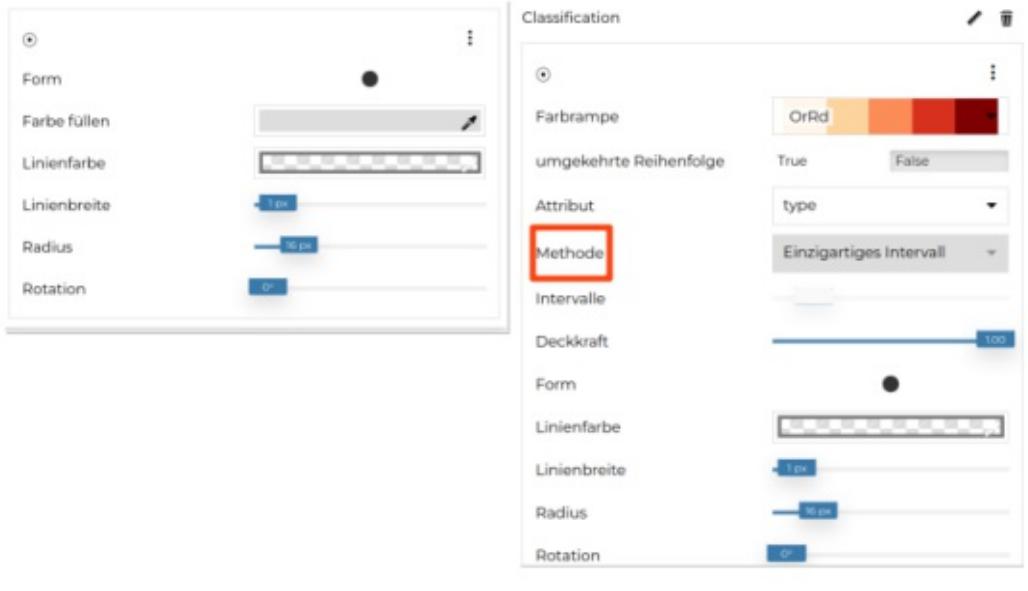


Außerdem gibt es die Möglichkeit, für die verschiedenen Regeln, die Daten zu filtern (wenn man zum Beispiel nur bestimmte Daten darstellen möchte – type=forest –) oder nur innerhalb eines bestimmten Maßstabs die Daten darstellen zu lassen.



Innerhalb der Markierungsoption\* haben wir 2 verschiedene Legenden zur Verfügung:

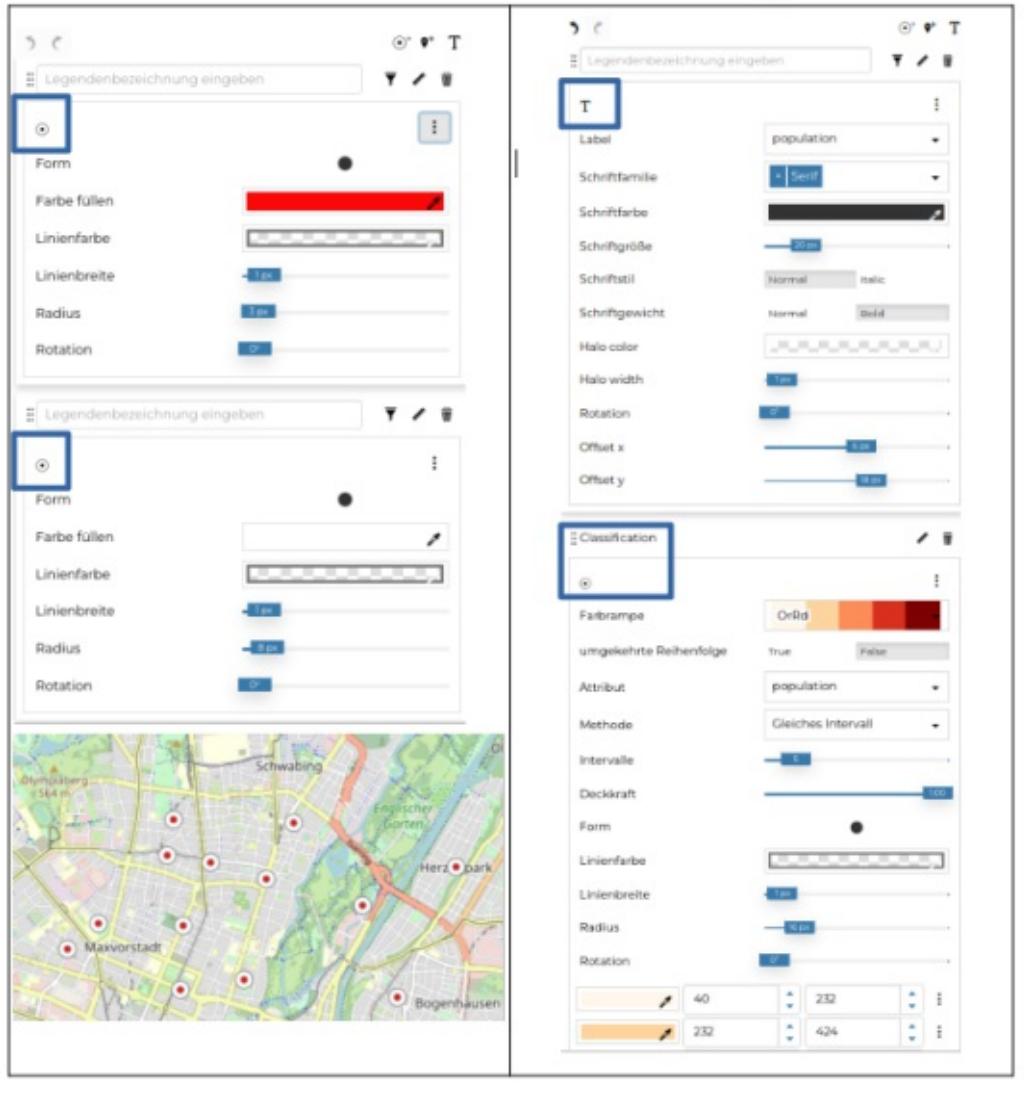
- Einfacher Stil
- Klassifizierungsstil (mit verschiedenen Methoden)



## Übung

Symbologie von places

1. Erstellen Sie diese zwei verschiedenen Legenden für die Ressource *places* und speichern Sie die Ergebnisse (mit der Änderung .sld):

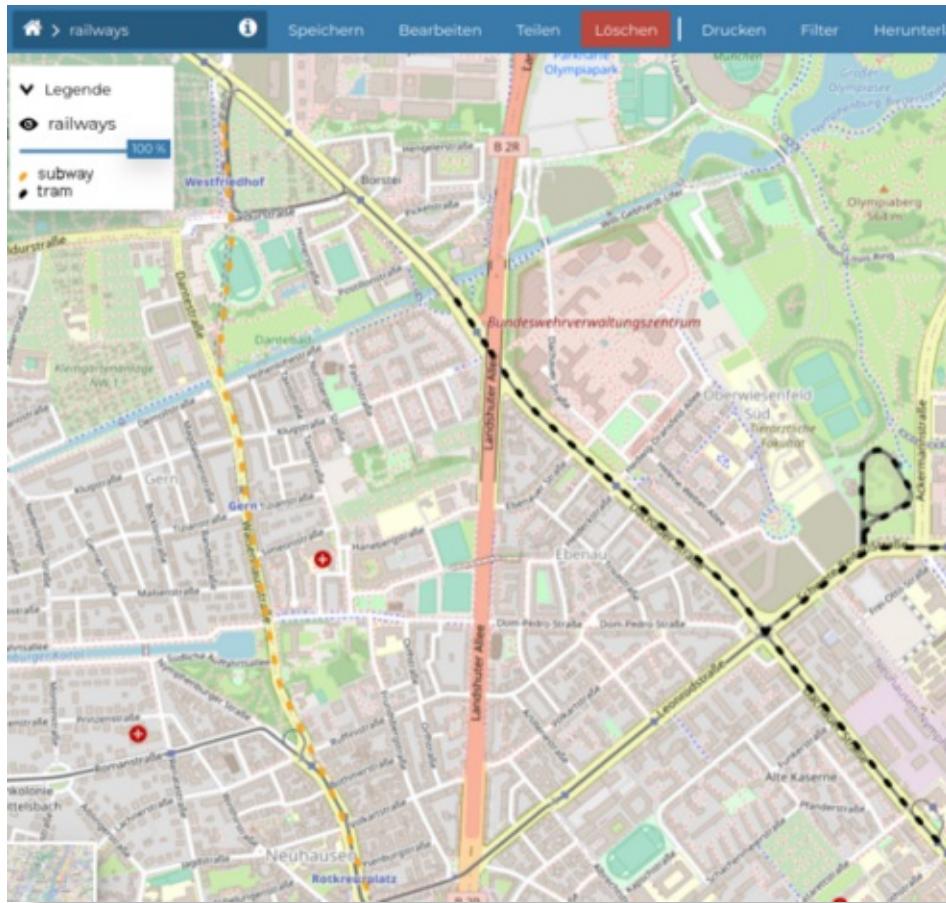


2. Lassen Sie die *places* mit einem einfachen Symbol darstellen und laden Sie die gespeicherten Stile (sld Dateien) hoch (Bearbeiten → Stil hochladen)

## Übung

### Symbologie von railways

1. Stellen Sie die railways wie in der unteren Abbildung dar. Benutzen Sie den einfachen Stil und erstellen sie die entsprechenden Filter (*type=subway* und *type =tram*).

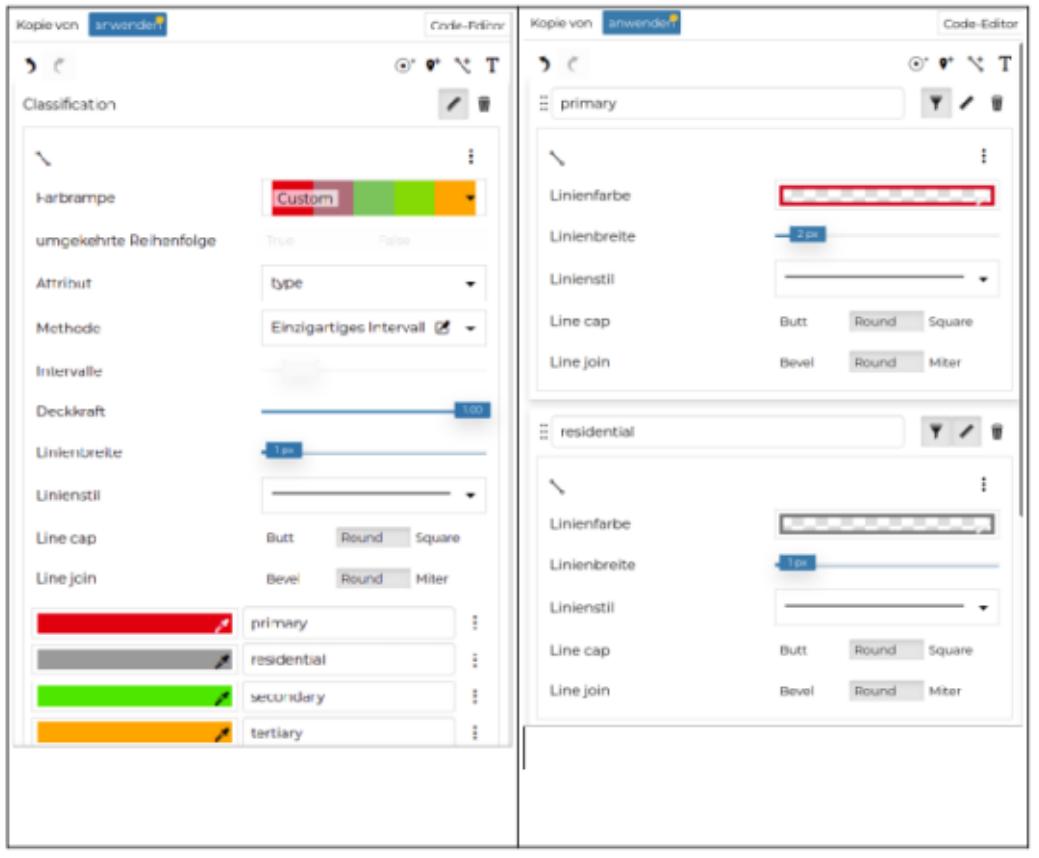


2. Speichern Sie den Stil als sld und laden Sie diese Datei in GeoNode hoch.
3. Benachrichtigen Sie Ihre Kollegen, dass es eine neue Legende gibt und prüfen Sie Ihre Mailbox bzgl. der gesendeten/empfangenen Nachrichten.

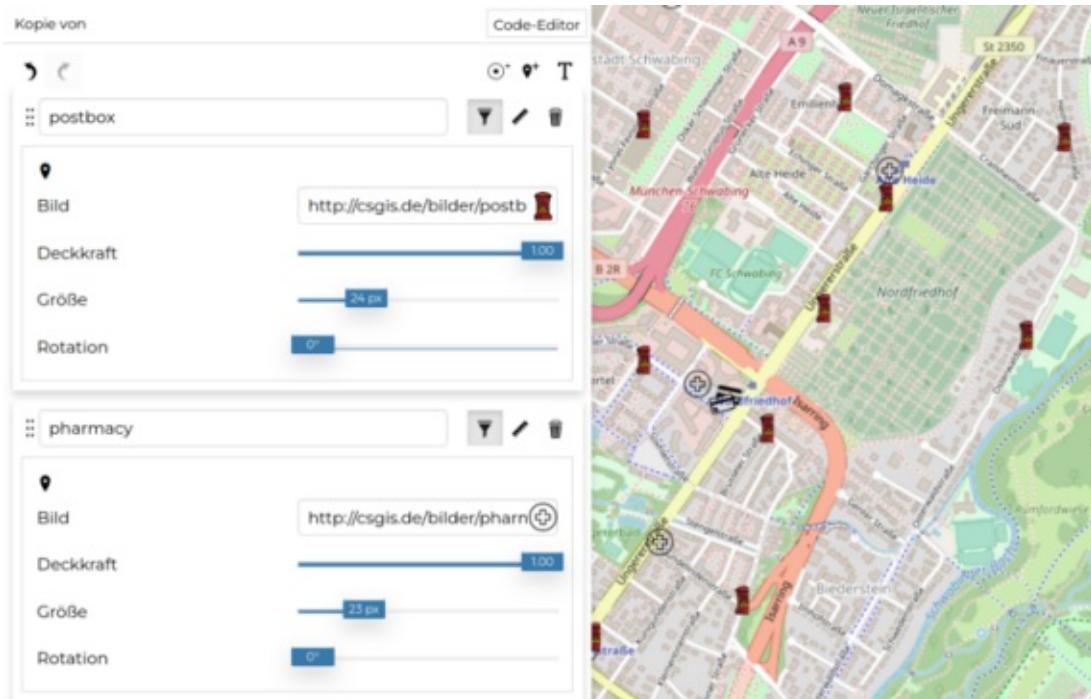
## Übung

### Symbolik von roads

1. Lassen Sie roads klassifiziert darstellen nur mit folgenden Kategorien der Spalte "Type": primary, residential, secondary, tertiary. Roads soll nur innerhalb dieser Maßstäbe erscheinen 1:144448 und 1:9028
2. Wie kann man erreichen, dass nur der Typ residential innerhalb dieser Maßstäbe erscheint aber die anderen Typen immer sichtbar bleiben?



Die Symbolregel ermöglicht die Erstellung von Legenden mit Symbolen wie Grafikdateien (svg, png, etc.). Man braucht die url, bei der die Bilder gehostet sind



## Metadaten bearbeiten

Die Metadaten in GeoNode dienen 2 wesentlichen Zwecken:

- sie geben zusätzliche Auskunft über die Daten.
- sie vereinfachen die Suche von Ressourcen durch andere Benutzer und damit können die Daten leichter

abgerufen werden.

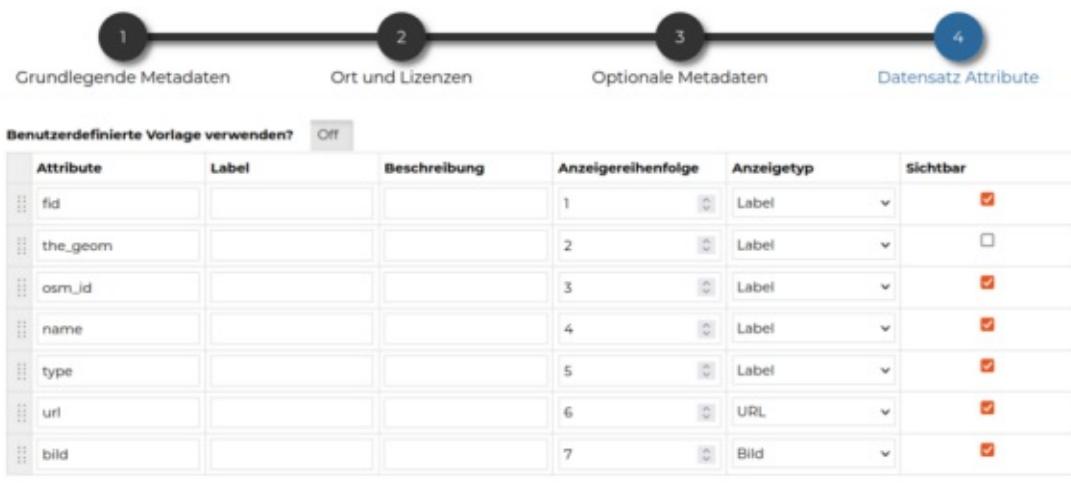
Ausführliche Informationen über die Bearbeitung von Metadaten finden Sie [hier](#)

## Übung

Metadaten von administrative

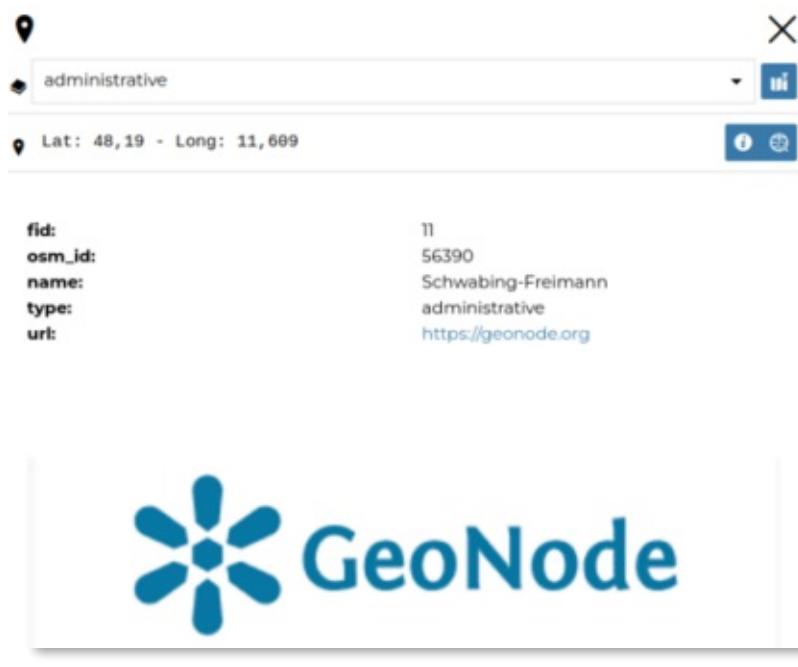
1. Thumbnail aktualisieren
2. Kategorie population speichern unter den grundlegenden Metadaten
3. Unter regions soll Germany gespeichert werden
4. Speichern Sie die Metadaten und suchen Sie auf der Startseite in GeoNode nach Daten mit z. B. der Kategorie population oder nach der Region Germany
5. Probieren Sie die Suche nach Daten mit der Filteroption "Ausmaß"

*Im Schritt 4 (Datensatz Attribute) lässt sich unter Anzeigetyp speichern, wie die gespeicherten Daten in einer Spalte angezeigt werden sollen. Mit Klick auf ein Objekt in der Karte erscheinen die Informationen entsprechend formatiert.*



The screenshot shows the GeoNode metadata editor interface. At the top, there is a progress bar with four numbered circles: 1 (Grundlegende Metadaten), 2 (Ort und Lizenzen), 3 (Optionale Metadaten), and 4 (Datensatz Attribute). Step 4 is highlighted with a blue circle. Below the progress bar, there is a section titled "Benutzerdefinierte Vorlage verwenden?" with a "Off" button. The main area is a table with columns: Attribute, Label, Beschreibung, Anzeigereihenfolge, Anzeigetyp, and Sichtbar. The table contains the following data:

Attribute	Label	Beschreibung	Anzeigereihenfolge	Anzeigetyp	Sichtbar
fid			1	Label	<input checked="" type="checkbox"/>
the_geom			2	Label	<input type="checkbox"/>
osm_id			3	Label	<input checked="" type="checkbox"/>
name			4	Label	<input checked="" type="checkbox"/>
type			5	Label	<input checked="" type="checkbox"/>
url			6	URL	<input checked="" type="checkbox"/>
bild			7	Bild	<input checked="" type="checkbox"/>



The screenshot shows the details page for a resource named "administrative". At the top, there is a location icon, the resource name, and a "More" button. Below that, it shows the coordinates "Lat: 48,19 - Long: 11,609" and two small buttons. The main content area displays the following attribute information:

fid:	11
osm_id:	56390
name:	Schwabing-Freimann
type:	administrative
url:	<a href="https://geonode.org">https://geonode.org</a>

## Ressource teilen

Beim Erstellen oder Hochladen eines neuen Datasets müssen Sie festlegen, wer diesen Datensatz anzeigen, herunterladen, bearbeiten und verwalten kann. Standardmäßig können nur Eigentümer Datensätze bearbeiten und verwalten, jeder kann sie ansehen.

Weiterführende Informationen finden Sie unter [Dataset permissions](#)

Sie können die folgenden Berechtigungen festlegen:

- Anzeigen (ermöglicht das Anzeigen des Datensatzes).
- Herunterladen (ermöglicht das Anzeigen und Herunterladen des Datensatzes).
- Bearbeiten (ermöglicht das Ändern der Metadaten, der Attribute und Geometrien und des Stils).
- Verwalten (ermöglicht das Bearbeiten, Löschen, Ändern der Freigabeoptionen und Publizieren eines Datensatzes). Datensätze, die nicht publiziert sind, können nur vom Admin gesehen und bearbeitet werden.

## Übung

Berechtigungen von places

1. Stellen Sie sicher, dass die places nicht heruntergeladen werden können. Melden Sie sich ab und prüfen Sie, dass keine Option zum Herunterladen angeboten ist.
2. Erlauben Sie, dass die places von anderen Benutzern verwaltet werden können

Name	Berechtigungen
joseca	Verwalten

## Filtern

Sie können mit dieser Option die Anzeige der Daten beeinflussen mit folgenden Möglichkeiten:

- Attribut
- Interessengebiet

Beide Optionen können auch gleichzeitig benutzt werden:

Attributfilter

Übereinstimmung **einer** der folgenden Bedingungen:

population >= 600

Interessensgebiet

Filtertyp: Polygon  
Geometrische Operation: Schneldat

## Herunterladen

Mit dieser Funktion können Sie die Datasets herunterladen in mehreren Formaten (GeoJSON, GML, Shapefile, CSV, GeoPackage oder KML).

Sie können entscheiden, ob Sie die Daten mit der ursprünglichen Projektion (prj Datei, die beim Import der Daten benutzt wurde) herunterladen werden oder in WGS84 (EPSG 4326).

Außerdem können Sie definieren, ob Sie nur die gefilterten Daten oder die Daten des aktuellen Kartenfensters herunterladen möchten.

**Daten exportieren**

Dateiformat: Shapefile

Räumliches Bezugssystem: Native

Zuschneiden des Datensatzes in das aktuelle Ansichtsfenster

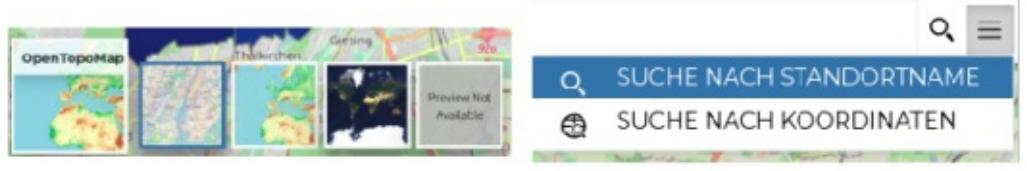
Laden Sie den gefilterten Datensatz herunter

**Export**

# Karten erstellen

Zu einer Karte können mehrere datasets hinzugefügt werden. Karten sind eigene GeoNode Ressourcen, die man publizieren oder mit anderen Benutzern bearbeiten/teilen kann.

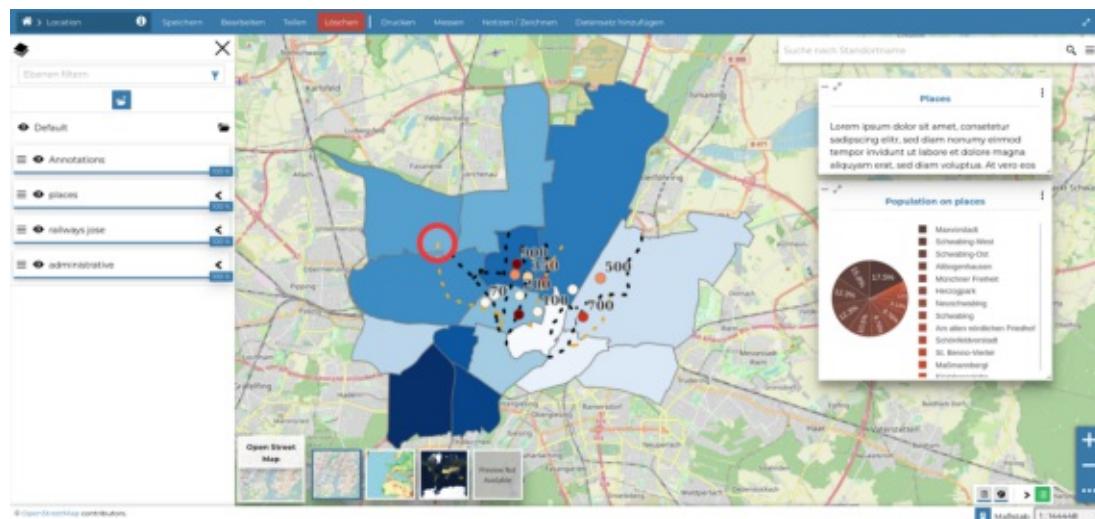
In der Karte stehen mehrere Hintergrundkarten oder eine Adresssuche zur Verfügung.



Zusätzlich können mehrere **Widgets** (z.B. Diagramme, Text, Tabellen) hinzugefügt werden, um die Karte mit anderen Informationen zu begleiten.

## Übung

Erstellung folgender Karte

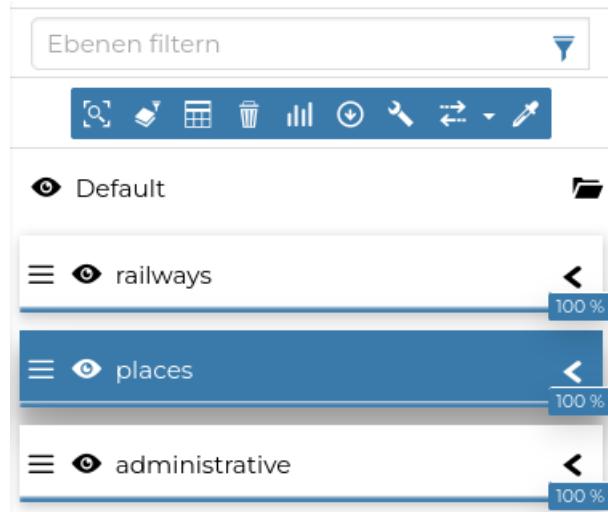


1) Ressource hinzufügen → Karte erstellen → Speichern

2) Datensatz hinzufügen → Im Datensatzkatalog die Ressource administrative, railways und places auswählen.

3) Mit Klick auf eine Ebenen erscheint eine Reihe von interessanten Funktionalitäten. Wir können hier zum Beispiel

- die Ebene filtern und nur einen Teil der Daten in der Karte anzeigen lassen
- die Attributabelle öffnen und die Daten hier bearbeiten
- Widgets für die ausgewählte Ebene erstellen
- die Symbologie der Ebene verändern



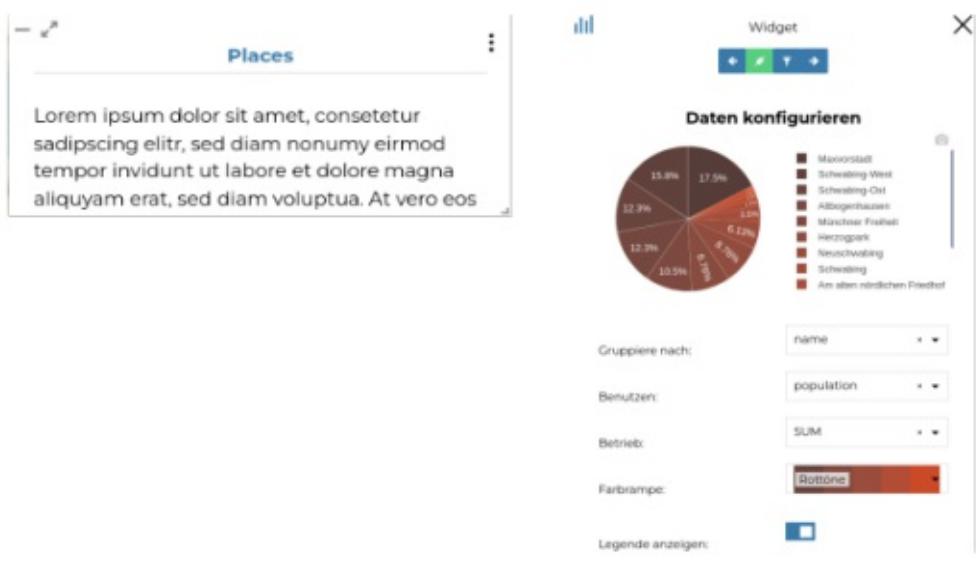
4) Verändern Sie die Symbologie von *places*: öffnen Sie die Datei *places.sld*, kopieren Sie den Inhalt und fügen Sie ihn in den Code-Editor ein.

In der Karte können wir die Symbole einer Ebene ändern, ohne die Symbologie der ursprünglichen Ressource zu verändern

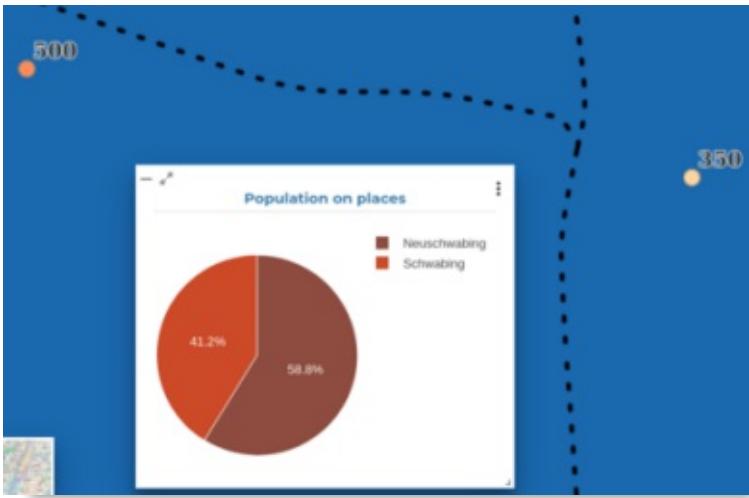
An den Stilen vorgenommene Änderungen gelten nur für den aktuellen Layer in Ihrer Karte und nicht für den ursprünglichen Datensatz.

5) Prüfen Sie, dass sich die Symbologie der GeoNode Ressource *places* nicht geändert hat.

6) Begleiten Sie die Karte mit folgenden Widgets des Layers *places*:

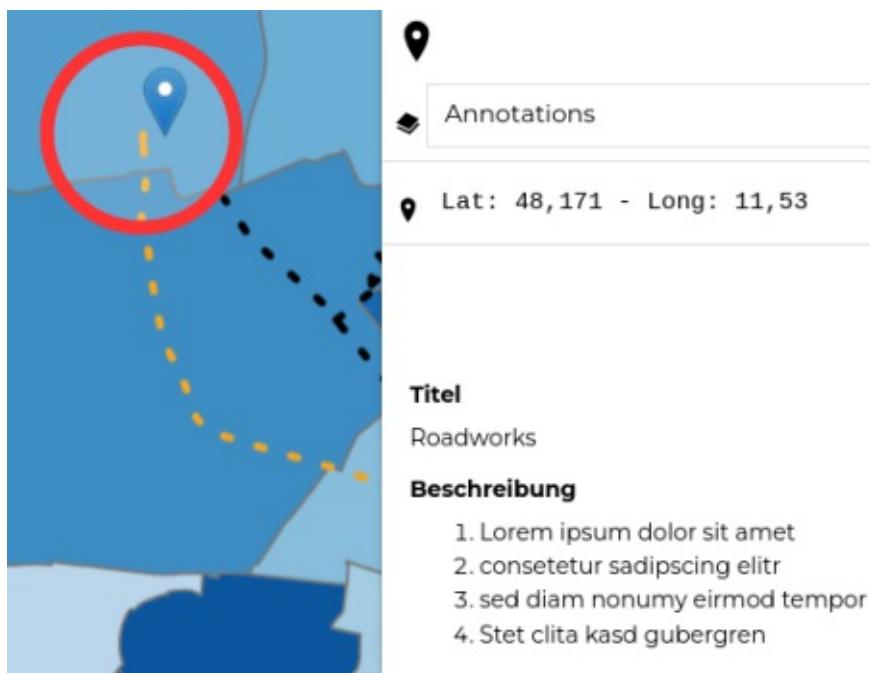


7) Beachten Sie, die Diagramme zeigen immer die Daten, die man im aktuellen Ausschnitt der Karte sieht.



8) Fügen Sie die Tabelle der Ebene administrative in die Karte ein.

9) Mit dem Tool Notizen/Zeichen fügen Sie eine Anmerkung (Annotation) hinzu. Danach lassen Sie sich mit einfaches Klick auf die Karte die Informationen anzeigen.



GeoNode gibt die Informationen von allen Ebenen aus, auf die man geklickt hat und den Hinweis auf Ebenen,

Für die folgenden Ebenen gibt es keine Objekte: **railways, places**

die keine Daten enthalten:

10) Schalten Sie alle Widgets ein und aus und lassen Sie nur den Text und das Diagramm sichtbar.

11) Lassen Sie die Ebene *administrative* transparent darstellen

administrative

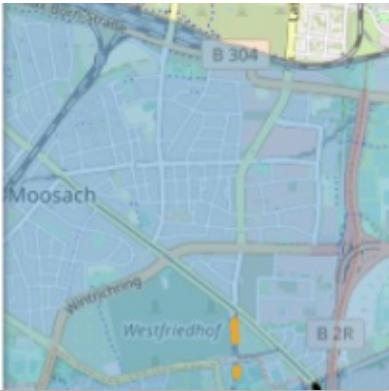
> Allgemeine Einstellungen

▼ Sichtbarkeitseinstellungen

Deckkraft %

50

Sichtbarkeitsgrenzen 



12) Wie bei allen GeoNode Ressourcen können Sie Berechtigungen an die Karte vergeben, um zum Beispiel zu erlauben, dass jemand anders diese Karte mitgestalten kann.

Mit Personen oder Gruppen teilen

Diese Seite  
<https://geonode-training.csgis.de/catalogue/#/map/18>

Diese Karte Einbetten  
<https://geonode-training.csgis.de/maps/18/embed>

Eigentümer:  
 joseca2

Jeder Ansehen

Registrierte Mitglieder

+ Benutzer / Gruppen hinzufügen

Nach Namen oder Berechtigungen filtern

Name	Berechtigungen
joseca	 Bearbeiten

13) Speichern Sie Ihre Karte und lassen Sie eine PDF im A4 Format ausdrucken.

Drucken

**Titel**

**Auflösung:**  
 96 dpi

**Beschreibung**

**Format**  
 PDF

Maßstab 1:273.438  in Druck einschließen

Raster mit Etiketten hinzufügen

**Layout**

**Blattgröße:**  
 A4

Legende einschließen

Legende auf einer separaten Seite

Querformat  Hochformat

**Legenden Optionen**

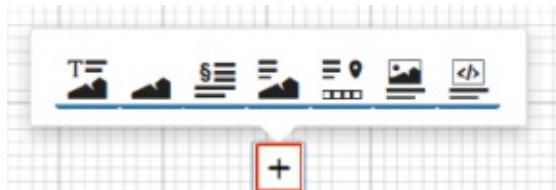
**Drucken**

# Geostories

Eine Geostory ist eine GeoNode Ressource, mit der man einen Artikel mit interaktiven Inhalten (Videos, Text, Bilder, Webseiten, dynamische Karten, andere GeoNode Ressourcen etc.) veröffentlichen kann.

[testlink](#)

Eine Geostory kann mit folgenden Elementen aufgebaut werden:



- Titelabschnitt
- Bannerbereich
- Absatzabschnitt
- Immersive Section
- Geokarussel
- Medienabschnitt
- Webseitenabschnitt

- Ein Beispiel mit allen Elementen einer Geostory finden Sie [hier](#)
- Im [Handbuch von Mapstore](#) sind alle Funktionen detailliert beschrieben.

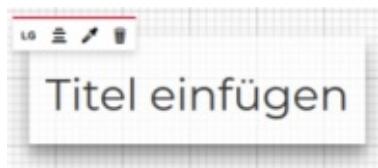
Mit der nächsten Übung bauen wir eine Geostory auf.

## Übung

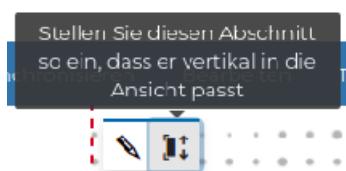
Erstellung einer Geostory

1. Ressource hinzufügen → Geostory erstellen → Speichern
2. Im ersten Schritt wird uns angeboten, dass wir mit einem Titelabschnitt anfangen. Hier finden wir 2 widgets:

Mit einfachem Klick bearbeiten und formatieren Sie den Titel.



Die Größe des Titelabschnittes können Sie anpassen oder einen Hintergrund hinzufügen mit Bildern, Videos oder Karten.



Der erste Teil unserer Geostory wäre fertig. Wir haben jetzt den Titelabschnitt mit 2 Elementen: Titel und Hintergrund (einer GeoNode Karte).



3. Im zweiten Schritt erweitern wir unsere Geostory mit einem Absatzabschnitt.

A screenshot of the "Absatz Abschnitt" (Text Paragraph) editor. It shows a header "Absatz Abschnitt" with a "type: paragraph" entry. Below it is a "Text" section with a "type: text" entry. There are search icons next to each entry. The entire editor has a light gray background.

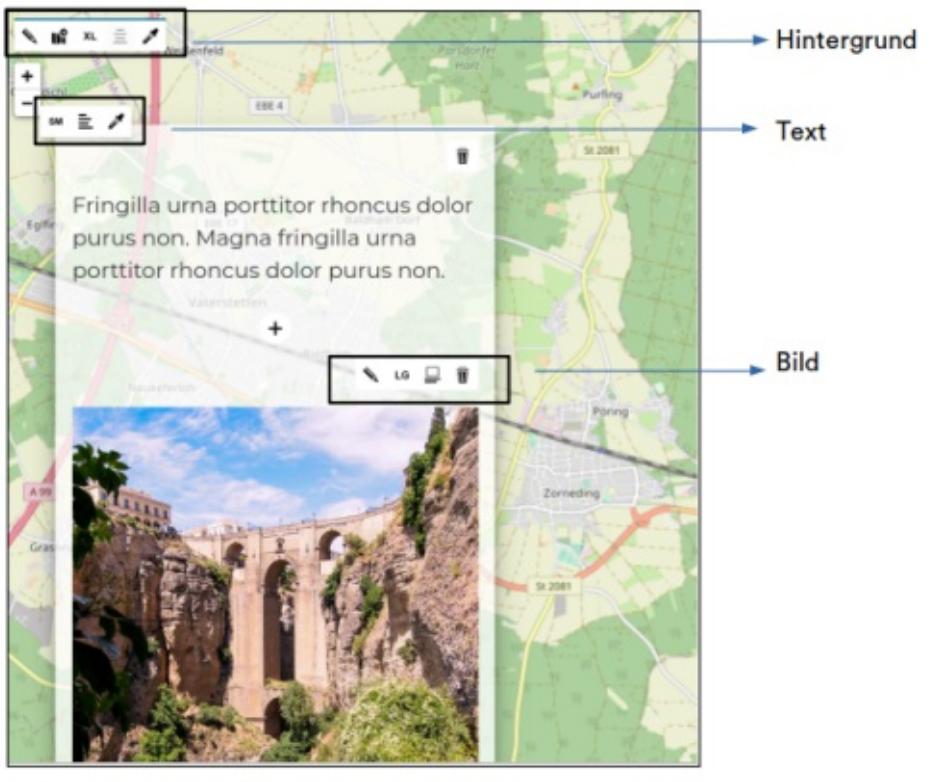
Hier können wir hinzufügen:



- Text
- Medien (Bilder, Videos, Karten)
- eine Webseite

4. Lassen Sie uns die Geostory mit einer immersive section erweitern. Sie besteht aus zwei Elementen: dem Hintergrund und dem Inhalt.

Der Unterschied zu der vorherigen Option (Absatzabschnitt) ist, dass die Inhalte im gleichen Abschnitt zusammen bleiben.



5. Im nächsten Schritt beschäftigen wir uns mit der Geokarussel Option:



- Der Hintergrund dient dem gesamten Abschnitt.
- Der Text und das untere kleine Bild werden mit einem bestimmten Ort in der Karte verknüpft.
- Sie können damit beliebige Anmerkungen zu bestimmten Orten hinzufügen.

# Dashboards

Sie eignen sich gut für die Präsentation von Ergebnissen oder Zusammenfassungen von Daten (siehe [zum Beispiel](#)).

## Übung

### Erstellung eines Dashboards

- Um ein Dashboard zu machen, müssen Sie wie gewohnt die entsprechende GeoNode Ressource erstellen: Ressource hinzufügen → Dashboard erstellen → Speichern

In ein Dashboard können Sie folgende widgets integrieren:

The screenshot shows a modal window titled "Widget" with the sub-instruction "Wählen Sie den Widgettyp". It lists five widget types: "Grafik" (with a bar chart icon), "Text" (with a text document icon), "Tabelle" (with a grid icon), "Zahler" (with a number icon), and "Karte" (with a map pin icon). Below each type is a small explanatory text.



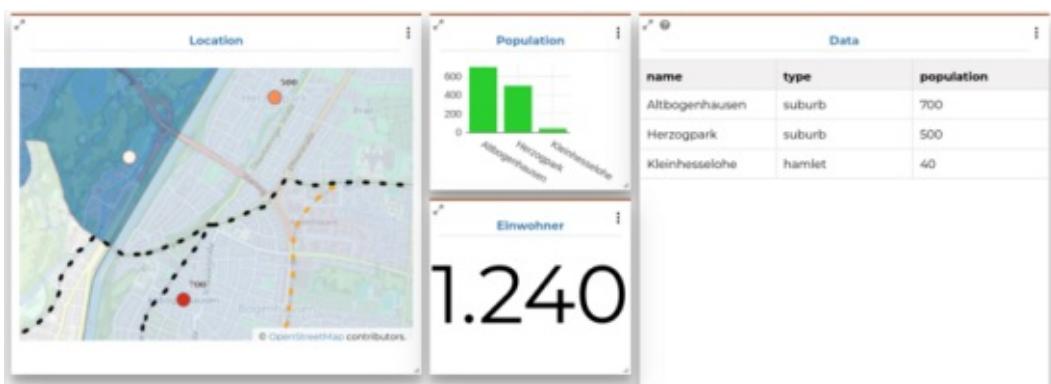
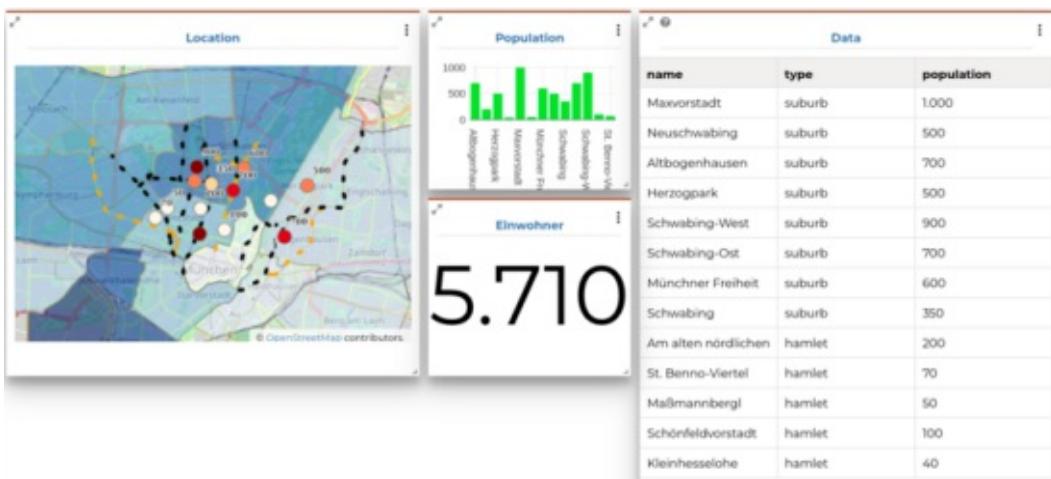
- Fügen Sie zuerst die Karte hinzu, die Sie in der Übung "Karten erstellen" gespeichert haben.
- Danach integrieren Sie in das Dashboard folgendes Diagramm mit den Einwohnerzahlen.

Im Dashboard können die Inhalte dynamisch mit anderen Inhalten verknüpft sein.

The screenshot shows the configuration interface for a dashboard. On the left, there's a "Daten konfigurieren" section with a bar chart showing population data for various locations. The chart has green bars and labels like "Gesamtpopulation über 1000", "Kleinbauten", "Hausbesitz", etc. On the right, there's a map titled "Location" showing spatial data with colored regions and numerical values (e.g., 100, 200, 300) overlaid. A red dashed line outlines the area around the map, indicating the scope of the connection. Below the map, there are dropdown menus for "X-Attribut" (set to "name"), "Y-Attribut" (set to "population"), "Betrieb" (set to "SJM"), and "Farbe" (set to "Grun").

- Fügen Sie die Tabelle der places zum Dashboard hinzu. Diese Daten sollen auch mit der Karte und dem Diagramm verknüpft werden. Mit "Verbindungen ausblenden" sehen Sie eine rote Umrandung oben über den widgets, die Sie informiert, welche Elemente des Dashboards miteinander verknüpft sind.

5. Als letztes können Sie zum Beispiel einen Zähler mit der gesamten Einwohnerzahl hinzufügen.



## 2. Geonode mit Admin Rechten

---

### Inhalt:

Dieser Schulungsbereich gibt eine Einführung für Administratoren.

Wir werden uns mit folgenden Themen befassen:

- Welche Berechtigung besitzt die Admin-Rolle
- Wie werden Gruppenrechte vergeben
- Die Benutzung des Django Admins um beispielsweise User, oder Datasets zu verwalten
- Wie funktionieren Ankündigungen

# Die GeoNode Rollen

Nach der Installation von GeoNode kennt das System folgende Rollen:

- Anonym
  - Alle nicht angemeldeten Besucher
- Registriert
  - Nutzer mit einem Benutzer-Account
- Mitarbeiter
  - Registrierte Nutzer mit erweiterten Rechten
- Administratoren
  - Registrierte Nutzer mit vollständigen Rechten

Im folgenden betrachten wir die dritte und vierte Rolle Mitarbeiter und Administrator.

## Der Administrator im Frontend

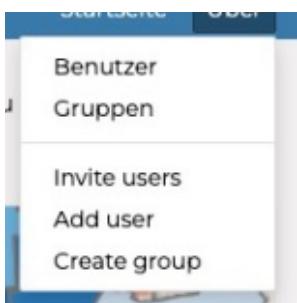
Frontend meint den für die Öffentlichkeit sichtbaren Bereich. Ihm gegenüber steht das Backend, welches das System im Hintergrund bezeichnet. Administratoren Rollen dürfen per se alles sehen, bearbeiten oder löschen.

Das erste Administratoren-Konto wird bei der Installation des Systems angelegt.

Die Definition des Users befindet sich [hier](#).

### Zusätzliche Menüpunkte

Administratoren werden im Menü zusätzliche Optionen angezeigt:



Dies sind:

- Invite User
  - Neue Benutzer einladen
- Add User
  - Neue User anlegen
- Create Group
  - Gruppen erstellen um User zu gruppieren.

### Invite User

Über die Maske Invite User besteht die Möglichkeit, eine E-Mail Einladung / Aufforderung an Dritte zu senden.

Der E-Mail Text bittet um Registrierung.

The screenshot shows a web browser window with the URL [geonode-training.csgis.de](http://geonode-training.csgis.de). The page has a blue header bar with the Geonode logo, a search bar, and navigation links for 'Daten', 'Karten', 'GeoStories', 'Dashboards', 'Beachtenswert', 'Startseite', 'Über', and 'Deutsch'. The main content area is titled 'Benutzer einladen' and contains a form with a single input field labeled 'Email' and a button labeled 'Absenden'.

## Add User

Über die Maske "Add User" besteht die Möglichkeit, neue Nutzer anzulegen.

The screenshot shows a web browser window with the URL [geonode-training.csgis.de](http://geonode-training.csgis.de). The title bar says 'Django-Verwaltung'. The main content area is titled 'Benutzer hinzufügen' and contains three input fields: 'Benutzername' (with placeholder 'Erforderlich. 150 Zeichen oder weniger. Nur Buchstaben, Ziffern und @/./-/\_.'), 'Kennwort', and 'Passwort bestätigen' (with placeholder 'Bitte das selbe Passwort zur Bestätigung erneut eingeben.'). At the bottom, there are three buttons: 'Sichern und weiter bearbeiten', 'Sichern und neu hinzufügen', and 'Speichern'.

## Create Group

Über die Maske "Create Group" können Gruppen angelegt werden.

The screenshot shows a web browser window with the URL 'geonode-training.csgis.de'. The page title is 'Erstellen Sie eine Gruppe'. There are three input fields: 'Titel' (Title), 'Logo' (Logo), and 'Beschreibung' (Description). Below the 'Logo' field is a note: 'Datei auswählen Keine Datei ausgewählt' (File select, No file selected). At the bottom of the form are links: 'geonode.org', 'Developers', and 'About'.

Gruppen sind eine Möglichkeit User zusammenzufassen. Dies ist sinnvoll um Datensätze schnell mit Rechten für einen größeren Personenkreis zu geben.

Standardmäßig dürfen nur Administratoren neue Gruppen anlegen sowie Gruppen Administratoren festlegen. Wie wir auch der Mitarbeiter-Rolle erlauben Gruppen anzulegen, sehen wir später.

Beim Anlegen einer neuen Gruppe stehen folgende Eingabefelder zur Verfügung:

- Titel der Gruppe
- Logo der Gruppe
  - Wird für die Übersicht verwendet
- Beschreibung der Gruppe
- E-Mail
  - E-Mail Adresse die verwendet wird, um ein oder alle Gruppenmitglieder, ähnlich einer Mailingliste, zu kontaktieren
- Schlüsselwörter
  - Eine durch Leerzeichen oder Kommas getrennte Stichwortliste
- Zugriff
  - Öffentlich: Jeder registrierte Nutzer kann eine öffentliche Gruppe betrachten und dieser beitreten
  - Öffentlich (nur auf Einladung): Jeder registrierte Benutzer kann die Gruppe betrachten. Nur eingeladene Benutzer können beitreten.
  - Nur eingeladene Benutzer können teilnehmen
- Kategorien
  - Weitere Gruppierung der Gruppe. Funktioniert in GeoNode 4 nur fehlerbehaftet (Bug: Kann nicht mehr gelöscht werden)

Nach dem Anlegen einer neuen Gruppe erscheint diese für alle User in der Gruppen-Übersicht:

Weiterhin können Administratoren User hinzufügen sowie Details der Gruppe ändern.

## Übungen

1. Legen Sie eine neue Gruppe an und weisen Sie der neuen Gruppe einige Mitglieder zu.

## Weiterführende Links

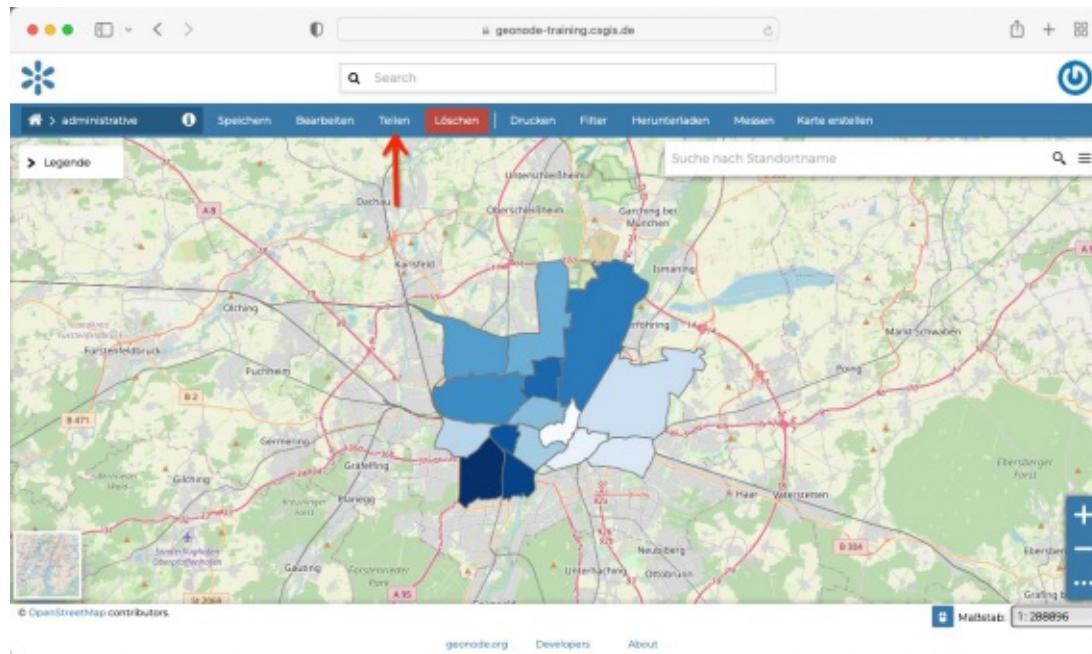
- [GeoNode Docs – Gruppen anlegen](#)

# Gruppenrechte

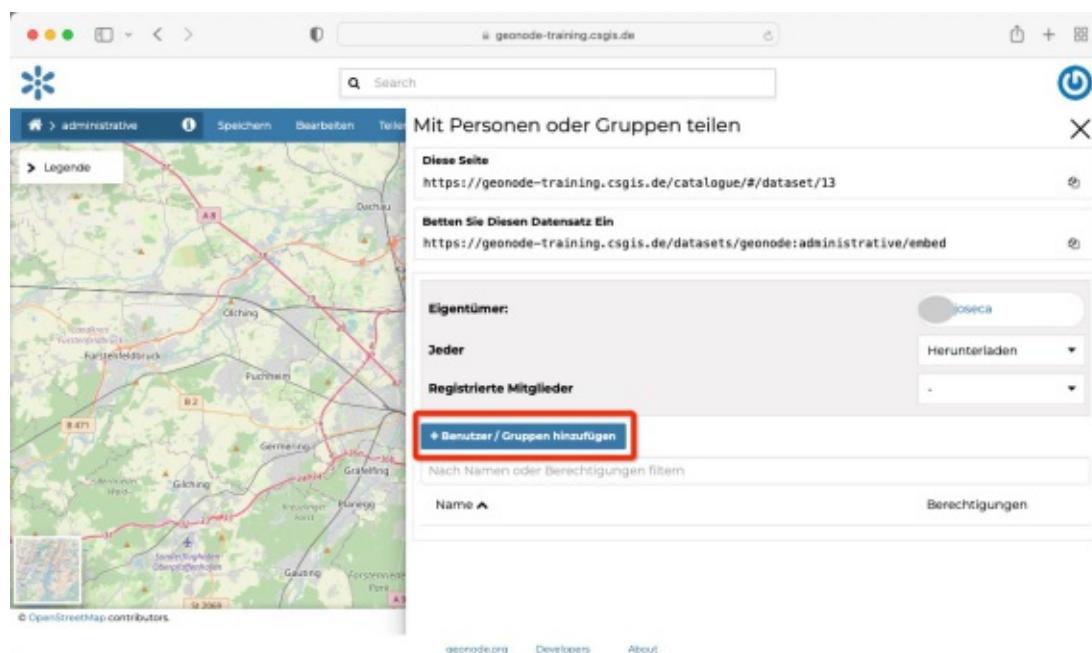
Um Ihren Datensätze-Gruppenberechtigungen zuzuweisen, müssen Sie kein Administrator sein. Jedoch, wie im vorherigen Kapitel gezeigt, um Gruppen zu erstellen.

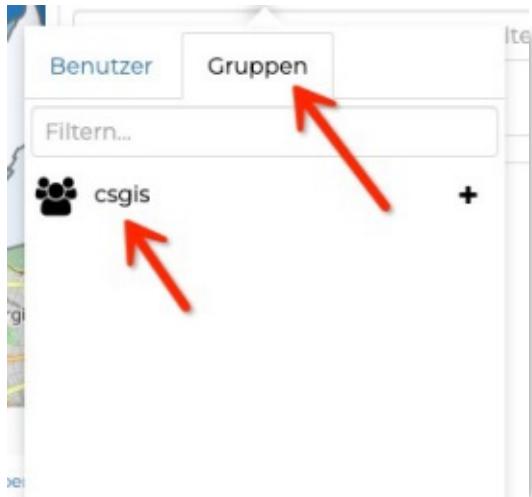
## Rechte zuweisen

Ihren Datensätzen und Karten können über das Menü teilen, Gruppen-Rechte hinzugefügt werden.



Im bekannten Panel zur Rechtevergabe wählen Sie die jeweiligen Gruppen:





Hier nach vergeben Sie die gewünschte Sichtbarkeit

A screenshot of a 'Berechtigungen' (Permissions) dialog box. At the top, there is a button '+ Benutzer / Gruppen hinzufügen' and a filter input field 'Nach Namen oder Berechtigungen filtern'. The main area shows a table with columns 'Name' and 'Berechtigungen'. A row for the 'csgis' group is selected, indicated by a red arrow. In the 'Berechtigungen' column, a dropdown menu is open, also indicated by a red arrow. The menu items are: Ansehen (selected), Herunterladen, Bearbeiten, and Verwalten.

## Übungen

1. Vergeben Sie bei Ihrem Datensatz Gruppenrechte für eine beliebige Gruppe.

## Weiterführende Links

1. [GeoNode Docs, Share Option](#)

# Django Admin

Das GeoNode System basiert auf einem Python Web-Framework namens [Django](#).

Neben Hilfestellungen in vielerlei Bereichen, stellt es eine automatisierte Administrations-Oberfläche zur Verfügung. Die Oberfläche generiert sich hierbei aus den Datenbankdefinitionen der einzelnen “Apps”.

Im Django Kontext sind Apps, in Module aufgeteilte Abschnitte einer Website wie Shop, Benutzerverwalten etc. GeoNode besitzt zahlreiche GIS spezifische “Apps”. Wie Datasets, Maps, etc.

Um auf die Administrations-Oberfläche zugreifen zu können, muss ein User Mitarbeiter, oder Administrator Status besitzen.

Die Administrations-Oberfläche erreichen Sie nach dem Login über das Menü, welches die eingestellte Sprache berücksichtigt.



Die hierauf folgende Übersicht listet alle “Django Apps”, die GeoNode besitzt.

A screenshot of the Django Admin interface. The top navigation bar shows 'Django-Verwaltung', 'GeoNode', 'admin', and 'Auf der Website anzeigen'. The left sidebar is titled 'Website-Verwaltung' and lists various models: Actstream, Actions, Fellows, Ankündigungen, Announcements, Dismissals, Authentifizierung und Autorisierung, Gruppen, Avatar, Avatars, Backup/Restore, Restored backups, Base, Configuration, Contact roles, Hierarchical keywords, Licenses, and Help. The right panel displays a list titled 'Neueste Aktionen' with entries: 'Meine Aktionen' (with three items: 'Ankündigung', 'Announcement object (3)', and 'MyMenu'), 'XXX' (with three items: 'XXX', 'MyMenu', and 'MyMenu'), 'MyMenu' (with three items: 'MyMenu', 'MyMenu', and 'MyMenu'), 'Information' (with two items: 'Information' and 'Information'), and 'MyMenu' (with one item: 'MyMenu').

# Benutzer verwalten

Betrachten wir zunächst einen sehr einfachen, jedoch wesentlichen Abschnitt des Django Admins. Die Benutzerverwaltung.

The screenshot shows the Django Admin interface for the 'Geonode' application. The left sidebar has a tree structure with various models like 'Notification checks', 'Notification metric definitions', 'Notification receivers', etc. Under the 'Benutzer' section, there is a link to 'Benutzer'. An arrow points to this link. Below it, there are links for 'Phinx Notifications', 'Notice queue batches', 'Notice settings', 'Notice types', 'Processing', 'Sequenzen', and 'Dienste'. The main content area is currently empty, showing a message '1 von 2 Treffer' (1 of 2 results) and a button 'Fertig' (Done).

## auflisten

Klicken Sie auf "Benutzer".

In der hierauf öffnenden Übersicht zeigen sich alle im System vorhandenen Benutzer. Die Tabelle zeigt hierbei ID, Name, E-Mail Adresse, Aktiv-Status des Benutzers.

The screenshot shows the 'Benutzer' list view in the Django Admin interface. The top navigation bar shows 'Geonode' and 'admin'. Below it, the breadcrumb navigation shows 'Startseite > People > Benutzer'. A button '+ Benutzer hinzufügen' is visible. The main area is a table with the following data:

ID	Benutzername	Name der Organisation	E-Mail-Adresse	Vorname	Nachname	Mitarbeiter-Status	Aktiv
-1	AnonymousUser	-	-	-	-	✗	✓
1000	admin	-	toni.schoenbuchner@csgis.de	-	-	✗	✓
1002	jose	-	jose.conalejo@csgis.de	-	-	✗	✓
1003	joseca	-	jose.conalejo@cupit.net	-	-	✗	✓
1004	joseca2	-	jose.conalejo@yahoo.es	-	-	✗	✓
1005	toni-mitarbeiter	-	toni.schoenbuchner@cupit.net	Toni	Schoenbuchner	✗	✓

At the bottom, there is a footer with a dropdown menu, a status message '0 von 6 ausgewählt' (0 of 6 selected), and a button 'Auszählen' (Select All).

Achten Sie darauf, dass alle Nutzer eine E-Mail-Adresse hinterlegt haben!

## löschen

Über die Checkbox und das Menü am Seitenfuß können Sie User löschen.

Weiterhin über den Button in der rechten oberen Ecke neue User anlegen.

The screenshot shows the Django Admin interface for the 'User' model. At the top right, there is a blue button labeled '+ Benutzer hinzufügen'. Below it, the main table lists six users. The user with ID 1002, named 'jose', is highlighted with a yellow background. A red arrow points to the ID column of this user. At the bottom of the table, there is a search bar with placeholder text 'suchen' and a dropdown menu, with a red arrow pointing to the dropdown icon. The status bar at the bottom indicates '1 von 6 ausgewählt' and has a 'Ausfiltern' button.

## editieren

Durch Klick auf die ID gelangen Sie zur Detailseite des jeweiligen Benutzers, um diesen zu editieren.

The screenshot shows the Django Admin detail page for the user 'jose'. The top navigation bar shows 'Django-Verwaltung', 'Geonode', 'admin', and 'Auf der Website anzeigen'. The main content area is titled 'Benutzer ändern'. It contains several sections: 'Persönliche Informationen' (Vorname: Jose, Nachname: Canalejo, E-Mail-Adresse: jose.canalejo@cgis.de), 'Berechtigungen' (checkboxes for 'Aktiv', 'Mitarbeiter-Status', and 'Administrator-Status'), and a footer with buttons 'Löschen', 'Sichern und weiter bearbeiten', 'Sichern und neu hinzufügen', and 'Speichern'. Red arrows highlight the 'Aktiv' checkbox in the permissions section, the 'Speichern' button at the bottom right, and the 'Löschen' button at the bottom left.

Über den Seitenfuß können Sie den User löschen oder Änderungen speichern.

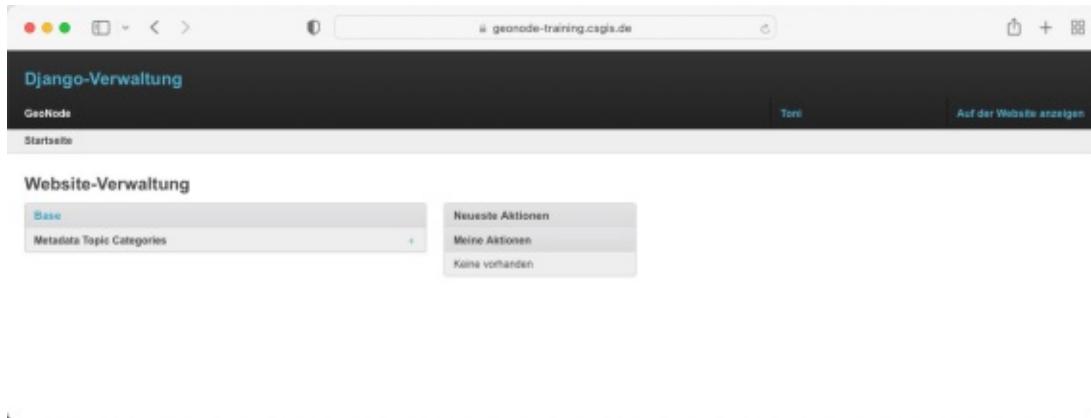
Zu beachten sind folgende Berechtigungen:

- Aktiv
  - Legt fest, ob dieser Benutzer aktiv ist
- Mitarbeiter Status
  - Legt fest, ob sich der Benutzer an der Administrationsseite anmelden kann
- Administrator Status
  - Legt fest, dass der Benutzer alle Berechtigungen hat, ohne diese einzeln zuweisen zu müssen

## Rechte zuweisen

Wie wir gesehen haben, dürfen Administratoren alle Django Admin Bereiche aufrufen und ausführen.

Benutzer mit Mitarbeiter-Status können sich zwar in die Django Oberfläche einloggen, besitzen aber so gut wie keine Möglichkeiten administrative Aufgaben zu übernehmen.



## Gruppe anlegen

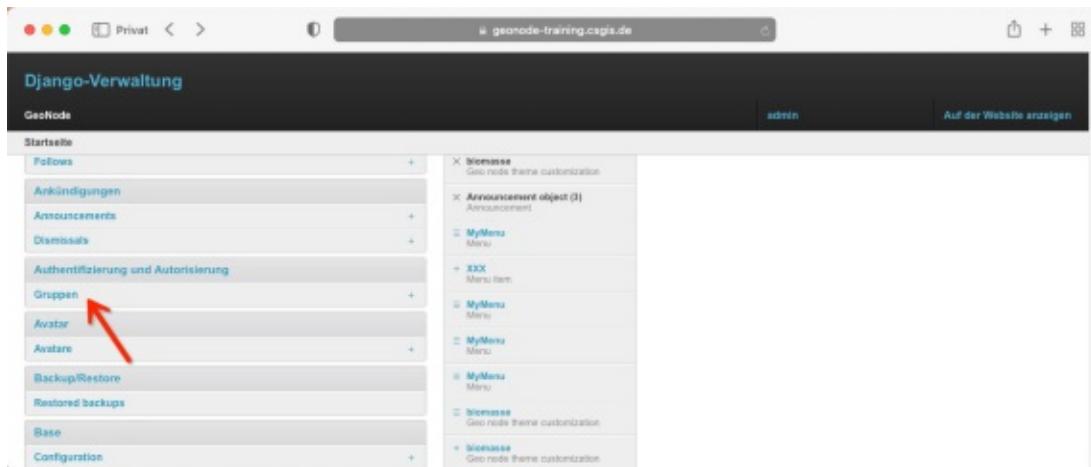
Um Mitarbeiter mit weiteren Rechten auszustatten, ist es nötig

- eine Gruppe für die Mitarbeiter zu erstellen
- der Gruppe die jeweiligen Rechte zu geben
- die Mitarbeiter der Gruppe hinzuzufügen

## Gruppe / Rechte erstellen

Als Administrator legen wir im Django Admin eine neue Gruppe mit frei wählbarem Namen an. Wir verwenden hier "Editoren".

In der Übersicht klicken wir auf Gruppen:



Und drücken in der hierauf folgenden Ansicht den Button, um eine neue Gruppe zu generieren.

Django-Verwaltung

GeNode admin Auf der Website anzeigen

Startseite > Authentifizierung und Autorisierung > Gruppen

Gruppen

S gesetzt

- Gruppe
- Gruppen Administratoren
- anonymous
- contributors
- osgis
- restviewers

8 von 5 ausgewählt Ausführen

In der folgenden Übersicht vergeben wir den Namen "Editoren", weisen aus der Liste die benötigten Rechte zu und speichern die neue Gruppe abschließend ab.

Django-Verwaltung

GeNode admin Auf der Website anzeigen

Startseite > Authentifizierung und Autorisierung > Gruppen > Hinzufügen Gruppe

Gruppe hinzufügen

Name	Edithen
	<input checked="" type="checkbox"/>

Berechtigungen

Verfügbare Berechtigungen

Filter

ausgewählte Berechtigungen

All auswählen Alle entfernen

Haben Sie die Shift-Taste (⌘ für Mac) während des Klickens gedrückt, um mehrere Einträge auszuwählen.

Sichern und weiter bearbeiten Sicher und neu hinzufügen Speichern

## Mitarbeiter hinzufügen

In der Übersicht klicken wir auf Benutzer,

Services

+

Benutzer

Benutzer

+

Pinax Notifications

und weisen dem jeweiligen Mitarbeiter unsere neue Editoren-Rolle zu.

Django-Verwaltung

Geonode

Startseite > People > Benutzer > test-mitarbeiter

Legt fest, dass der Benutzer alle Berechtigungen hat, ohne diese einzeln zuweisen zu müssen.

Gruppen:

Vergängliche Gruppen	Ausgewählte Gruppen
Gruppen Administratoren csgis	anonymous contributors registered-members Edatoren

Wichtige Termine

Letzte Anmeldung: 30.09.2022 09:41:12

**Löschen** **Sichern und weiter bearbeiten** **Sichern und neu hinzufügen** **Speichern**

Loggt sich der Mitarbeiter nun mit seinem Account ein, besitzt er über erweiterte Berechtigungen, um Datensätze zu verwalten oder im Frontend neue User-Gruppen anzulegen.

Django-Verwaltung

Geonode

Startseite

Website-Verwaltung

- Base
- Metadata Topic Categories
- Dataset**
- Datasets
- Gruppen
- Group profiles

Neueste Aktionen

Meine Aktionen

Keine vorhanden

Wir würden erwarten, dass der User im Frontend ebenfalls das Menü zum Anlegen neuer Gruppen erhält.

Startseite Über D

en

Benutzer

Gruppen

Invite users

Add user

Create group

Dem ist leider nicht so, hierbei handelt es sich um einen Bug in GeoNode4. Die neue Gruppe kann jedoch trotzdem bei direkt Aufruf des links <https://geonode-training.csgis.de/groups/create/> oder über die Django Administrationsoberfläche erzeugt werden.

# Dataset verwalten

Im letzten Kapitel haben wir uns die Administration von Benutzer und Gruppen näher angesehen. Gehen wir einen Schritt weiter und werfen einen Blick auf die *Dataset Verwaltung*.

Datasets hießen vor der GeoNode Version 3.3 Layer

The screenshot shows the Django Admin interface for GeoNode. The left sidebar has a tree structure with 'Task results', 'Dataset' (which is selected and highlighted in blue), 'Attributes', 'Datasets' (with a red arrow pointing to it), and 'Styles'. The main content area is currently empty, showing the header '1 von 2 Treffer' and 'Beginnt mit'.

## Stapelverarbeitung

Die Tabelle der bestehenden Datasets gibt uns einen kurzen Überblick über die wichtigsten Informationen.

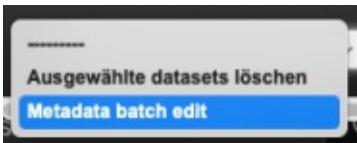
Zum Beispiel wann der Datensatz erzeugt wurde, den Titel, oder die ID.

Die Option "dataset hinzufügen" sollte nicht verwendet werden. Neue Ebenen sollen über das Frontend hinzugefügt werden.

Interessant ist jedoch die Möglichkeit, mehrere Datasets auszuwählen und in Stapelverarbeitung Rechte zuzuweisen. Hierzu kann das untere Menü der Fußleiste verwendet werden.

This screenshot shows the 'Datasets' list page in the Django Admin. A red box highlights the checkbox for the first dataset in the list. A red arrow points from the bottom of the list to the footer, which contains the text '8 von 17 ausgewählt' and the button 'Auszählen'. Another red arrow points to the top right of the page, where there is a large red button labeled '+ dataset hinzufügen'. The table lists 17 datasets, each with columns for ID, Alternative Title [en], Date, Category, and Group. The last column contains icons for edit, delete, and group assignment.

ID	Alternative Title [en]	Datum	Kategorie	Gruppe
50	geonode:unser_test2	28. September 2022 09:17	-----	v → + × -----
49	geonode:landuse_geonode	27. September 2022 19:21	-----	v → + × -----
48	geonode:landuse3	27. September 2022 12:19	-----	v → + × -----
46	geonode:unser_test	27. September 2022 10:43	-----	v → + × -----
45	ignareas_de_equus_confinatus_BH090	23. August 2022 15:57	-----	v → + × -----
42	geonode:kartierung	28. September 2022 14:15	-----	v → + × -----
41	geonode:landuse0	28. September 2022 12:31	-----	v → + × -----
40	geonode:landuse	28. September 2022 11:56	-----	v → + × -----
35	geonode:places_31488	23. September 2022 14:30	-----	v → + × -----
21	geonode:roads	19. September 2022 14:11	-----	v → + × -----
19	geonode:places	18. September 2022 14:29	Population	v → + × -----



## Detailansicht

In der Detailansicht eines Datensatzes sehen wir dass zahlreiche Felder wie die Kurzbeschreibung einfach editiert werden können.

Dies entspricht der Möglichkeit aus dem Frontend, die Metadaten des Datensatzes anzupassen.

Andere Felder ergeben sich rein aus der Tatsache, dass der Django Admin "einen Spiegel" des Datenbanksatzes darstellt. Hierdurch zeigen sich zahlreiche Felder, die von Administratoren (ohne tiefes Hintergrundwissen) nicht editiert werden sollten. Als Beispiel das CSW Feld:

```

CSW beliebiger Text
846c490f-c245-49fb-989c-77811374d6c9 eng utf8 dataset toni.schoenbuchner@csgis.de author toni.schoenbuchner@csgis.de
pointOfContact toni.schoenbuchner@csgis.de author toni.schoenbuchner@csgis.de
Geographic information - Metadata ISO 19115:2003 4326 EPSG 6.11 2022-09-28T08:17:45Z ISO 19115:2003 -
2022-09-28T08:17:39Z publication mapDigital Keine Zusammenfassung vorhanden None completed
toni.schoenbuchner@csgis.de originator None Thumbnail for 'unser_test2' imagingping ESRI Shapefile 1.0
Global place license Not Specified: The original author did not specify a license. None eng utf8
31,900091171264553 133,21420669555653 12,885190087775703 62,13697439572867 Keine Information angegeben 0
https://geonode-training.csgis.de/catalogue/#/dataset/50 WWW-LINK-1.0-http--link Online link to the 'unser_test2' description

```



# GeoNode Theming

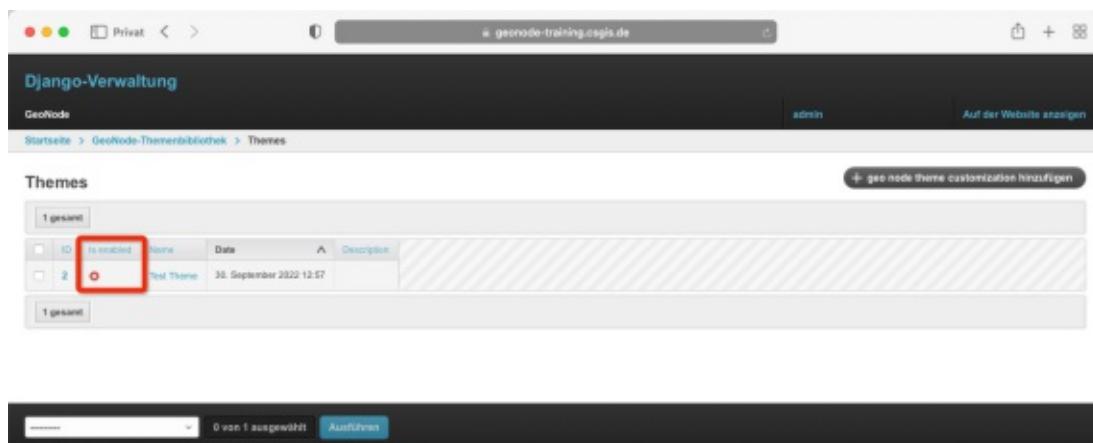
GeoNode bietet standardmäßig einige *Theming-Optionen*, die direkt über die Administrationsoberfläche verwaltet werden können. In den meisten Fällen können Sie mit diesen Optionen das Aussehen von GeoNode ganz einfach ändern, ohne eine einzige Zeile HTML oder CSS zu verändern.

Im Django Admin öffnen wir die Theme-Administration über den Link Themes:



The screenshot shows the Django Admin interface for GeoNode. The left sidebar has a section titled 'GeoNode-Themenbibliothek' with three items: 'Jumbotron theme slides', 'Themes' (which is highlighted with a red box), and 'Gruppen'. The main content area shows a table with one row, labeled '1 gesamt'. The table columns are 'ID', 'Name', 'Date', and 'Description'. The first row contains '1', 'First Theme', '30. September 2022 12:57', and an empty 'Description' field. At the top right of the table, there is a button labeled '+ geo node theme customization hinzufügen'.

Die hierauf folgende Ansicht listet alle vorhandenen Themes. Achten Sie auf den Status "is enabled". Trägt ein Datensatz "is enabled" (grüner Haken) wird das Thema verwendet.



This screenshot shows the same 'Themes' list page as the previous one, but with a red box highlighting the 'is enabled' column in the table header. The first row shows a red circle with a white 'o' indicating it is not enabled.

Über die bekannte Schaltfläche im rechten oberen Eck "geonode theme customisation hinzufügen" legen wir ein neues Thema an.

Die Oberfläche gibt uns folgende Möglichkeiten:

- Name
  - Der Name des Themas (Erscheint nirgendwo)
- Description
  - Eine Beschreibung (Erscheint nirgendwo)
- Is enabled
  - Setzt das Thema aktiv
- Logo
  - Einbinden eines eigenen Logos
- Custom CSS rules
  - Erlaubt das hinzufügen eigener CSS Regeln zum Styling
- Jumbotron Background
  - Erlaubt das hinzufügen einer Grafik im großen Banner Bereich
- Hide Text in jumbotron
  - Angehakt wird der Text im Banner der Startseite unterdrückt
- Welcome theme
  - Slideshow oder Jumbotron background

- Jumbotron Slideshow
  - Falls Slideshow gewählt ist, Slides Auswahl
- Jumbotron Title
  - Der Titel im Banner der Startseite
- Jumbotron content
  - Der im Banner angezeigte Text

Tipp: Verwenden Sie den [Theme-Generator](#), um die “Custom CSS rules” zu erzeugen und hierdurch die komplette Farbgebung zu beeinflussen.

Django-Verwaltung  
GeoNode  
Startseite > GeoNode-Themenbibliothek > Themes > dbfz

geo node theme customization ändern

Name: mein Theme  
This will not appear anywhere.

Description: Eine Beschreibung  
This will not appear anywhere.

Is enabled  
Enabling this theme will disable the current enabled theme (if any).

Logo: Derzeit: [img/2022/09/logoDBFZ\\_1650980.png](#)  
 Übernehmen  
Andere: [Datei auswählen](#) Keine Datei ausgewählt

Custom CSS rules:

Löschen    Sichern und weiter bearbeiten    Sichern und neu hinzufügen    Speichern

zeigt sich auf der Startseite

geonode-training.csgis.de

BBFZ

# Guten Tag

Dies ist ein Beispieltext

Daten Karten GeoStories Dashboards Beachtenswert Startseite Über Deutsch

Filter 28 Ressourcen gefunden Ressource hinzufügen MyMenu Sortieren nach

Humedal	unser_test2	education
Zona inundada con un régimen prácticamente permanente, regu...	-	-
admin	admin	jessica2
Ansehen	Ansehen	Ansehen

geonode.org Developers About

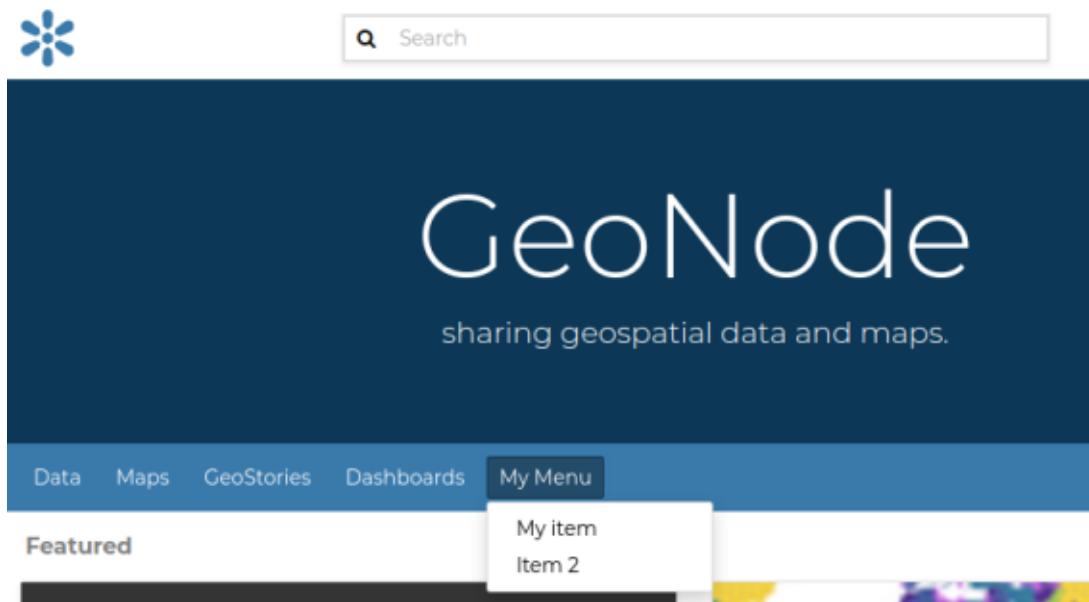
## Weiterführende Links

- [GeoNode Docs Theming](#)



# Eigene Menüpunkte erstellen

GeoNode bietet eine integrierte Funktionen, mit denen sie die Menüs einfach anpassen können.



## Neues Menü anlegen

Die relevanten Punkte finden wir im Django Admin unter “Menu Items”, “Menu placeholders”, “Menus”.

A screenshot of the Django Admin sidebar. The sidebar is organized into sections with a light gray header and a white body. The sections include: Base, Configuration, Contact roles, Hierarchical keywords, Licenses, Links, Menu items (which is highlighted with a red rectangle), Menu placeholders, Menus (which is also highlighted with a red rectangle), Metadata Regions, Metadata Restriction Code Types, Metadata Spatial Representation Types, Metadata Topic Categories, Thesauri, Thesaurus Keyword Labels, Thesaurus Keywords, and Thesaurus Labels. Each section has a plus sign (+) at the end of its name.

Wir öffnen Menus und klicken in der darauf folgenden Maske auf “menu hinzufügen”. In der darauffolgenden Maske vergeben wir einen Titel für das Menü (wird nur intern verwendet). Die Position (Feld Placeholder), sowie die Reihenfolge, falls mehrere Menus unter dem Placeholder erscheinen sollen.

The screenshot shows the Django admin interface for managing menus. A specific menu item, 'MyMenu', is being edited. The form fields include 'Title' (set to 'Unsere Forschungsdaten'), 'Placeholder' (set to 'TOPBAR\_MENU\_LEFT'), and 'Order' (set to '1'). At the bottom of the form, there are four buttons: 'Löschen' (Delete), 'Sichern und weiter bearbeiten' (Save and continue editing), 'Sichern und neu hinzufügen' (Save and add another), and 'Speichern' (Save). The 'Speichern' button is highlighted.

Und speichern das Menü ab.

Placeholder bezeichnet einen Ort im Template, an dem das Menü ausgegeben wird. Standardmäßig besitzt GeoNode 4 vordefinierte Positionen. Über das Template Tag `render_nav_menu 'CUSTOM_MENU'` könnten wir einen neuen Platzhalter an anderer Stelle definieren.

## Neuen Menüpunkt anlegen

Nachdem wir ein neues Menü definiert haben, legen wir in diesem Schritt zwei neue Menüpunkte an:

- über diese Plattform
- Alle Vectordaten

In der Django Übersicht wählen wir nun den Menüpunkt "Menu Items"



## Erster Menüpunkt

Und legen uns in der darauffolgenden Maske über den bekannten rechten Button "menu item hinzufügen" ein neues Menu Item an:

Die sich hierauf öffnende Maske bietet die Formularfelder:

- Title
  - Der Titel des Menüpunktes
- Menu
  - Das Menu indem der Button erscheinen sollen (wir haben vorhin ein Menü mit dem Namen “Unsere Forschungsdaten erstellt”)
- Order
  - Die Reihenfolgen falls mehrere Menüpunkte erscheinen
- Blank Target
  - Der Link öffnet in einem neuen Browser-Tab
- Url
  - die URL auf die der Link verweisen soll

Für unseren ersten Menüpunkt füllen wir die Felder wie folgt

Feld	Wert
Title	about
Menu	Unsere Forschungsdaten
Order	1
Blank Target	nicht gesetzt
Url	<a href="https://geonode-training.csgis.de/about/">https://geonode-training.csgis.de/about/</a>

Hiernach klicken wir auf “Sicher und neu hinzufügen”.

menu item hinzufügen

Title	about
Menu	Unsere Forschungsdaten
Order	1
Blank target	<input type="checkbox"/>
Url	https://geonode-training.csgis.de/about/

Sichern und weiter bearbeiten    Sichern und neu hinzufügen    Speichern

## Zweiter Menüpunkt

Für unseren nächsten Datensatz öffnen wir ein neues Browserfenster. Navigieren zu den Datensätzen und setzen den Haken bei Vektor und kopieren uns die URL aus der Adresszeile:

Daten Karten GeoStories Dashboards Beachtenswert about

Filter Suchen... Ressourcen

- Vorgestellt
- Datensätze**
- Vektor**
- Raster
- Remote
- Zeitserien

Dokumente Karten GeoStories Dashboards

Kategorien Wählen sie Kategorien aus

Schlüsselwörter

13 Ressourcen gefunden

unser\_test2

education

geonode.org Developers About

Zurück in der Maskedes neuen Menüpunkts füllen wir die Werte wie folgt:

Feld	Wert
Title	Unsere Vektoredaten
Menu	Unsere Forschungsdaten
Order	2
Blank Target	nicht gesetzt
Url	https://geonode-training.csgis.de/catalogue/#/search/?f=dataset&f=store-vector

Wir speichern und schließen den Menpunkt über speichern und betrachten das Ergebnis im Frontend.

## Darstellung des Menüs

Unser neues Menü zeigt sich in der oberen Navigationsleiste der Seite.

The screenshot shows the top navigation bar of the GeoNode interface. It includes a logo, a search bar, and links for 'Registrieren' and 'Anmelden'. Below the search bar is a dropdown menu with the text 'Unsere Forschungsdaten' and a submenu item 'about Unsere Vektordaten'. A red arrow points from the text 'about Unsere Vektordaten' towards the 'About' link in the main menu bar.

Ein Klick auf About bringt die User zur About Seite:

The screenshot shows the 'Über GeoNode' (About GeoNode) page. The title 'Über GeoNode' is at the top. Below it is a detailed description of what GeoNode is and how it works. There are sections for data management tools, social features, and developer support. At the bottom of the page is a footer with links to 'geonode.org', 'Developers', and 'About'.

Wie Sie diese Seite überschreiben, sehen Sie im nächsten Abschnitt "Einführung in die Entwicklung".

Ein Klick auf "Unsere Vektordaten" zeigt eine vorab gefilterte Liste an Vektor Datensätzen.

The screenshot shows the 'Unsere Forschungsdaten' (Our Research Data) page. It displays a list of vector datasets. On the left is a map showing a blue polygon labeled 'unser\_test2'. In the center is a dataset labeled 'education' with a person icon. On the right is a map showing several blue regions labeled 'administrative'. Each dataset has a 'Ansehen' (View) button below it.

Mit dieser Technik ließen sich natürlich auch Links denken die beispielsweise Datensätze speziell einer Gruppe oder einer Region auflisten.

## Weiterführende Links

- GeoNode Docs: Menus, Items und Placeholder

# Metadaten Modell

Besondere Bedeutung kommt den Metadaten jedes Datensatzes hinzu. Das Metadaten-Modell von GeoNode orientiert sich an dem Metadatenstandard ISO 19115:2003.

Sobald der Upload abgeschlossen ist, kann der Nutzer die Metadaten der Ressource füllen. Dieser werden hiernach über die CSW (OGC Catalogue Service) Endpunkte und APIs verfügbar gemacht.

Benutzer können auch ein XML-Metadaten-Dokument (ISO-, FGDC- und Dublin Core-Format) hochladen, um wichtige GeoNode-Metadatenelemente automatisch zu füllen.

## Endpunkte für den Metadatenkatalog

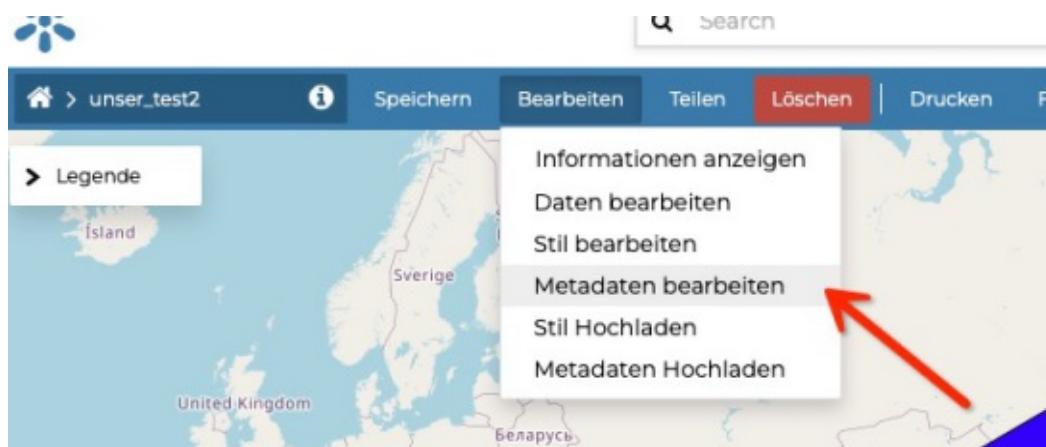
- Die Rest Schnittstelle
  - <https://geonode-training.csgis.de/api/v2/datasets?format=json>
- Der CSW Endpunkt
  - <https://geonode-training.csgis.de/catalogue/csw?service=CSW&version=2.0.2&request=GetCapabilities>

Dies erlaubt anderen Applikationen auf die durch GeoNode publizierten Daten in maschinenlesbarer Form zuzugreifen.

Als Metadaten Server kommt [pycsw](#) zum Einsatz.

Das manchmal in Dokumentation genannte “[geonetwork](#)” kann aktuell nicht als Backend genutzt werden.

Wie bereits bekannt, können User den Metadatensatz Ihrer Daten editieren. Dies erfolgt im Frontend auf Ebene Datensatz über “Metadaten bearbeiten”.



Administratoren können in beschränkten Umfang Einfluss auf die angebotenen Daten Einfluss nehmen.

Zwei einfache Beispiele:

## Lizenzen anpassen

In diesem Beispiel legen wir eine neue auswählbare Lizenz an. Im Django Admin navigieren wir über den Menüpunkt

zur Liste der Lizenzen wo wir auf “Lizenz hinzufügen” klicken

The screenshot shows the 'Licenses' list page in the GeoNode Django Admin. On the left is a table with columns 'Name' and 'Count'. The table contains several rows: 'NextView' (Count 7), 'Not Specified' (Count 1), 'Open Data Commons Open Database License / ODbL' (Count 6), 'Public Domain' (Count 4), 'Public Domain / USG' (Count 6), 'Varied / Derived' (Count 3), and 'Varied / Original' (Count 2). At the top right of the page is a button labeled '+ Lizenz hinzufügen'. A red arrow points to this button. Below the table, there is a search bar and a message indicating 8 of 7 items are selected.

Natürlich hätten wir hierüber auch die Möglichkeit bestehende zu verändern oder zu löschen.

In der folgenden Maske legen wir unsere Testlizenz an und speichern diese ab:

The screenshot shows the 'license hinzufügen' creation form. The form has the following fields:

- Name [ex]: Private License
- Abbreviation: priv1.0
- Description [er]: A custom license
- URL: http://example.com
- License text [er]: A custom license

At the bottom of the form are three buttons: 'Sichern und weiter bearbeiten', 'Sichern und neu hinzufügen', and 'Speichern'. The 'Speichern' button is highlighted.

Hiernach steht den Usern die neue Lizenz im Frontend zur Auswahl bereit:

Grundlegende Metadaten

Sprache

- NextView
- Not Specified
- Open Data Commons Open Database License / OSM
- Private License
- Public Domain
- Public Domain / USG
- Varied / Derived
- Varied / Original

Regions

Data quality

B I

\* Feld ist a gekennzeichnet

## Metadata Topic Categories

Als zweite Änderung wollen wir eine bestehende "Metadata Topic Kategorie" entfernen sowie eine neue hinzufügen.

Im Django Admin wählen wir den Link "Metadata Topic Category":

Metadata Spatial Representation Types

Metadata Topic Categories

Thesauri

In der hierauf folgenden Maske scrollen wir etwas nach unten und klicken auf "society".

<input type="checkbox"/>	population	Settlements, anthropology, archaeology, education, traditional beliefs, manners and customs, demographic data, recreational areas and activities, social impact assessments, crime and justice, census information, Economic activities, conditions and employment.	Population	female	<input checked="" type="radio"/>
<input type="checkbox"/>	society	Characteristics of society and cultures. Examples: settlements, anthropology, archaeology, education, traditional beliefs, manners and customs, demographic data, recreational areas and activities, social impact assessments, crime and justice, census information.	Society	Re-commennts	<input checked="" type="radio"/>
<input type="checkbox"/>	something	something	something	fa-females	<input checked="" type="radio"/>
<input type="checkbox"/>	structure	Man-made construction. Examples: buildings, museums, churches, factories, houses, monuments, parks, towers.	Structure	Re-building	<input checked="" type="radio"/>

Wir entfernen den Haken bei "is choice" und speichern den Datensatz ab:

Resource description type: society

Is choice

Fa class: fa-commennts

Zurück in der Liste aller Topic Kategorien wählen wir über den rechten oberen Button "Hinzufügen topic category" und füllen das Formular mit Beispiel-Daten.

The screenshot shows the Django Admin interface for a 'topic category'. The page title is 'topic category hinzufügen | Django-Systemverwaltung'. The URL is 'geonode/training/geonode/admin/base/topiccategory/add/'. The top navigation bar includes 'admin' and 'Auf der Website anzeigen'. The main form has the following fields:

- Identifier: Soil
- Description [en]: Soil data
- GeoNode description [en]: Soil data
- Is choice
- Fa class: fa-times

At the bottom are three buttons: 'Sichern und weiter bearbeiten', 'Sichern und neu hinzufügen', and 'Speichern'.

Beim Betrachten der Metadatenmaske verschwindet "Society" aus der Auswahl. Weiterhin zeigt sich unsere neue Metadaten-Kategorie "Soil data".

The screenshot shows the 'Advanced Metadata' interface. A dropdown menu is open under the 'Optional' tab, listing several categories:

- Location
- Climatology Meteorology Atmosphere
- Farming
- Population
- Soil data** (highlighted with a red border)

A note at the bottom of the dropdown says: **\* Feld ist als verpflichtend im Metadatenschema gekennzeichnet**.

# GeoNode Harvesting

Harvesting bezeichnet das maschinell unterstützte sammeln und speichern von Datensätzen anderer.

GeoNode ist in der Lage, Ressourcen-Metadaten von mehreren entfernten Diensten zu speichern.

In diesem Bereich werfen wir den Blick auf die Konfiguration, um von entfernten Diensten, eine Liste relevanter Ressourcen abzurufen und hierdurch eine Kopie zu speichern. Neben einem einmaligen "harvesten" besteht weiterhin die Möglichkeit, die Quelle in definierten Zeitabschnitten erneut zu besuchen.

Standardmäßig unterstützt GeoNode das "harvesten" von:

- Andere entfernte GeoNode-Instanzen
- OGC WMS-Servern
- ArcGIS REST-Dienste

Weiterhin besteht die Möglichkeit, eigene Harvester zu programmieren.

## Einmaliges Harvestern

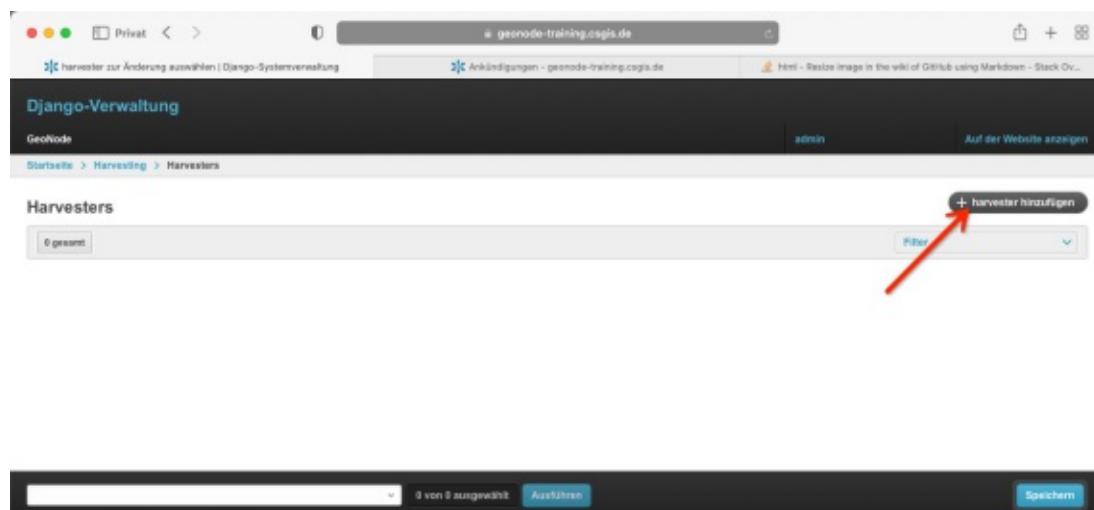
Dieser Workflow wird meist manuell vom Benutzer ausgeführt.

### Manuellen Harvester anlegen

Klicken Sie im Django Admin auf den Link zur Harvester Übersicht.



Über den Button im rechten oberen Eck legen Sie einen neuen Harvester an.



Das sich hierauf öffnende Menü zeigt folgende Konfigurationsmöglichkeiten:

- Name
  - Der Name des Harvester zur internen Bezeichnung
- remote\_url

- Basis-URL des zu erfassenden Remote-Dienstes, z. B. <https://master.demo.geonode.org>
- scheduling\_enabled
  - Gibt an, ob das Harvesting periodisch vom Harvesting-Scheduler durchgeführt werden soll oder nicht.
- harvesting\_session\_update\_frequency
  - Wie oft (in Minuten) sollen neue Harvesting-Sitzungen automatisch geplant werden?
- refresh\_harvestable\_resources\_update\_frequency
  - Wie oft (in Minuten) sollen neue Aktualisierungssitzungen automatisch eingeplant werden?
- Check availability frequency
  - Wie oft (in Minuten) soll die Verfügbarkeit des externen Diensts überprüft werden?
- default\_owner
  - Welcher GeoNode-Benutzer soll zum Eigentümer der Ressourcen ernannt werden?
- harvest\_new\_resources\_by\_default
  - Sollen neue Remote-Ressourcen automatisch geharvested werden?
- delete\_orphan\_resources\_automatically
  - Verwaiste Ressourcen sind Ressourcen, die zuvor durch eine Harvesting-Operation erstellt wurden, die GeoNode aber nicht mehr auf dem zu harvestenden Remote-Dienst finden kann. Sollen diese Ressourcen automatisch aus GeoNode gelöscht werden? Dies gilt auch, wenn eine Harvesterkonfiguration gelöscht wird. In diesem Fall werden alle Ressourcen, die von diesem Harvester stammen, als verwaist betrachtet.
- harvester\_type
  - Typ des Harvester-Workers, der für das Harvesting verwendet wird.
- Harvester type specific configuration
  - Spezifische Konfiguration für den gewählten Harvester Typ. Konfigurationsobjekt wird als JSON erwartet.

**harvester hinzufügen**

Name	Manueller Harvester Harvester name
Remote url	<a href="https://master.demo.geonode.org/W">https://master.demo.geonode.org/W</a> Base URL of the remote service that is to be harvested
Scheduling enabled	<input type="checkbox"/> Whether to periodically schedule this harvester to look for resources on the remote service
Harvesting session update frequency	90 How often (in minutes) should new harvesting sessions be automatically scheduled?
Refresh harvestable resources update frequency	90 How often (in minutes) should new refresh sessions be automatically scheduled?
Check availability frequency	10 How often (in minutes) should the remote service be checked for availability?
Default owner	admin Default owner of harvested resources
<input type="checkbox"/> Harvest new resources by default. Should new resources be harvested automatically without explicit selection?	
<input type="checkbox"/> Delete orphan resources automatically. Orphan resources are those that have previously been created by results of a harvesting operation but that GeoNode can no longer find on the remote	

[Sichern und weiter bearbeiten](#) [Sichern und neu hinzufügen](#) [Speichern](#)

Für unseren Test verwenden wir folgende Konfiguration:

Konfiguration	Wert
Name	Manueller Harvester
Remote URL	https://master.demo.geonode.org
Scheduling enabled	Nein
Harvesting session update frequency	bleibt ohne Auswirkungen da nur einmalig geharvested wird
Refresh harvestable resources update frequency	bleibt ohne Auswirkungen da nur einmalig geharvested wird
Check availability frequency	1
Check availability frequency	admin
Harvester type	geonode.harvesting.harvesters.geonodeharvester.GeonodeUnifiedHarvesterWorker
Harvester type specific configuration	{"resource_title_filter": "tl_2018_02_anrc0"}

Nach dem Speichern des Objekts wird der neue Harvester gelistet.

Wir sehen der Status zeigt “ready” und unsere Konfiguration, um nur Ebenen mit dem Titel “tl\_2018\_02\_anrc0” zu harvesten, wurde gesetzt.

Wir setzen den Haken vor dem Harvester und wählen im unteren Menü die Option “Update harvestable resources for selected harvesters”



Zurück im Django Hauptmenü rufen wir die Liste der “Harvestable resources” auf.

## Harvesting-Administration

The screenshot shows a sidebar with categories: 'Harvesting', 'Asynchronous harvesting sessions', 'Harvestable resources' (which has a red arrow pointing to it), and 'Harvesters'. Below the sidebar is a table header for 'Harvestable resources' with columns: ID, Last harvested, Last harvesting succeeded, Last harvested, Unique identifier, Harvester, Should be harvested, Previous resource type. A '+' button is at the top right of the table.

Und sehen die geharvestete Resource. Um diese lokal zu speichern setzen wir am linken Rand der Tabelle den Haken, und wählen im unteren Auswahlfeld “Harvest selected resources”.

The screenshot shows the 'Harvestable resources' list page. A row is selected (highlighted with a red box). At the bottom, there is a dropdown menu 'Harvest selected resources' with a red arrow, and a button 'Auswählen' (Select) with another red arrow. A 'Speichern' (Save) button is also visible.

Nach einiger Zeit und neu laden der Maske sollte uns ein grünes Symbol über das erfolgreiche Speichern des Datensatzes angezeigt werden.

A screenshot of a table showing harvested data. The first column contains checkboxes. The second column is 'ID' with value '823'. The third column is 'Last harvested' with value '1. Oktober 2022 09:30' and a green checkmark icon with a red arrow pointing to it. The fourth column is 'Last harvesting succeeded' with value '1. Oktober 2022 09:30'.

	ID	Last harvested	Last harvesting succeeded
<input type="checkbox"/>	823	1. Oktober 2022 09:30	1. Oktober 2022 09:30

Unser neuer Datensatz ist hiernach im GeoNode Portal sichtbar:

The screenshot shows the GeoNode portal homepage. On the left, there is a map with a yellow polygon labeled 'tl\_2018\_02\_anrc0'. To its right is a database icon and a card for 'Humedal' with the text 'Zona inundada con un régimen prácticamente permanente, regulada...'. Further right is another map with a blue polygon labeled 'unser\_test2'. The top navigation bar includes links for 'Daten', 'Karten', 'GeoStories', 'Dashboards', 'Beachtenswert', 'Registrieren', and 'Anmelden'.

## Periodisches harvesten

Das periodisches abholen und speichern entfernter Datensätze funktioniert fast identisch zum manuellen Workflow.

Um die Funktion zu illustrieren, werden wir den vorab angelegten Harvester leicht abändern.

## löschen der bestehenden Ebene

Im Django Admin navigieren wir zur [Verwaltung von Datensätzen](#) (diesmal im Django Admin). Und löschen die vorab hinzugefügte Ebene “geonode:tl\_2018\_02\_anrc0”.

The screenshot shows the 'Datasets' list view in the Django Admin. A single dataset, 'tl\_2018\_02\_anrc0' (ID 51), is selected, indicated by a red arrow pointing to the checkbox in the first column. The 'Ausgewählte datasets löschen' (Delete selected datasets) button at the bottom left is also highlighted with a red arrow. The 'Ausführen' (Execute) button at the bottom right is also highlighted with a red arrow.

Die hierauf folgende Rückfrage zum Löschen, bestätigen wir mit “Ja, ich bin sicher”.

In der [Übersicht der Harvestable resources](#), löschen wir den vorab gefundenen Datensatz:

The screenshot shows the 'Harvestable resources' list view in the Django Admin. A single resource, 'tl\_2018\_02\_anrc0' (ID 823), is selected, indicated by a red arrow pointing to the checkbox in the first column. The 'Ausgewählte harvestable resources löschen' (Delete selected harvestable resources) button at the bottom left is also highlighted with a red arrow. The 'Ausführen' (Execute) button at the bottom right is also highlighted with a red arrow.

Zurück in der [Übersicht aller Harvester](#) editieren wir den vorab angelegten Harvester “Manueller Harvester” durch Klick auf die ID.

In der Maske ändern wir folgende Angaben:

Konfiguration	Wert
Name	Automatischer Harvester
Scheduling enable	Checkbox angehakt
Harvesting session update frequency	2
Refresh harvestable resources update frequency	2
Harvest new resources by default	Checkbox angehakt
Delete orphan resources automatically	Checkbox angehakt

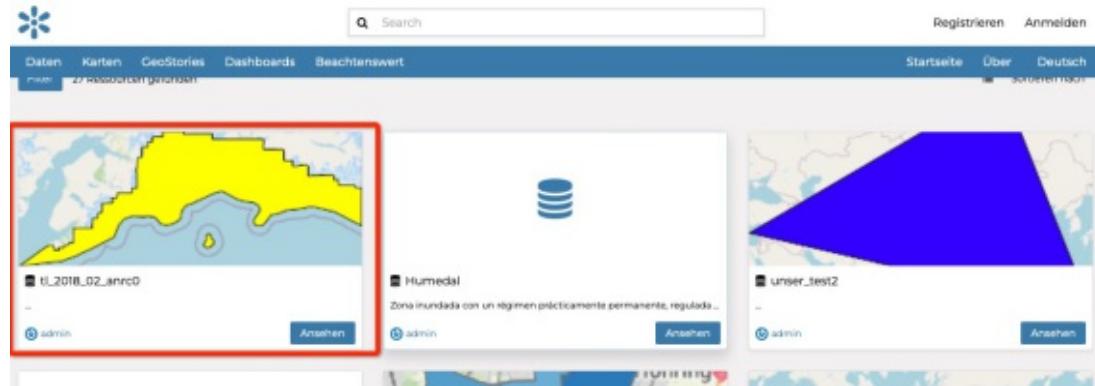
Und speichern die Konfiguration ab.

Nach einiger Zeit sollte die geharvestete Ressource in der [Übersicht der zu harvesten Ressourcen](#) auftauchen.

## Harvestable resources

1 gesamt								
ID	Last refreshed	Last harvesting succeeded	Last harvested	Unique identifier	Title	Harvester	Should be harvested	Remote resource type
824	1. Oktober 2022 10:07	1. Oktober 2022 10:05	879	E_2018_02_anrc0	Automatischer Harvester	<input checked="" type="checkbox"/>	dataset	
1 gesamt								

Sowie die Ebene im Portal sichtbar werden.



## Weiterführende Links

- [GeoNode Docs: Harvesting](#)

# Ankündigungen (announcements)

Als Administrator müssen Sie von Zeit zu Zeit Ankündigungen an User veröffentlichen.

GeoNode “announcements” ermöglichen genau das; ein Administrator hat die Möglichkeit:

- drei Arten von Nachrichten entsprechend ihrem Schweregrad zu erstellen
- ihre Gültigkeit in Bezug auf den Zeitraum zu bestimmen (Start- und Ablaufdatum der Ankündigung)
- wer sie sehen kann oder nicht (jeder oder nur die registrierten Mitglieder)
- wann Benutzer die Nachricht ausblenden können

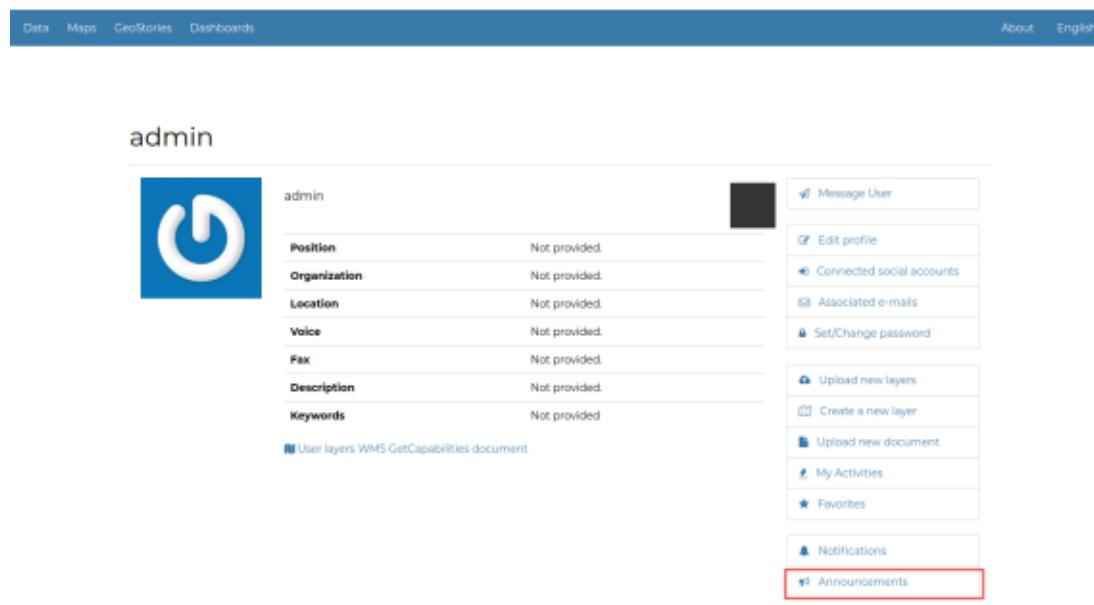
Es gibt drei Arten von Bekanntmachungen, die nach ihrem Schweregrad unterschieden werden: Allgemein, Warnung und Kritisch. Der Unterschied liegt in der Farbe der Ankündigungsbox.

Nur Administratoren und Mitarbeiter können Ankündigungen erstellen und verwalten.

Derzeit gibt es zwei Möglichkeiten, auf die Liste der Ankündigungen zuzugreifen und sie zu verwalten. Die Funktionalitäten sind für beide Oberflächen fast gleich, außer dass es über das Admin-Panel möglich ist, auch die “Ablehnungen” (dismissal) zu verwalten.

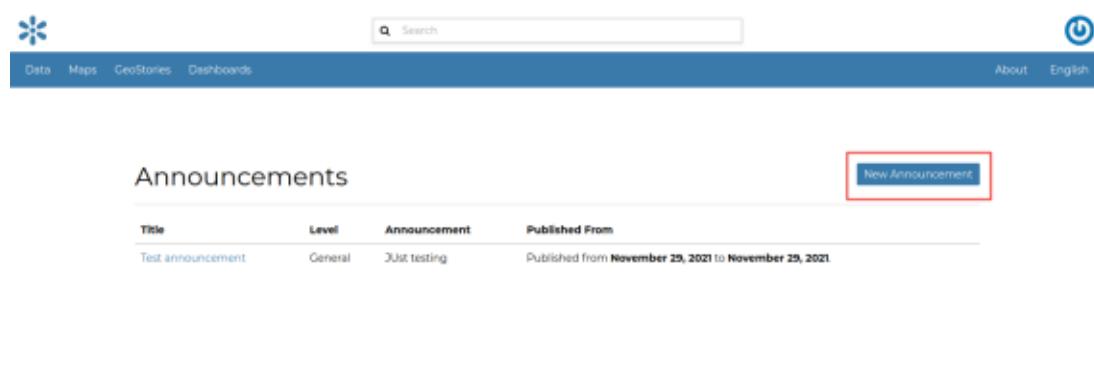
## Announcement im Frontend über Profil anlegen

Klicken Sie im Profilbereich auf den Link Ankündigungen (Announcements)



The screenshot shows the GeoNode Admin Panel. At the top, there's a navigation bar with links for Data, Maps, GeoStories, Dashboards, About, and English. Below the navigation is a user profile section for 'admin'. It includes a blue placeholder icon for a profile picture, the username 'admin', and a list of user details: Position (Not provided), Organization (Not provided), Location (Not provided), Voice (Not provided), Fax (Not provided), Description (Not provided), and Keywords (Not provided). Below this list is a link to 'User layers: WMS GetCapabilities document'. To the right of the user details is a sidebar with several links: Message User, Edit profile, Connected social accounts, Associated e-mails, Set/Change password, Upload new layers, Create a new layer, Upload new document, My Activities, Favorites, Notifications, and Announcements. The 'Announcements' link is highlighted with a red box.

Klicken Sie entweder auf “Neue Ankündigung”, um eine neue Bekanntmachung zu erstellen, oder auf den Titel einer bestehenden Ankündigung, um deren Inhalt zu verwalten.



The screenshot shows the 'Announcements' list in the GeoNode Admin Panel. At the top, there's a search bar and a navigation bar with links for Data, Maps, GeoStories, Dashboards, About, and English. Below the navigation is a header 'Announcements' with a 'New Announcement' button, which is highlighted with a red box. The main area displays a table with columns: Title, Level, Announcement, and Published From. There is one entry: 'Test announcement' (Level: General, Announcement: Just testing, Published From: November 29, 2021 to November 29, 2021).

# Ankündigung über den Django Admin

The screenshot shows the GeoNode Site administration interface. On the left, there's a sidebar with various administrative sections like Accounts, Email addresses, Actstream, Actions, Follows, Announcements, Authentication and Authorization, Groups, Avatar, Avatars, Backup/Restore, Restored backups, Base, Configuration, Contact roles, and Curated thumbnails. The 'Announcements' section is highlighted with a red box. On the right, there's a 'Recent actions' sidebar listing recent activities such as 'Announcement object (1)', 'Transport', 'Transportation planners', 't', 'joe', 'Dev', 'Slide 3', and 'Slide 2'.

Das Verwalten von Ankündigungen über das Admin-Panel ist im Grunde dasselbe; die Felder für das Formular sind identisch.

The screenshot shows the Django administration interface for adding a new announcement. The form fields include: Title (empty input field), Level (dropdown set to 'General'), Content (large text area), Site wide (checkbox unchecked), Members only (checkbox unchecked), Publish\_start (date and time set to 2023-11-29 13:54:51), Publish\_end (date and time set to 2023-11-29 13:54:51), and Dismissal type (dropdown set to 'Session Only Dismissal').

Der Zugriff auf die Ankündigungsoptionen über das Admin-Panel ermöglicht Ihnen auch die Verwaltung von "Ablehnungen". Über diese Oberfläche können Sie selektiv entscheiden, welche Mitglieder eine bestimmte Ablehnung sehen können und welche nicht, oder Sie können sie zwingen, die Nachrichten erneut zu sehen, indem Sie die Ablehnungen entsprechend löschen.

- Keine Ablehnungen erlaubt:
  - Das Mitteilungsfeld der Ankündigung kann nicht geschlossen werden.
- "Session Only Dismissal",
  - die Standardeinstellung, ermöglicht es, die Nachrichtenbox der Ankündigung für die aktuelle Browser-Sitzung zu schließen. Sie wird beim nächsten Zugriff wieder angezeigt.

- Dauerhafte Ablehnungen erlaubt:
  - die Nachrichtenbox erscheint für das aktuelle Mitglied nicht mehr, sobald sie geschlossen wurde.

Django administration

GeoNode

Home > Announcements > Dismissals > Add dismissal

### Add dismissal

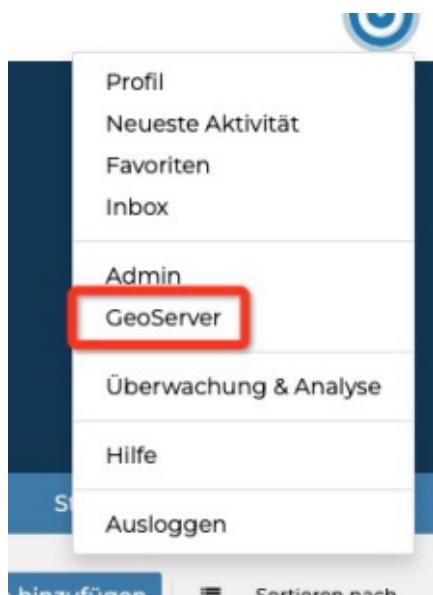
User	*****	▼	→	+
Announcement	Announcement object (1)	▼	→	+
Dismissed at	2021-11-29		14:00:08	

```
<link rel="stylesheet" media="all" href="../styles.css" />
<div id="logo"><a href="https://csgis.de">© CSGIS 2022</a></div>
<div id="menu"></div>
<div id="jumpMenu"></div>
<script src="../menu.js"></script>
<script src="../jumpmenu.js"></script> <!-- the Menu -->
```

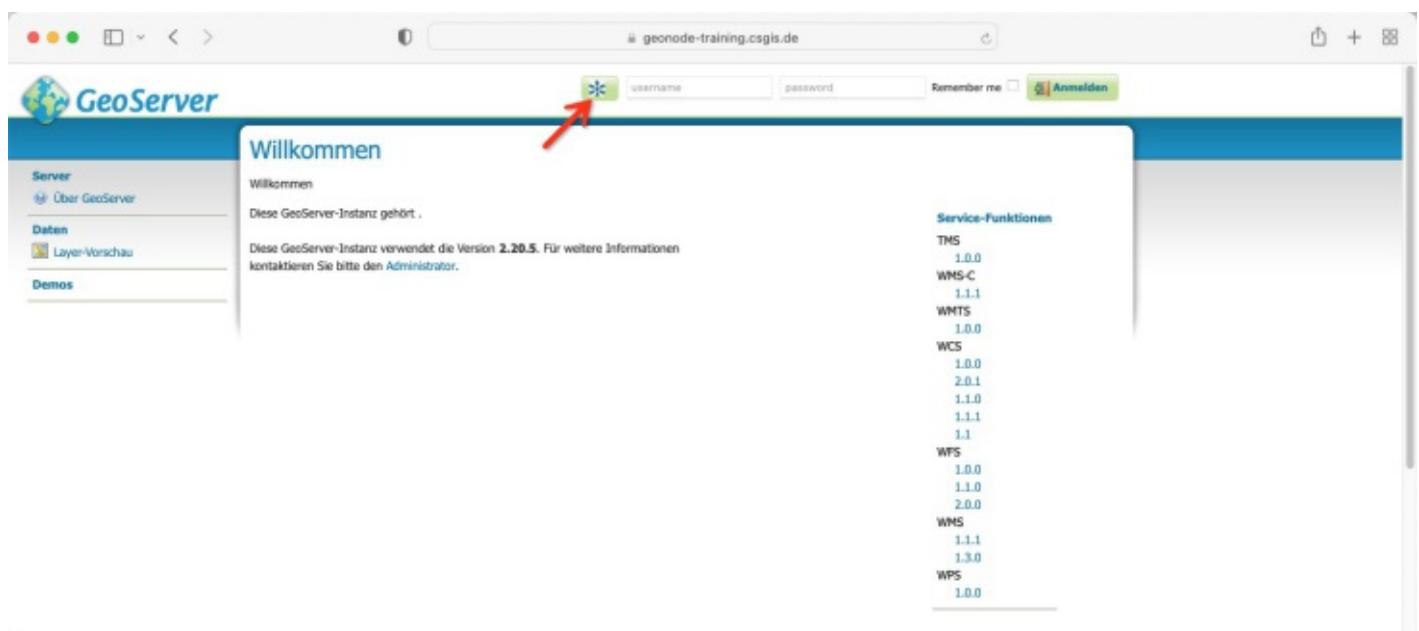
## GeoServer Admin Oberfläche

Neben der “Django Admin Oberfläche” gibt es noch eine zweite wesentliche Oberfläche für Administratoren. Wie wir später noch genauer sehen werden, agiert hinter “Django” der OpenSource-Kartenserver “GeoServer”. Dieser ist für die Bereitstellung der Geo-Daten zuständig.

Wir erreichen die Oberfläche über den Link im oberen rechten Menü.



Über das kleine GeoNode Symbol



können wir uns in die Administrationsoberfläche einloggen.

GeoServer

Willkommen

Angemeldet als admin. [Abmelden](#)

Server

- Serverstatus
- Protokollierung
- Kontaktdaten
- Über GeoServer
- Prozessstatus

Daten

- Layer-Vorschau
- Daten importieren
- Arbeitsbereiche
- Datenspeicher
- Layer
- Gruppenlayer
- Stile
- Backup & Restore

Dienste

- WMS
- WMTS
- WCS
- WFS
- WPS

Einstellungen

- Global
- IAI
- Raster
- Importer

Kartenkacheln-Cache

- Gecachte Layer
- Caching Standards
- GridSets
- Festplattenkontingent
- BlobStores

Sicherheit

- Einstellungen
- Authentifizierung
- Kennwörter

Willkommen

Diese GeoServer-Instanz gehört .

14 Layer [Layer hinzufügen](#)

2 Datenspeicher [Datenspeicher hinzufügen](#)

1 Arbeitsbereiche [Arbeitsbereich hinzufügen](#)

⚠ Der Standard-Benutzer-/Gruppen-Dienst sollte Hash (Digest)-Passwortverschlüsselung verwenden.

ⓘ Starke Kryptographie verfügbar

Diese GeoServer-Instanz verwendet die Version **2.20.5**. Für weitere Informationen kontaktieren Sie bitte den Administrator.

Service-Funktionen

Service	Version
TMS	1.0.0
WMS-C	1.1.1
WMTS	1.0.0
WCS	1.0.0 2.0.1 1.1.0 1.1.1 1.1
WFS	1.0.0 1.1.0 2.0.0
WMS	1.1.1 1.3.0
WPS	1.0.0

geonode-training.csgis.de

### 3. Einführung in die Entwicklung mit GeoNode

---

#### Inhalt

In diesem Bereich werden wir uns mit den ersten Schritten zur Entwicklung mit GeoNode ansehen. Leider ist die GeoNode Infrastruktur technisch alles andere als einfach aufgebaut.

Je nachdem an welchem Bereich Sie arbeiten wollen, werden Kenntnisse in folgenden Punkten benötigt:

Python3	Die Programmiersprache Python
CSS	Cascading Stylesheets – Auszeichnungssprache für HTML
Less	Ein “preprocessor” für CSS
HTML	Hyper Text Markup Language - Definiert die Elemente von Internetseiten
React	Ein Javascript Framework zur Erstellung von Frontends
Redux	Ein Komponente für React um den “state” zu verwalten
Mapstore2	Ein Karten Framework
RSJX	Eine Javascript Bibliothek für “reaktives Javascript”
Django	Ein Webframework basierend auf Python
Jinja2	Eine Templatesprache für Django
sql	Die von dem Datenbankserver verwendete Sprache, um Abfragen zu tätigen
git	Eine Versionsverwaltung von Code
GIS	spezifisches Wissen bezüglich räumlicher Daten

# GeoNode Quelltext

Um im folgenden Änderungen oder Neuentwicklungen an unserer Instanz vorzunehmen, werfen wir zunächst einen Blick auf das Code Repository von GeoNode. Dieses findet sich unter der URL:

<https://github.com/geonode/geonode>

Der eigentliche Programm-Code befindet sich hier im Ordner "geonode".

The screenshot shows the GitHub repository page for 'GeoNode / geonode'. The 'Code' tab is active, displaying the master branch. There are 27 branches and 464 tags. A list of recent commits is shown, with the 'geonode' folder highlighted by a red box. The right sidebar contains an 'About' section with a brief description of GeoNode as an open source platform for geospatial data, and sections for 'Readme', 'Licenses', 'Code of conduct', 'Star', 'Watch', and 'Fork'. The 'Releases' section shows a latest version of 4.0.1.

Die sich hierunter befindenden Ordner, stehen für sogenannte “Django-Apps”. Wir erinnern uns, Django, das Web-Framework hinter Django ermöglicht, die einzelnen Programmzbereiche in Module aufzuteilen.

The screenshot shows the GitHub repository page for 'GeoNode / geonode /'. The 'Code' tab is active, displaying the master branch. The commits are listed under subfolders: api, base, br, catalogue, client, documents, favorite, and metadata. The commits are from authors like marthamareal and giohappy, with messages like 'Modify Metadata form with permissions check' and dates ranging from 3 days ago to 2 months ago.

Dem/der aufmerksamen Zuhörer\*in wird nun auffallen, dass sich die Namen der einzelnen Ordner, im “Django Admin” wiederfinden. Grund hierfür ist, dass Django die Administrationsoberfläche automatisiert aus den einzelnen Apps erstellt.

Betrachten wir uns eine App wie beispielsweise die “people”-App, die die Benutzer des Portals regelt, sollen 3 Dateien mit besonderem Augenmerk beachtet werden.

- models.py
- templates/\*

- views.py

Diese Struktur bezeichnet sich als MVT (Model-View-Template). Ein Programmiermuster das die Grundlage aller Django-Apps darstellt.

## models.py

Hierbei stellt die *models.py* die jeweilige Datenbank-Definition dar. Hierüber werden also alle benötigten Tabellen und Funktionen rund um die Datenbank definiert.

## templates/\*

Im Ordner *templates*, befinden sich alle Dateien, die für die Ein-/Ausgabe zuständig sind. Dies sind größtenteils HTML Dateien die über die Templatesprache “Jinja2” die Ausgabe im Browser besorgen.

## views.py

*views.py* stellt Funktionen, die als Bindeglied zwischen Datenbankdefinitionen (*models.py*) und Ein-/Ausgabe Ansichten (*templates/\**) stehen.

master   geonode / geonode / people /		
		Go to file
marthamareal [Fixes #9919] User endpoint return 404 if user is not in registered m...	✓ 2e2bc67 on 30 Aug	History
... fixtures management migrations static/people/img <b>templates/people</b> templatetags __init__.py admin.py languages.py <b>models.py</b> <b>search_indexes.py</b> <b>urls.py</b> ← <b>views.py</b>	GeoNode test case refactoring [Flying] Replace "+" concatenated strings with f"strings" [Fixes #7984] Remove Registered Members group profile (#7985) Added the icons for people profile in correct path. Added the necessa... [Fixes #9748] Cleanup of stale code (#9693) [Fixes #8504] Map Thumbnails Broken GNIP 89: Architecture Design - Resource and Storage Manager Modules (#... [Fixes #7842] Rename layer to dataset (#7841) GNIP 89: Architecture Design - Resource and Storage Manager Modules (#... [Hardening] Few consistency checks on some models properties GNIP 88: Architecture Design - Resource and Storage Manager Modules (#... [Fixes #9381] Hide users to anonymous users and restrict visibility f... [Fixes #9381] Hide users to anonymous users and restrict visibility f...	5 years ago 2 years ago 13 months ago 10 years ago 2 months ago 3 years ago 15 months ago 15 months ago 15 months ago 11 months ago 15 months ago 4 months ago 4 months ago

Beachtenswert ist ebenfalls noch die sich sogut wie in allen Apps befindliche Datei “urls.py”. Sie enthält die URL Definitionen, um Browseranfragen an die richtige Stelle zu leiten.

## GeoNode Entwicklungsumgebung

Eine grundlegende Regel bei der Entwicklung mit GeoNode ist, den Code-Kern, wenn möglich nie anzufassen. <https://github.com/GeoNode/geonode-project> gibt uns jedoch die Möglichkeit einzelne Bereiche in GeoNode zu überschreiben. Zudem kann das Projekt lokal gestartet werden, um Änderungen direkt live verfolgen zu können oder automatisierte Tests ablaufen zu lassen.

Da es unter Windows zu vielerlei Problemen mit abhängigen Bibliotheken kommt, soll die lokale Entwicklungsumgebung lediglich unter Linux oder OSX gestartet werden.

## Neues GeoNode Project erzeugen

```
git clone https://github.com/GeoNode/geonode-project.git -b master
source /usr/share/virtualenvwrapper/virtualenvwrapper.sh
mkvirtualenv --python=/usr/bin/python3 my_geonode
pip install Django==3.2.14

django-admin startproject --template=./geonode-project -e py,sh,md,rst,json,yml,ini,env,sample,pre
cd geonode_training_dev
```

Beachtenswert ist hierbei die Datei `src/requirements.txt` sie zeigt, dass der GeoNode Kern Quelltext in Form eines Python-Pakets als Abhängigkeit installiert wird.

```
git+https://github.com/GeoNode/geonode.git@master#egg=GeoNode
```

## Settings erzeugen

Das Herzstück der Konfiguration ist die Datei `src/settings.py` (wir werden Sie im letzten Abschnitt noch im Detail betrachten), die über Umgebungsvariablen in der Datei `.env` gespeist wird.

Wir erzeugen die Datei `.env` mit folgendem Befehl:

```
python create-envfile.py --geoserverpwd geoserver
```

## Datenbank anlegen

Anders als in der README des GeoNode-Project Repository müssen wir vorab noch zwei Postgres Datenbanken anlegen.

```
-- Geodatenbank anlegen
CREATE USER geonode_training_dev_data;
ALTER USER geonode_training_dev_data with encrypted password '< GEONODE_DATABASE_PASSWORD aus .ov
CREATE DATABASE geonode_training_dev_data;
GRANT ALL PRIVILEGES ON DATABASE geonode_training_dev_data TO geonode_training_dev_data;

-- Django Datenbank anlegen
CREATE USER geonode_training_dev;
ALTER USER geonode_training_dev with encrypted password '< GEONODE_GEODATABASE_PASSWORD aus .over
CREATE DATABASE geonode_training_dev;
GRANT ALL PRIVILEGES ON DATABASE geonode_training_dev TO geonode_training_dev;

-- in beiden Datenbanken Postgis installieren
CREATE EXTENSION IF NOT EXISTS postgis;
GRANT ALL ON geometry_columns TO PUBLIC;
GRANT ALL ON spatial_ref_sys TO PUBLIC;
```

## Abhängigkeiten installieren

Ebenfalls ist es erforderlich noch weitere Bibliotheken zu installieren sowie einige Management Dateien zu generieren:

```

cd src
pip install -r requirements.txt --upgrade
pip install -e . --upgrade

# Install GDAL Utilities for Python
pip install pygdal=="`gdal-config --version`.*"

# Dev scripts
mv ../../override_dev_env.sample ../../override_dev_env
mv manage_dev.sh.sample manage_dev.sh
mv paver_dev.sh.sample paver_dev.sh

source ../../override_dev_env

```

## Starten des Servers

Mit folgenden Befehlen sollten wir abschließend den lokalen Server starten können.

```

# Using the Default Settings
sh ./paver_dev.sh reset
sh ./paver_dev.sh setup
sh ./paver_dev.sh sync
sh ./paver_dev.sh start

```

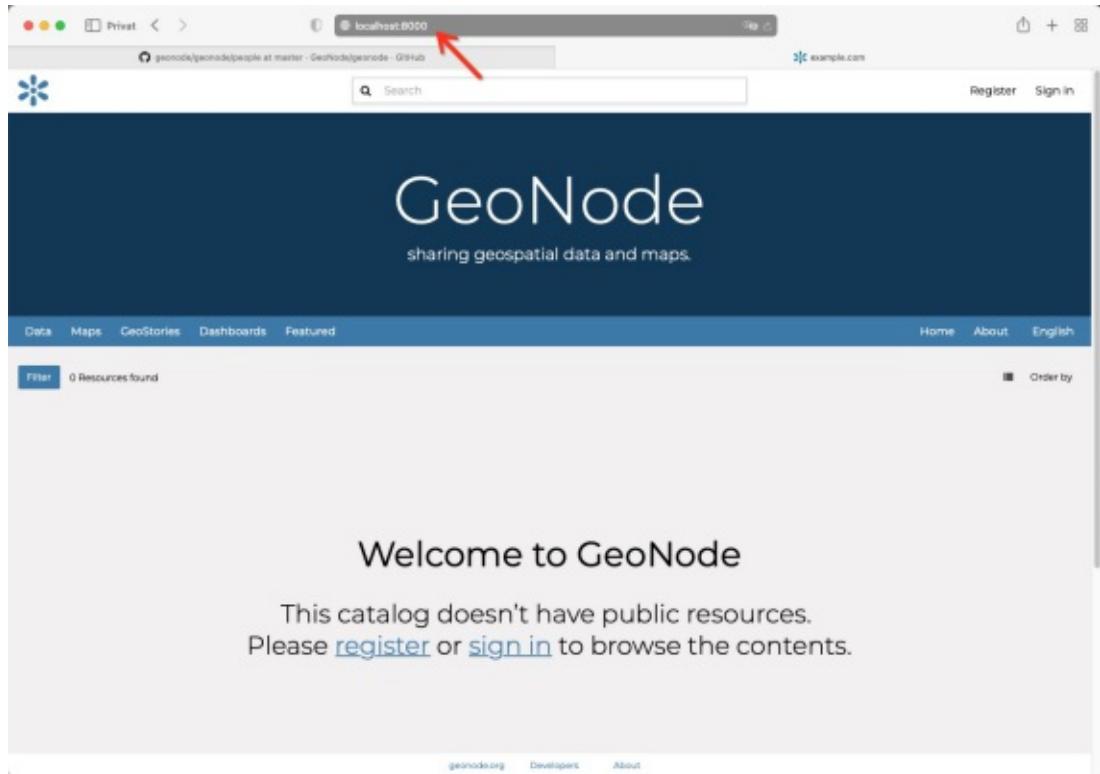
Wenn alles korrekt verlaufen ist, ist in der Terminal-Ausgabe folgende Meldung zu sehen:

```

WARNING: Illegal reflection access by Javassist.util.proxy.SecurityActions (file:/Users/tonischenbuehner/dev/github/gr-suth3/ils/gesserver/geoserver/WEB-INF/lib/javassist-3.12.0.GA.jar)
        to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,java.security.ProtectionDomain)
WARNING: Please consider reporting this to the maintainers of Javassist.util.proxy.SecurityActions
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
2022-08-03 12:15:14.095:INFO[oejs]ContextHandler.main: Started o.e.j.w.WebAppContext@2aaafab@(server,/,geoserver,file:///Users/tonischenbuehner/dev/github/gr-suth3/ils/geoserver/geoserver.war/-/geoserver)
2022-08-03 12:15:14.095:INFO[oejs]AbstractConnector:main: Started o.e.j.w.WebAppContext@2aaafab@(server,/,geoserver,jetty.log)
2022-08-03 12:15:14.094:INFO[oejs]AbstractConnector:main: Started ServerConnector@22a71001[HTTP/1.1,(http/1.1)]@0.0.0.0:8080
2022-08-03 12:15:14.095:INFO[oejs]ServerMain: Started @1999ms
2022-08-03 12:15:14.244:INFO[oejs]Geoserver:ctpl586270994-31: Initializing Spring DispatcherServlet 'dispatcher'
The logs are available at /Users/tonischenbuehner/dev/github/gr-suth3/ils/gesserver/jetty.log
---> paverdev.start.gis.server
---> paverdev.start.geoserver
python -m ignore manage.py runserver 0.0.0.0:8000 &
Performance system checks...
System check identified no issues (0 silenced).
October 03, 2022, 12:15:22
Geoserver version 2.2.2 using settings 'ils.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
Geoserver is now available.

```

Weiterhin kann das Projekt im Browser unter <http://localhost:8000> aufrufen werden.



## Weiterführende Links

---

- [GeoNode Docs – GeoNode Project](#)
- [GeoNode Docs – Docker als Entwicklungsumgebung](#)

# Templates überschreiben

In diesem sehr einfachen Beispiel werden wir die HTML Ausgabe eines Django Templates überschreiben.

Unter der Seite `/about` (z.B.: <https://geonode-training.csgis.de/about/>) publiziert GeoNode eine Seite, die über Sinn und Zweck der Plattform informiert.

Diese Seite generiert sich wie folgt:

## URL Defintion

Die Datei `url.py` verlinkt alle Anfragen an `about` über die durch Django bereitgestellte Funktion `TemplateView`, direkt mit der statischen HTML Datei `about.html` im Templates Ordner

```
url(r'^about/$',
    TemplateView.as_view(template_name='about.html'),
    name='about'),
```

[Code auf GitHub](#)

## Neue `about.html` anlegen

In Geonode-Projekt können wir diese Datei nun einfach überschreiben, indem wir im Ordner “Templates” eine Datei mit identischem Namen anlegen.

`templates/about.html`

```
{% extends "geonode_base.html" %}
{% load i18n %}
{% block title %}Unsere Forschungsdaten{% endblock %}

{% block body_outer %}
<div class="page-header">
<h2>Unsere Forschungsdaten</h2>
</div>
<p>
Diese Plattform dient der Publikation von Daten zum Thema Weltraum.</p>
<p>
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Inventore similique asperiores i...
<p>
Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit sint, labore sequi autem i...
{% endblock %}
```

Da der Entwicklungsserver nach jeder Änderung automatisch neu startet, sollten die Änderungen hiernach direkt im Browser sichtbar werden:

The screenshot shows a web browser window with the URL 'geonode/geonode/people.git master - GeoNode/geonode - GitHub'. The page title is 'Unsere Forschungsdaten'. The header includes a search bar, 'Register', 'Sign In', and language links for 'Home', 'About', and 'English'. Below the header, there's a navigation menu with 'Data', 'Maps', 'GeoStories', 'Dashboards', and 'Featured'. A main content area titled 'Unsere Forschungsdaten' contains placeholder text about research data publication. At the bottom, there are footer links for 'geonode.org', 'Developers', and 'About'.

In einem live System sind folgende Schritte nötig:

Entweder:

- neuen Build des Django Image erstellen (docker-compose build django; docker-compose up -d django)
- oder host Verzeichnis in Docker Container mounten: <https://github.com/GeoNode/geonode-project/blob/master/docker-compose.yml#L11> und den Container neu starten (docker-compose up -d django).

Tipp: Sie können nicht nur Templates, sondern so auch die Ausgabe der E-Mails überschreiben.

## Weiterführende Links

- [Das Django Template System](#)
- [GeoNode Docs – Templates überschreiben](#)
- [GeoNode Docs – Homepage anpassen](#)
- [GeoNode Docs – Frontend Development](#)

# Eigene Seite anlegen

Im vorherigen Abschnitt haben wir die bestehende Seite "about.html" überschrieben. In diesem Kapitel wollen wir eine neue Seite anlegen.

Hierfür werden wir eine neue HTML Datei erstellen sowie die urls.py verwenden, um die neue Seite zu verlinken.

## dsgvo.html anlegen

Zunächst legen wir wiederum im Templates Ordner unsere neue HTML Datei dsgvo.html an.

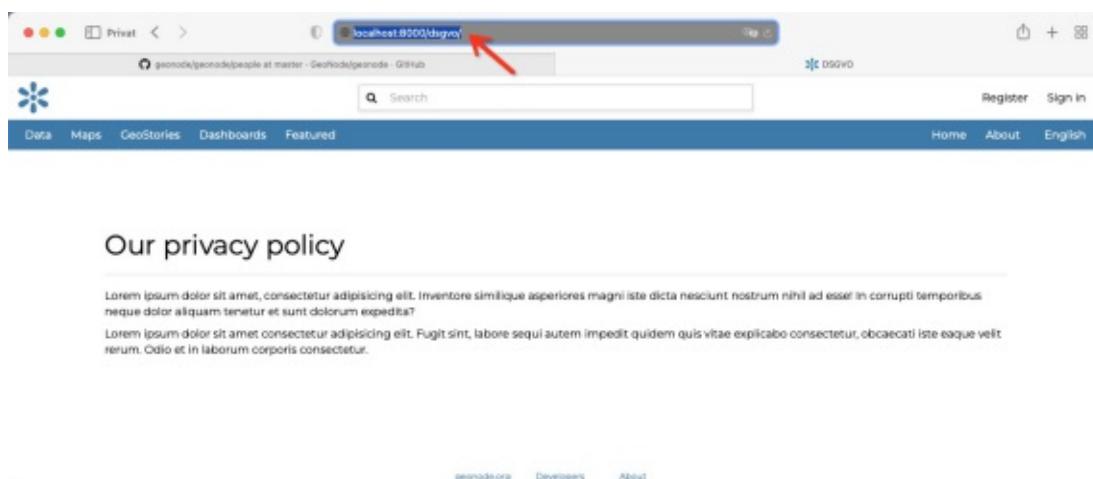
```
{% extends "geonode_base.html" %}  
{% load i18n %}  
{% block title %}DSGVO{% endblock %}  
  
{% block body_outer %}  
    <div class="page-header">  
        <h2>{% trans "Our privacy policy" %}</h2>  
    </div>  
    <p>  
        Lorem ipsum dolor sit amet, consectetur adipisicing elit. Inventore similique asperiores i  
        <p>  
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit sint, labore sequi autem i  
    {% endblock %}
```

## url.py bearbeiten

In einem zweiten Schritt verlinken wir eine URL-Anfrage an dsgvo mit unserer neu angelegten Datei:

```
from geonode.urls import urlpatterns  
from django.views.generic import TemplateView  
from django.conf.urls import url  
  
# You can register your own urlpatterns here  
urlpatterns = [  
    url(r'^dsgvo/$',  
        TemplateView.as_view(template_name='dsgvo.html'),  
        name='dsgvo'),  
] + urlpatterns
```

Hiernach sollte die neue DSGVO Seite im Browser sichtbar werden.



Tipp: in Kombination mit einem “Custom Menu” lassen sich so einfach eigene Seiten erzeugen.

## Weiterführende links

---

- [Django Docs - URLs definieren](#)

# Übersetzungen anlegen

In diesem Abschnitt wollen wir einen Blick auf das Übersetzungssystem von Django werfen. Ziel ist unsere vorab als zu übersetzend ausgezeichnete Überschrift

```
<h1>{% trans "Our privacy policy" %}</h1>
```

mit einer Übersetzung zu versehen. Django erlaubt das Anlegen von Übersetzungen über sogenannte `po` und `mo` Dateien.

Die `po` Datei stellt hierbei die Quelldatei. Die `mo` Datei die kompilierte Datei mit Übersetzungen dar.

Zunächst können wir den Umfang der Sprachen in unserer `settings.py` Datei wie folgt einschränken

```
LANGUAGES = (
    ('en-us', 'English'),
    ('de-de', 'Deutsch'),
)
```

## Locale Dateien anlegen

Im Hauptverzeichnis unseres GeoNode-Projekts legen wir den neuen Ordner `locale` an.

```
cd .. # wir verlassen den Ordner src
(e) tonischonbuchner@Tonis-MacBook-Pro ~ /dev/geonode4_project/geonode_training_dev/src $% mkdir .
```

Über das durch Django bereitgestellte Kommando `makemessages` lassen wir alle zur Übersetzung angelegten Textfragmente in `po` Dateien einsammeln:

```
django-admin makemessages --no-location -l en -l de -d django -e "html"
```

`--locale` gibt hierbei die zu erzeugenden Übersetzungen an. `-e "html"` gibt die Extension der Dateien an, die wir berücksichtigen wollen

Hiernach sollten wir die neuen Ordner je Sprache sowie die Quelldateien im Ordner “locale” vorfinden.

```
(e) tonischonbuchner@Tonis-MacBook-Pro $% tree locale
locale
├── de
│   └── LC_MESSAGES
│       └── django.po
└── en
    └── LC_MESSAGES
        └── django.po

4 directories, 2 files
```

## po Dateien bearbeiten

Im nächsten Schritt legen wir in `locale/de/LC_MESSAGES/django.po` eine Übersetzung für unsere Überschrift an:

```
...
msgid "Our privacy policy"
msgstr "Unsere Datenschutzgrundverordnung"
...
```

## po Dateien in mo Dateien kompilieren

Nachdem wir die Quelldateien übersetzt haben können wir Django anweisen die Dateien zu kompilieren.

```
django-admin compilemessages --locale de --locale en
processing file django.po in /Users/tonischonbuchner/dev/github/gn-auth3/geonode_training/locale/
processing file django.po in /Users/tonischonbuchner/dev/github/gn-auth3/ils/geonode_training/de/
```

Hiernach sollten wir in unserem `locale` Ordner die neu kompilierten `*.mo` Dateien sehen.

```
locale
└── de
    └── LC_MESSAGES
        ├── django.mo
        └── django.po
└── en
    └── LC_MESSAGES
        ├── django.mo
        └── django.po
```

Um die Änderungen online zu stellen, kopieren wir den `locale` Ordner eine Ebene tiefer ins `src` Verzeichnis.

Das Projektverzeichnis erhält beim erstellen von GeoNode-Project den von Ihnen angegebenen Namen. Es handelt sich hierbei um dieses Verzeichnis: [https://github.com/GeoNode/geonode-project/tree/master/src/project\\_name](https://github.com/GeoNode/geonode-project/tree/master/src/project_name)

```
cp -r locale src/geonode_training_dev
```

Wir starten den Development-Server neu:

```
paver stop
paver start
```

Hiernach sollten die Übersetzungen der Sprachen im Browser angezeigt werden.

Vergessen Sie in live Umgebungen nicht den Python Server neu zu starten

The screenshot shows a web browser window with a blue header bar. The title bar says "geonode:people at master · GeoNode:geonode · Bitbucket". The header includes a "Search" input field, "Register", and "Sign in" buttons. Below the header is a navigation menu with links: "Data", "Maps", "GeoStories", "Dashboards", and "Featured". The main content area has a heading "Our privacy policy" followed by two paragraphs of placeholder text: "Lorem ipsum dolor sit amet, consectetur adipisciing elit. Inventore similique asperiores magni iste dicta nesciunt nostrum nihil ad esse. In corrupti temporibus neque dolor aliquam tenetur et sunt dolorum expedita?Lorem ipsum dolor sit amet consectetur adipisciing elit. Fugit sint, labore sequi autem impedit quidem quis vitae explicabo consectetur, obcaecati iste eaque velit. rerum. Odio et in laborum corporis consectetur."

The screenshot shows a web browser window with a blue header bar. The title bar says "geonode:people at master · GeoNode:geonode · Bitbucket". The header includes a "Search" input field, "Registerieren", and "Anmelden" buttons. Below the header is a navigation menu with links: "Daten", "Karten", "GeoStories", "Dashboards", and "Beachtenwert". The main content area has a heading "Unsere Datenschutzverordnung" followed by two paragraphs of placeholder text: "Lorem ipsum dolor sit amet, consectetur adipisciing elit. Inventore similique asperiores magni iste dicta nesciunt nostrum nihil ad esse. In corrupti temporibus neque dolor aliquam tenetur et sunt dolorum expedita?Lorem ipsum dolor sit amet consectetur adipisciing elit. Fugit sint, labore sequi autem impedit quidem quis vitae explicabo consectetur, obcaecati iste eaque velit. rerum. Odio et in laborum corporis consectetur."

## Weiterführende Links

- [Django Docs – Internationalization and localization](#)

# Export geopackage APP

In diesem Abschnitt wollen wir uns die Grundzüge einer eigenen App ansehen.

Unsere APP wird Usern die Möglichkeit geben Vektor-Datensätze auszuwählen und diese gesammelt in einem geopackage herunterzuladen.

Die Hauptarbeit wird hierbei von dem im Hintergrund agierenden Kartenserver `geoserver` erledigt. Dieser stellt über den [WPS Endpunkt](#) bereits die Möglichkeit, Ebenen als Geopackage zu exportieren. Als Input erwartet er ein XML Dokument mit den Layer Definitionen.

ACHTUNG: Der gezeigte Code ist nicht produktionsfähig!

Er vermisst wichtige Sicherheitsabfragen und Fehlerbehandlungen.

Programm Ablauf:

```
-> User wählt Datensätze zum Export über Checkboxen  
-> JS sendet Namen der Ebenen an JSON View als POST request  
-> Django sendet request an GeoServer WPS Endpunkt um das geopackage erstellen zu lassen  
-> JS nimmt link zu Geopackage Download entgegen und gibt diesen an User aus
```

## Eine neue Django APP erstellen

Django macht uns den Start für eigene Entwicklungen sehr leicht. Das Kommando

```
python manage.py startapp geopackage_collection
```

erzeugt uns einen neuen Ordner innerhalb unserer Verzeichnisstruktur mit den wichtigsten Dateien wie `views.py`, `models.py` etc. bereits angelegt.

```
geopackage_collection  
└── __init__.py  
└── admin.py  
└── apps.py  
└── helper.py  
└── migrations  
    └── __init__.py  
└── models.py  
└── templates  
    └── base  
        └── resourcebase_list.html  
└── tests.py  
└── views.py
```

5 directories, 16 files

Unsere neue App müssen wir lediglich noch in den Django settings vermerken:

```
if PROJECT_NAME not in INSTALLED_APPS:  
    INSTALLED_APPS += (PROJECT_NAME,)  
    INSTALLED_APPS += ('geopackage_collection',)
```

## Anlegen der `views.py`

Das Bindeglied zwischen Datenbank und Ausgabe stellt die Datei `views.py` dar.

Wir legen diese mit folgendem Inhalt an:

```
from django.views.generic.list import ListView
from django.http import JsonResponse
from django.contrib.auth.decorators import login_required
from django.contrib.auth.mixins import LoginRequiredMixin
from django.views.decorators.http import require_http_methods
from geonode.base.models import ResourceBase
from .helper import get_wfs_Template, http_client
import json

class GpkgList(LoginRequiredMixin, ListView):
    """Django List View um alle Datensätze auszugeben"""
    queryset = ResourceBase.objects.filter(subtype="vector")

    @login_required
    @require_http_methods(['POST'])
    def gpkg_json_result(request):
        """View um Datensätze entgegen zu nehmen und
        die URL zum Datensatz zurück zu geben.
        """
        request_payload = json.loads(request.body.decode("utf-8"))
        datasets = request_payload['datasets']
        wfs_template = get_wfs_Template(datasets)
        wps_return = http_client(wfs_template)
        data = {"result_link": wps_return}

        return JsonResponse(data, safe=False)
```

## Anlegen der helpers.py

Die angelegte views.py greift auf einige Hilfefunktionen zurück. So zum Beispiel eine Funktion, die das XML für die Anfrage an GeoServer erstellt, oder einen einfach abstrahierten HTTP Klienten. Wir legen die Funktionen in einer neuen Datei namens `helpers.py` an.

```
from django.conf import settings
import requests
import logging

# Neuen Logger erzeugen
logger = logging.getLogger("geonode")

# Zugangsdaten zum GeoServer für http_client()
geoserver_user = settings.OCG_SERVER_DEFAULT_USER
geoserver_user_password = settings.OCG_SERVER_DEFAULT_PASSWORD
geoserver_settings_url = settings.GEOSERVER_WEB_UI_LOCATION
geoserver_url = f"{geoserver_settings_url}wps"

wfs_template = """<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS">
    <ows:Identifier>gs:GeoPackage</ows:Identifier>
    <wps:DataInputs>
        <wps:Input>
            <ows:Identifier>contents</ows:Identifier>
            <wps>Data>
                <wps:ComplexData mimeType="text/xml; subtype=geoserver/geopackage">
                    <![CDATA[
                        <geopackage xmlns="http://www.opengis.net/gpkg"
                            {partials}
                        </geopackage>
                    ]]>
                </wps:ComplexData>
```

```

        </wps:Output>
    </wps:DataInputs>
<wps:ResponseForm>
    <wps:RawDataOutput>
        <ows:Identifier>geopackage</ows:Identifier>
    </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
"""

def get_build_partials(datasets):
    """Erzeuge für jede zu ladende Ebene einen XML Knoten."""
    partials_source = '<features identifier="{dataset}" name="places">' \
                      '<featuretype>{dataset}</featuretype>' \
                      '</features>'

    partials_payload = ""
    for dataset in datasets:
        partials_payload += partials_source.format(dataset = dataset)

    return partials_payload

def get_wfs_Template(datasets):
    """Gibt das fertige XML Dokument an den View zurück."""
    generated_partials = get_build_partials(datasets)
    wfs_payload = wfs_template.format(partials = generated_partials)
    return wfs_payload

def http_client(geoserver_payload):
    """Hilfefunktion um HTTP POST request abzusetzen."""
    try:
        geoserver_response = requests.post(
            geoserver_url,
            data=geoserver_payload.encode('utf-8'),
            auth=(
                geoserver_user,
                geoserver_user_password
            ))

        if (geoserver_response.status_code < 200 or geoserver_response.status_code > 201):
            logger.error(geoserver_response.content)
        return geoserver_response.text
    except Exception as e:
        raise e

```

## Template anlegen

Nun erstellen wir im Verzeichnis unserer App ein neues Verzeichnis namens `templates` , hierin ein Verzeichnis mit dem Namen `base` und legen die Datei `resourcebase_list.html` mit folgendem Inhalt an:

```

{% extends "geonode_base.html" %}

{% block body_outer %}
<h2>GeoPackage erzeugen</h2>
<hr>
<table class="table">
    <thead>
        <tr>
            <th scope="col">Auswahl</th>

```

```

    <tr scope="col">Auswahl</tr>
    <th scope="col">Vector Dataset</th>
  </tr>
</thead>
<tbody>
  {% for object in object_list %}
  <tr>
    <th scope="row"><input type="checkbox" name="layerselect" value="{{ object.alternate }}>
    <td>{{ object.title }}</td>
  </tr>
  {% empty %}
    No objects yet.
  {% endfor %}
</tbody>
</table>
<div id="result"></div>
<button id="submit">Generiere Geopackage von Auswahl</button>

<script>

/* Hilffunktion um korrekten Cookie zu denden */
function getCookie(name) {
  var cookieValue = null;
  if (document.cookie && document.cookie !== '') {
    var cookies = document.cookie.split(';');
    for (var i = 0; i < cookies.length; i++) {
      var cookie = cookies[i].trim();
      if (cookie.substring(0, name.length + 1) === (name + '=')) {
        cookieValue = decodeURIComponent(cookie.substring(name.length + 1));
        break;
      }
    }
  }
  return cookieValue;
}

const result_div = document.getElementById("result");
/* Auf Button klick auswahl senden und Resultat in Empfang nehmen */
var element = document.getElementById("submit")
.onclick = function() {
  let node_values = [];
  var checkedBoxes = document.querySelectorAll('input[name=layerselect]:checked');
  checkedBoxes.forEach((node) => {
    node_values.push(node.value)
  });

  let data = JSON.stringify({"datasets": node_values});
  let csrftoken = getCookie('csrftoken');
  let response = fetch("/gpkg/gpkg_json_result/", {
    method: 'POST',
    body: data,
    headers: { 'Accept': 'application/json, text/plain, */*',
               'Content-Type': 'application/json',
               "X-CSRFToken": csrftoken},
  })
  .then(response => response.json())
  .then(res => {
    result_div.innerHTML = `<a href='${res["result_link"]}' target='_blank'>Geopackage</a>`;
    this.remove()
  }).catch(err => {
    console.log(err)
    result_div.innerHTML = err
  });
};

</script>
{% endblock %}

```

## urls.py

Abschließend verknüpfen wir unsere views.py mit unseren URL-Definitionen der Datei urls.py .

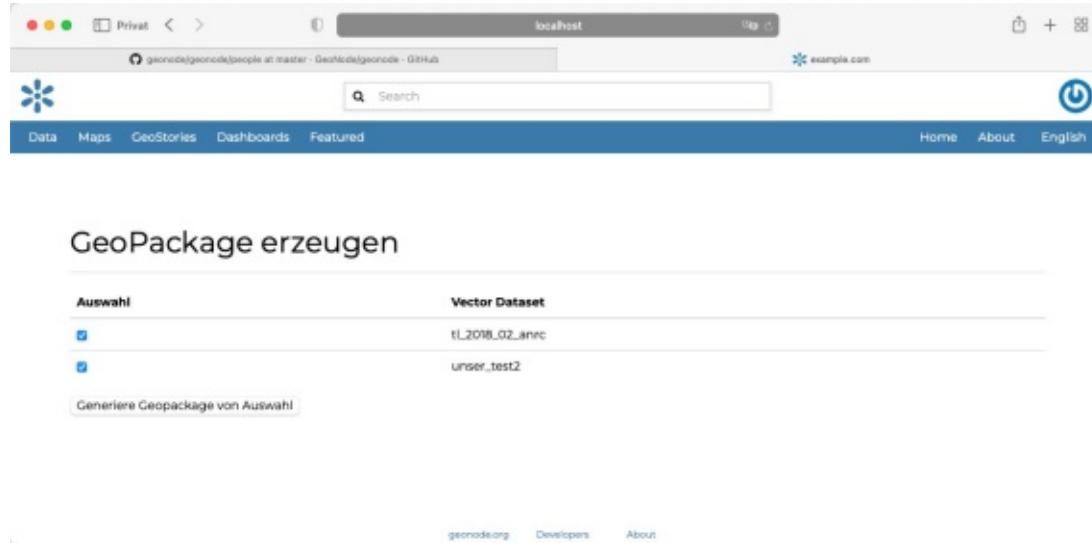
```
from geonode.urls import urlpatterns
from django.views.generic import TemplateView
from django.conf.urls import url

from geopackage_collection.views import GpkgList, gpkg_json_result
from django.urls import path

# You can register your own urlpatterns here
urlpatterns = [
    url(r'^gpkg/$',
        GpkgList.as_view(),
        name='gpkg'),
    url(r'^gpkg/gpkg_json_result/$',
        gpkg_json_result,
        name='gpkg_json_result'),
] + urlpatterns
```

## Ergebnis

Unter unserer neuen URL `http://localhost:8000/gpkg` sollten wir alle Vektor Datensätze aufgelistet bekommen.



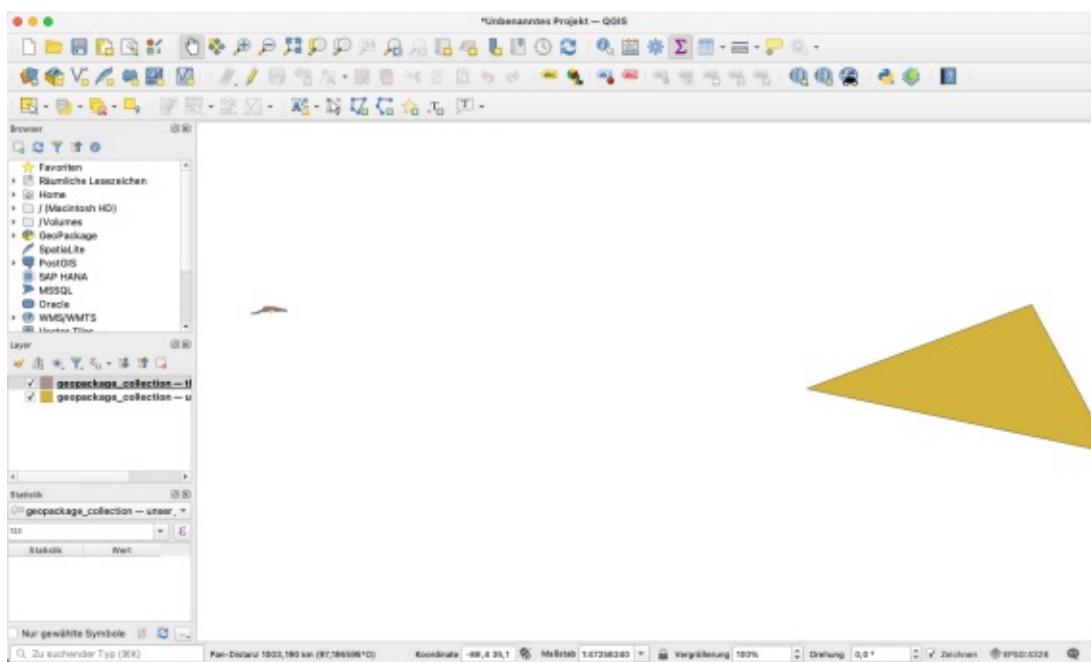
Auswahl	Vector Dataset
<input checked="" type="checkbox"/>	tL2018_02_anec
<input checked="" type="checkbox"/>	unser_test2

Generiere Geopackage von Auswahl

Treffen wir eine Auswahl und klicken "Generiere Geopackage von Auswahl" sollte der Link zum Download des Geopackage nach einiger Zeit angezeigt werden:

The screenshot shows the GeoNode interface for creating a GeoPackage. On the left, there's a sidebar with 'Auswahl' (Selection) and 'Vector Dataset' sections. Under 'Auswahl', there are two checked items: 'tL2018\_02\_anec' and 'unser\_test2'. Below these is a button labeled 'Geopackage herunterladen' (Download Geopackage). A red arrow points to this button.

Öffnen wir die heruntergeladene Datei in QGIS sehen wir, die beiden exportieren Ebenen:



## Fazit

Wie wir gesehen haben, ist es relativ einfach auf vorhandene Datensätze zuzugreifen oder eigenen Seiten zu erstellen die Nutzereingaben entgegennehmen. Das Thema Entwicklung unter GeoNode bleibt dennoch komplex, verlangt es dem Entwickler doch ein breites Wissen in unterschiedlichen Bereichen ab.

Selbstverständlich ließe sich dieses Beispiel noch weiter ausarbeiten. Zum Beispiel ein eigenes Datenbankmodell erstellen, um Daten zu speichern und die Django Administrationsoberfläche zu verwenden.

## Weiterführende Links

- [Einführung in Django Apps](#)
- [Django Datenbank Abstraktion](#)
- [GeoServer WPS Endpunkt Dokumentation](#)

# Geändertes Frontend in GeoNode Version 4

In den vorausgehenden Kapiteln haben wir gesehen, wie wir Templates überschreiben können. Vergleichen wir jedoch die Oberfläche der GeoNode Version 3.3 mit der Oberfläche der Version 4, stechen große Unterschiede ins Auge.

The screenshot shows the 'Welcome' page of GeoNode Version 4.0. At the top, there's a navigation bar with links for 'Daten', 'Karten', 'Apps', 'Über', 'Geostory', 'Suchen', 'Registrieren', and 'Anmelden'. Below the navigation is a large blue header with the word 'Welcome' in white. Underneath it, a sub-header states 'GeoNode is an open source platform for sharing geospatial data and maps.' There's a button labeled 'Erste Schritte >'. The main content area has a dark background with a search bar containing a magnifying glass icon and the text 'Suchen'. Below the search bar is a link 'Erweiterte Suche'. On the left side, there's a diamond-shaped icon and the text '2156 Ebenen'. On the right side, there's a location pin icon and the text '310 Karten'. At the bottom, a note says 'Hier können Sie nach Geodaten suchen, die von anderen Benutzern... Die Daten sind zum durchsuchen, verschneiden und stylen verfügbar.'

The screenshot shows the homepage of GeoNode Version 3.3. The top navigation bar includes 'Daten', 'Karten', 'GeoStories', 'Dashboards', 'Beachmenwert', 'Startseite', 'Über', 'Test menu 01', and 'Deutsch'. The main title 'GeoNode' is prominently displayed with the subtitle 'sharing geospatial data and maps.' Below the title, there are four featured items: 'Lotes\_AdR\_10k\_pontos' (by guilocalpettari@gmail.com), 'AG\_PRESERVE' (by DavidQuartz), '7 m ASL' (by Ticino Wildfires Dashboard), and 'Pineland\_Residential\_clipped\_from\_co...' (by HAO). The bottom section displays a grid of four map thumbnails: 'Ciudad de México Mexico' (geonode.org), 'Developers' (geonode.org), 'About' (geonode.org), and another map.

Der Vergleich zeigt dass, die Oberfläche für die Version 4 weitreichende Änderungen erfahren hat.

Django Templates durch React ersetzt

Da bereits die Kartenkomponente von GeoNode3 auf dem [React](#) basierte Framework [Mapstore2](#) basierte, wurde entschieden weitere Komponenten (wie die Startseite) ebenfalls hierüber abwickeln zu lassen.

React ist eine JavaScript-Programmbibliothek zur Erstellung von webbasierten Benutzeroberflächen.

Das bedeutet leider, dass nicht mehr alle Templates, die vormals durch Django ausgegeben wurden, wie gezeigt überschrieben werden können. Für diese Seitenbereiche ist es unter Umständen erforderlich, die Mapstore2 Komponente zu “forken” und einen eigenen Build zu erstellen.

Aufgrund der Komplexität der Komponente ist die Entwicklung hier nur *sehr* erfahrenen Javascript Entwicklern anzuraten.

## Weiterführende Links

---

- [Github GeoNode Mapstore Client Readme](#)
- [Video Tutorial Developing with Mapstore](#)

## 4. System Administration

---

In diesem Abschnitt beschäftigen wir uns mit der eigentlichen Systemarchitektur.

Wir werden sehen, welche Komponenten wie zusammenhängen.

Weiterhin einen kurzen Blick auf Backups und “Command Line” Kommandos und weitere Möglichkeiten für Systemadministratoren werfen.

User, die in diesem Bereich arbeiten sollten ein fundiertes Wissen in

- Linux Administration
- Docker
- shell

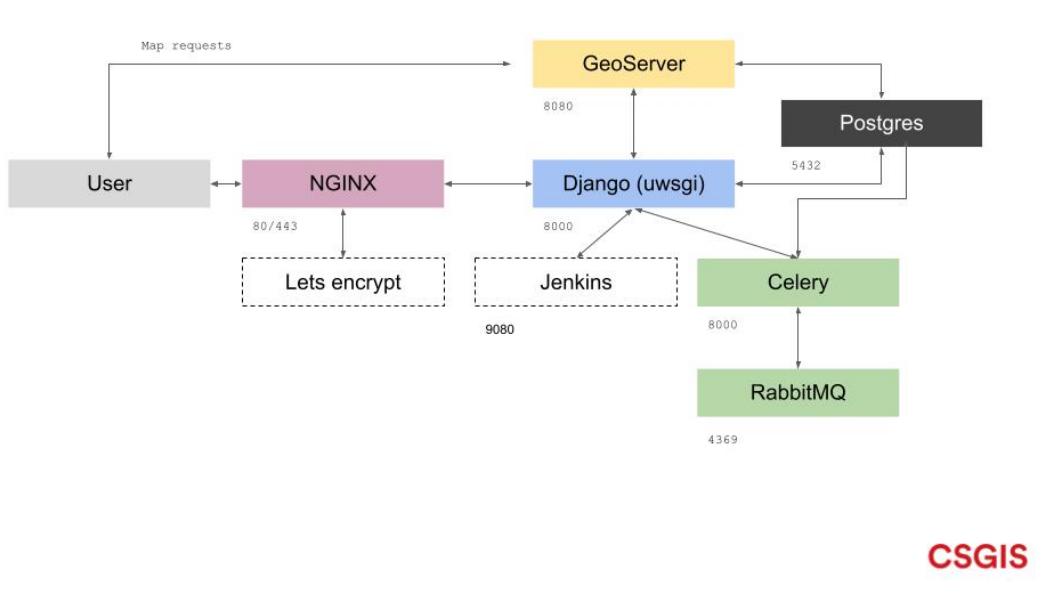
besitzen.

# Die Komponenten im Zusammenspiel.

Das was wir als GeoNode beschreiben ist eigentlich ein Zusammenspiel aus unterschiedlichen Open-Source Komponenten.

Die Aufgaben sind wie folgt:

- NGINX
  - Frontend Server, nimmt Anfragen entgegen und leitet dieses weiter (Proxy)
- Lets Encrypt
  - Stellt https Zertifikate bereit
- Django
  - Das Python Framework, dass sich größtenteils um die “Business Logik” des Systems kümmert. Also zum Beispiel die Datenhaltung, die API oder die URL Definitionen regelt.
- Jenkins
  - Erlaubt die Automatisierung von diversen Abläufen. Im Kontext von GeoNode wird, ist optional für die Erstellung von Backups verwendet
- GeoServer
  - Ein auf den OGC Standards basierter Kartenserver. Dieser verarbeitet/liefert letztendlich die Geo-Daten aus.
- Celery
  - Eine python Komponente um Abläufe im Hintergrund (asynchron) auszuführen. Wird zum Beispiel für das Erstellen von Thumbnails verwendet, um so die User Oberfläche nicht zu blockieren.
- RabbitMQ
  - Speichert die Aufgaben (Tasks) für Celery
- Postgres
  - Ein relationaler Datenbank Server der über die POSTGIS Extension leistungsstarke GIS Funktionalitäten bereitstellt.



Betrachten wir die laufenden Container einer Docker Umgebung, zeigen sich exakt diese Dienste als Container abstrahiert:

Docker ist eine Software zur Isolierung von Anwendungen mithilfe von Containervirtualisierung.

```
toni@localhost:~/geonode_training$ docker-compose ps
```

Name	Command	State	
celery4geonode_training	/usr/src/geonode_training/ ...	Up	8000/tcp
db4geonode_training	docker-entrypoint.sh postg ...	Up (healthy)	127.0.0.1:5432->5432
djang4geonode_training	/usr/src/geonode_training/ ...	Up (healthy)	8000/tcp
geoserver4geonode_training	/usr/local/tomcat/tmp/entr ...	Up (healthy)	0.0.0.0:8080->8080
gsconf4geonode_training	sleep infinity	Up (healthy)	
jenkins4geonode_training	/sbin/tini -- /usr/local/b ...	Up	0.0.0.0:50000->50000
letsencrypt4geonode_training	./docker-entrypoint.sh /bi ...	Up	
nginx4geonode_training	/docker-entrypoint.sh ngin ...	Up	0.0.0.0:443->443/tcp
rabbitmq4geonode_training	docker-entrypoint.sh rabbi ...	Up	25672/tcp, 4369/tcp

# Installationsarten

---

GeoNode kann auf unterschiedliche Weise in zwei verschiedenen Ausprägungen installiert werden.

## Ausprägungen

---

1. Es wird der “Geonode-Core” installiert
  - i. Wir haben den Quellcode bereits in vorausgehenden Kapiteln gesehen
2. Es wird “GeoNode Project” installiert
  - i. “GeoNode Project” ist ein “Wrapper” um den GeoNode Core der erlaubt, Dateien zu überschreiben ohne die Core Daten anzufassen. Auch wenn sich hierdurch die Komplexität erhöht, ist es die erste Wahl, sobald es um Anpassungen oder eigene Entwicklungen geht.
3. SPC-Geonode
  - i. taucht vereinzelt noch in der Dokumentation auf. Obsolete GeoNode Konfiguration.

## Installationswege

---

Für beide Ausprägungen sind alle Installationswege möglich.

1. *Docker (Empfohlener Installationspfad)*
  - i. Die aktuell zu empfehlende Umgebung. Die einzelnen Komponenten werden automatisiert korrekt in Container abstrahiert und konfiguriert.
2. *Manuelle Installation*
  - i. Bei diesem Installationsweg installiert der Systemadministrator alle Komponenten einzeln “von Hand”. Dies erfolgt auf einer Ubuntu oder Debian Distribution. (Vereinzelt sind auch Installationen auf Red Hat erfolgt). Aufgrund von fehlenden Abhängigkeiten ist eine Installation auf Windows Servern nicht zu installieren.
3. *Ansible*
  - i. Ist eine Automatisierungssoftware welche die manuelle Installation in sogenannten “Playbooks” automatisiert abläuft. Im GeoNode Project Verzeichnis befindet sich ein [Beispiel](#) Playbook. Status: Unklar.
4. *Vagrant*
  - i. Ist eine Automatisierung, um virtuelle Server zu konfigurieren. Eine Konfigurationsdatei findet sich [hier](#). Sie wird verwendet, um schnell eine virtuelle Testinstanz mit Docker zu erstellen.

## Weiterführende Links

---

- [docs.geonode.org – Installationsanleitungen](http://docs.geonode.org)

# Die Django Settings

Die vielleicht wesentlichste Datei bezüglich Einstellungen von GeoNode ist die Datei `settings.py`. Diese Einstellungsdatei ist allen Django Projekten genuin. Somit nichts GeoNode spezifisches.

Einen guten Einstiegspunkt liefert daher die [offizielle Django Dokumentation](#).

Im Haupt-Repository von GeoNode (core) befindet sich die Datei unter folgendem Pfad:

- <https://github.com/GeoNode/geonode/blob/master/geonode/settings.py>

In GeoNode-Projekt unter:

- [https://github.com/GeoNode/geonode-project/blob/master/src/project\\_name/settings.py](https://github.com/GeoNode/geonode-project/blob/master/src/project_name/settings.py)

Hier zeigt sich eine Besonderheit in Zeile 31ff.:

```
# Load more settings from a file called local_settings.py if it exists
try:
    from geonode_training.local_settings import *
#    from geonode.local_settings import *
except ImportError:
    from geonode.settings import *
```

Zu Beginn der `settings.py` wird versucht eine Datei namens `local_settings.py` zu laden. Sie soll erlauben GeoNode mit eigenen Einstellungen zu erweitern bzw. bestehende zu ergänzen.

Wie schon bei der Entwicklung angesprochen soll weiterhin tunlichst vermieden werden, die Quelldateien abzuändern. Ein weitaus besserer Weg ist die Anpassung über Umgebungsvariablen.

## Die .env Datei

Betrachtet man alle Einstellungen der `settings.py` fällt auf, dass für jede versucht wird zunächst die Einstellung über eine Umgebungsvariable zu setzen.

Beispiel

```
LANGUAGE_CODE = os.getenv('LANGUAGE_CODE', "en")
```

Dies ermöglicht uns, das komplette System über die Datei `.env` zu konfigurieren.

Ihre Variablen werden in Docker Umgebungen standardmäßig beim Starten als Umgebungsvariablen gesetzt.

## Beachtenswerte Einstellungsmöglichkeiten

Betrachten wir anhand der `.env` Datei einige wichtige Einstellungen, die Sie kennen sollten. Diese sind jene, die Sie bei nicht Verwendung des Scripts zur Installation setzen müssen.

- `COMPOSE_PROJECT_NAME={{project_name}}`
  - Wird für die Namen der Docker Container verwendet
- `DJANGO_SETTINGS_MODULE={{project_name}}.settings`
  - Definiert den Namen der zu ladenden Settings Datei
- `POSTGRES_PASSWORD={pgpwd}`
  - Das password für den Datenbank Super User
- `GEONODE_DATABASE={{project_name}}`

- Der Name der Django Datenbank
- GEONODE\_DATABASE\_PASSWORD={dbpwd}
  - Das Passwort des Django Datenbank Users
- GEONODE\_GEODATABASE={{project\_name}}\_data
  - Die Postgis Datenbank für Vector Daten
- GEONODE\_GEODATABASE\_PASSWORD={geodbpwd}
  - Das Password für die Geodatenbank
- DATABASE\_URL=postgis://{{project\_name}}:{dbpwd}@db:5432/{{project\_name}}
  - Die Verbindungsparameter zur Datenbank
- GEODATABASE\_URL=postgis://{{project\_name}}\_data:{geodbpwd}@db:5432/{{project\_name}}\_data
  - Die Verbindungsparameter zur Geodatenbank
- SITEURL={siteurl}/
  - Die IP oder Domain der Seite inklusive http(s). Wird in vielen Templates verwendet
- ALLOWED\_HOSTS="['django', '\*', '{hostname}']"
  - Von den hier definierten Domains darf das Portal aufgerufen werden. Der \* steht für erlaube alle
- GEONODE\_LB\_HOST\_IP={hostname}
  - Die öffentliche Domain. Wird unter Docker für das Networking benötigt.
- PUBLIC\_PORT={public\_port}
  - Der öffentliche Port
- HTTP\_HOST={http\_host}
  - Die öffentliche Domain
- HTTPS\_HOST={https\_host}
  - Die öffentliche https Domain, falls https verwendet wird
- GEOSERVER\_WEB\_UI\_LOCATION={{geoserver\_ui}}/geoserver/
  - Die öffentliche URL des GeoServers
- GEOSERVER\_PUBLIC\_LOCATION={{geoserver\_ui}}/geoserver/
  - Die öffentliche URL des GeoServers
- GEOSERVER\_ADMIN\_PASSWORD={geoserverpwd}
- ADMIN\_PASSWORD={geonodepwd}
  - Das Password der bei Installation erstellten Admin Rolle
- ADMIN\_EMAIL={email}
  - Die E-Mail des Admins
- DEFAULT\_FROM\_EMAIL='{email}'
  - Absender von System E-Mails
- OAUTH2\_CLIENT\_ID={clientid}
  - Die ID der Geoserver oauth2 app
- OAUTH2\_CLIENT\_SECRET={clientsecret}
  - Der Key der Geoserver oauth2 app
- DEBUG={debug}
  - Debug Modus an oder aus? (True oder False). In Production immer False!
- SECRET\_KEY="{secret\_key}"
  - Ein Schlüssel, den Django zur Verschlüsselung von beispielsweise Formulardaten verwendet.

Eine vollständige Übersicht aller Variablen finden Sie [hier](#).

Tipp: Lesen Sie die Datei settings.py. Hier finden sich einige hilfreiche Kommentare der Entwickler.

## Weiterführende Links

- [Geonode Docs - Übersicht der Variablen](#)



# Management Kommandos

Management Kommandos sind Hilfsfunktionen für GeoNode-Wartungsaufgaben. Sie werden normalerweise von einer SSH/Bash-Shell auf dem Server ausgeführt, auf dem GeoNode läuft.

Ein Beispielaufruf sieht wie folgt aus:

```
python manage.py migrate_baseurl --help
usage: manage.py migrate_baseurl [-h] [-f] [--source-address SOURCE_ADDRESS] [--target-address TARGET_ADDRESS]
                                  [--pythonpath PYTHONPATH] [--traceback] [--no-color] [--force-color]
                                  [--skip-checks]
Migrate GeoNode VM Base URL

optional arguments:
  -h, --help            show this help message and exit
  -f, --force           Forces the execution without asking for confirmation.
  --source-address SOURCE_ADDRESS
                        Source Address (the one currently on DB e.g. http://192.168.1.23)
  --target-address TARGET_ADDRESS
                        Target Address (the one to be changed e.g. http://my-public.geonode.org)
  --version             show program's version number and exit
  -v {0,1,2,3}, --verbosity {0,1,2,3}
                        Verbosity level; 0=minimal output, 1=normal output, 2=verbose output, 3=very verbose output
  --settings SETTINGS   The Python path to a settings module, e.g. "myproject.settings.main". If not specified, the DJANGO_SETTINGS_MODULE environment variable is used.
  --pythonpath PYTHONPATH
                        A directory to add to the Python path, e.g. "/home/djangoprojects/myproject"
  --traceback           Raise on CommandError exceptions
  --no-color            Don't colorize the command output.
  --force-color          Force colorization of the command output.
  --skip-checks         Skip system checks.
```

Ein angefügtes `--help` gibt uns also die Hilfe zum jeweiligen Kommando an.

Um sicher zu gehen dass Django mit den korrekten Settings arbeitet, wird dem Befehl `DJANGO_SETTINGS_MODULE={Projekt_name}.settings` vorangestellt. `Projekt_name` bezeichnet hierbei den Namen Ihres GeoNode Projekts den sie auch in den `settings.py` verwenden. In Docker Umgebungen kann hierauf verzichtet werden.

Tipp: Sehen Sie sich auch die Ausgabe von `python manage.py --help` an. Sie enthält weitere Befehle, die direkt durch Django bereit gestellt werden.

## Übersicht über GeoNode spezifische Kommandos

### migrate\_baseurl

Mit dem Verwaltungsbefehl `migrate_baseurl` können Sie alle GeoNode-Links korrigieren. Dies ist zum Beispiel nötig, wenn Sie den Domänennamen oder die IP-Adresse Ihres Portals ändern wollen.

Beispiel:

```
python manage.py migrate_baseurl --source-address=127.0.0.1 --target-address=example.org
```

## Berechtigungen, Metadaten, Legenden und Download-Links aktualisieren

## sync\_geonode\_datasets

Mit diesem Befehl können Sie bereits vorhandene Berechtigungen zwischen Django und dem im Hintergrund agierenden GeoServer synchronisieren. Dies ist zum Beispiel nötig wenn Datasets trotz der richtigen Berechtigungen nicht geladen werden können.

Beispiel: Ich möchte alle Berechtigungen und Attribute der Datasets mit dem GeoServer aktualisieren/synchronisieren

```
manage.py sync_geonode_datasets --updatepermissions --updateattributes
```

Beispiel 2: Ich möchte die Thumbnails aller Datensätze neu generieren, die dem User toni gehören

```
python manage.py sync_geonode_datasets -u toni --updatethumbnails
```

## sync\_geonode\_maps

Dieser Befehl ist im Grunde ähnlich wie der vorherige, wirkt sich aber mit einigen Einschränkungen auf die Karten aus.

Beispiel: Ich möchte das Thumbnail der Karte 'This is a test Map' neu generieren.

```
manage.py sync_geonode_maps --updatethumbnails -f 'This is a test Map'
```

## set\_all\_layers\_metadata

Dieser Befehl ermöglicht das Zurücksetzen der Metadatenattribute und des Katalogschemas von Datensätzen. Der Befehl aktualisiert auch die CSW Catalogue XML und Links von GeoNode.

Beispiel: Nachdem ich die Basis-URL geändert habe, möchte ich das gesamte Katalogschema neu generieren.

```
python manage.py set_all_layers_metadata -d
```

## set\_layers\_permissions

GeoNode bietet ein sehr nützliches management Kommando, mit dem ein Administrator auf einfache Weise Berechtigungen für Gruppen und Benutzer auf einem oder mehreren Datensätzen hinzufügen/entfernen kann.

Beispiel: Weisen Sie den Benutzern user\_A und user\_B und der Gruppe group\_C Schreibrechte für die Datasets layer\_X und Dataset Y zu.

```
manage.py set_layers-permissions -p write -u user_A user_B -g group_C -r 'layer_X' 'Dataset Y'
```

## Daten in GeoNode laden

Es gibt Situationen, in denen es nicht möglich oder nicht sinnvoll ist, das Upload-Formular zu verwenden, um neue Datensätze über die Weboberfläche zu GeoNode hinzuzufügen. Zum Beispiel:

- Der Datensatz ist zu groß, um ihn über die Weboberfläche hochzuladen.
- Daten sollen von einem Speichermedium programmatisch importiert werden.
- Tabellen aus einer Datenbank sollen publiziert werden.

## importlayers

Lädt Dateien aus einem lokalen Verzeichnis, einschließlich Unterordnern, in GeoNode. Die Datensätze werden zur Django-Datenbank, der GeoServer-Konfiguration und dem pycsw-Metadaten-Index hinzugefügt. Momentan werden nur Dateien vom Typ Esri Shapefile (.shp) und GeoTiff (.tif) unterstützt. Um den Import durchzuführen zu können, muss GeoNode in Betrieb sein.

Beispiel: Ordner mit shapes in GeoNode laden

```
manage.py importlayers /Users/tonischonbuchner/Desktop/langesgrenzen_shp
```

## updatelayers

Während es wie gezeigt möglich ist, Datensätze direkt aus dem Dateisystem Ihres Servers in Ihren GeoNode zu importieren, haben Sie vielleicht einen bestehenden GeoServer, der bereits Daten/Ebenen enthält. Um diese in GeoNode zu publizieren, steht das management Kommando `updatelayers` zur Verfügung.

Beispiel: Importiere eine bestehende Ebene mit dem Namen `_1_SARMIENTO_ENERO_2018` von GeoServer in Geonode.

```
manage.py updatelayers -w geonode -f _1_SARMIENTO_ENERO_2018
```

## delete\_resources

Der `delete_resources` Management-Befehl erlaubt es, Ressourcen zu entfernen, die eine bestimmte Bedingung erfüllen, die in Form eines “Django Q()”-Ausdrucks angegeben ist.

Beispiel: Lösche alle Ebenen die `Wasser` enthalten; dem User `admin` gehörten und dessen ID 1 oder 2 ist. Sowie alle hierzu gehörenden Karten.

```
manage.py delete_resources -l 'Q(pk__in: [1, 2]) | Q(title__icontains:"water"))' 'Q(owner__name=admin)
```

## Weiterführende links

---

- [Geonode Dokumentation - Management Kommandos](#)

# Die Django shell

Neben den Managementkommandos des vorherigen Kapitels besitzen Administratoren mit Shell Zugang ein weiteres mächtiges Werkzeug. Die “Django Shell”.

Wir starten Sie mit

```
python ./manage.py shell
```

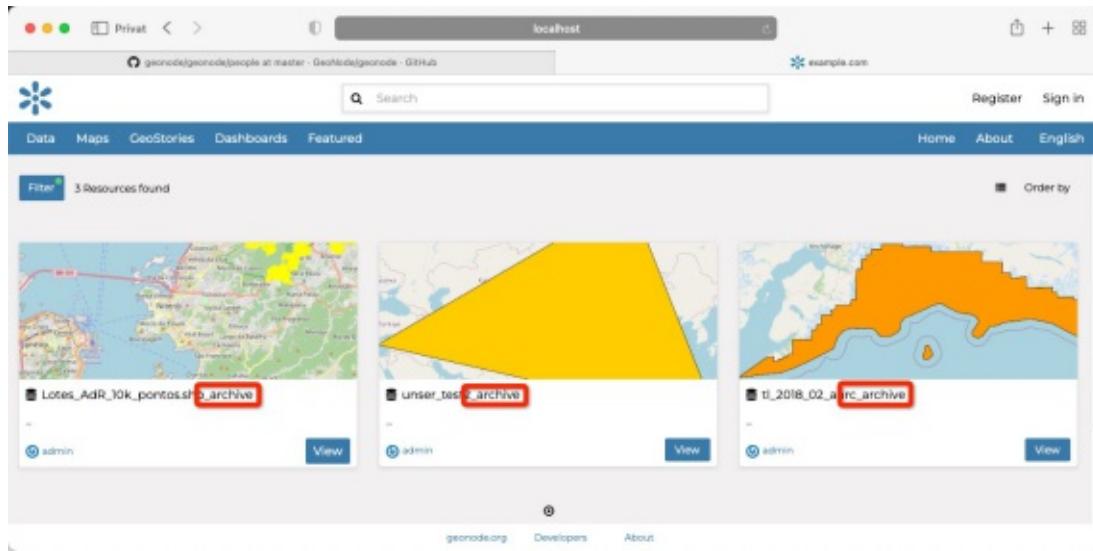
In dieser Python Sitzung haben wir Zugriff auf die GeoNode Datenbank und Ihre Datensätze.

## Beispiel: Ändern aller Titel bestehender Datensätze

```
$ ./manage_dev.sh shell

from geonode.base.models import ResourceBase
all_datasets = ResourceBase.objects.all()
for dataset in all_datasets:
    dataset.title += '_archive'
    dataset.save()
```

Betrachten wir hiernach alle Titel, sehen wir, dass die Datensätze bearbeitet wurden.



## Weiterführende Links

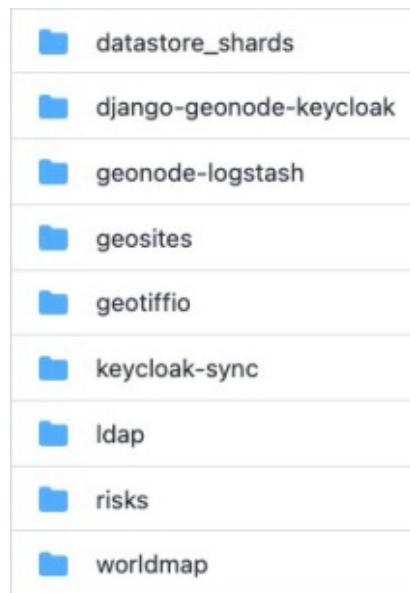
- [Django Dokumentation - manage.py](#)

# Contrib Apps

“Contrib Apps” sind Erweiterungen, die sie GeoNode hinzufügen können.

Diese finden Sie in folgendem Repository:

<https://github.com/GeoNode/geonode-contribs>



In Django Umgebungen sind die Contrib Apps bereits installiert. Auf zwei Erweiterungen soll hingewiesen werden.

## ldap Erweiterung

Erlaubt die Authentifizierung von Usern über LDAP. Die Konfiguration finden sie bereits in der bekannten `settings.py`

```
# LDAP
LDAP_ENABLED=False
LDAP_SERVER_URL=ldap://<the_ldap_server>
LDAP_BIND_DN=uid=ldapinfo,cn=users,dc=ad,dc=example,dc=org
LDAP_BIND_PASSWORD=<something_secret>
LDAP_USER_SEARCH_DN=dc=ad,dc=example,dc=org
LDAP_USER_SEARCH_FILTERSTR=(&(uid=%(user)s)(objectClass=person))
LDAP_GROUP_SEARCH_DN=cn=groups,dc=ad,dc=example,dc=org
LDAP_GROUP_SEARCH_FILTERSTR=(|(cn=abt1)(cn=abt2)(cn=abt3)(cn=abt4)(cn=abt5)(cn=abt6))
LDAP_GROUP_PROFILE_MEMBER_ATTR=uniqueMember
```

Siehe: <https://github.com/GeoNode/geonode-project/blob/master/.env.sample#L214-L223>

## keycloak-sync Erweiterung

Diese Contrib App erlaubt die Authentifizierung von Usern über einen eigenständigen [Keycloak Server](#). Keycloak als Authentifizierungsserver unterstützt wiederum eine Vielzahl an Authentifizierungsquellen.

Um das Modul zu konfigurieren müssen folgende Settings hinzugefügt werden.

```
if 'keycloaksync' not in INSTALLED_APPS:  
    INSTALLED_APPS += ('keycloaksync',)  
  
KEYCLOAK_URL=os.getenv('KEYCLOAK_URL', None)  
KEYCLOAK_CLIENT=os.getenv('KEYCLOAK_CLIENT', None)  
KEYCLOAK_CLIENT_ID=os.getenv('KEYCLOAK_CLIENT_ID', None)  
KEYCLOAK_CLIENT_SECRET=os.getenv('KEYCLOAK_CLIENT_SECRET', None)  
KEYCLOAK_REALM=os.getenv('KEYCLOAK_REALM', None)  
KEYCLOAK_USER=os.getenv('KEYCLOAK_USER', None)  
KEYCLOAK_PASSWORD=os.getenv('KEYCLOAK_PASSWORD', None)  
KEYCLOAK_USER_REALM=os.getenv('KEYCLOAK_USER_REALM', None)
```

Beachten Sie die [Readme](#) des Repository.

## Weiterführende Links

---

- [GeoNode Docs – Contrib Apps](#)
- [LDAP Contrib App](#)
- [Keycloak Contrib App](#)

# Die GeoNode Datenbanken

GeoNode arbeitet mit zwei Datenbanken:

1. geonode
2. geonode\_data

Wobei "geonode" durch den jeweiligen Projektnamen ersetzt wird

- geonode\_training
- geonode\_training\_data

In der Datenbank 1. geonode werden alle Tabellen der einzelnen Django Apps abgelegt.

The screenshot shows a tree view of a PostgreSQL database schema. The root node is 'geonode\_training'. Underneath it is a 'Schemata' node, which is also highlighted with a red border. Below 'Schemata' is the 'public' schema, indicated by a blue icon. Under 'public' is a 'Tabellen' node, indicated by an orange icon. A list of tables follows, each preceded by a blue table icon:  
account\_emailaddress  
account\_emailconfirmation  
actstream\_action  
actstream\_follow  
announcements\_announcement  
announcements\_dismissal  
auth\_group  
auth\_group\_permissions  
auth\_permission  
avatar\_avatar  
base\_configuration  
base\_contactrole  
base\_extrametadata  
base\_groupgeolimit  
base\_hierarchicalkeyword  
base\_license  
base\_link  
base\_menu

In der Datenbank 2. geonode\_data werden alle Tabellen von Vektor Datensätzen abgelegt. Auf diese Datenbank greift der GeoServer ebenfalls zu.

The screenshot shows a tree view of a PostgreSQL database schema. The root node is 'geonode\_training\_data'. Underneath it is a 'Schemata' node, which is highlighted with a red border. Below 'Schemata' is the 'public' schema, indicated by a blue icon. Under 'public' is a 'Tabellen' node, indicated by an orange icon. A list of tables follows, each preceded by a blue table icon and a size value in parentheses:  
administrative 184K  
education 184K  
gf\_adminrule 40K  
gf\_gfuser 32K  
gf\_gsinstance 16K  
gf\_gsuser 40K  
gf\_layer\_attributes 16K  
gf\_layer\_details 24K  
gf\_layer\_styles 24K  
gf\_rule 216K  
gf\_rule\_limits 24K  
gf\_user\_usergroups 8K  
gf\_usergroup 32K  
kartierung 40K  
landuse 168K  
landuse0 168K  
landuse3 144K

Achtung: Beide Datenbanken benötigen die POSTGIS Erweiterung!

# Datenbank Verbindung

Diese Anleitung zeigt den empfohlenen Weg, sich mit den Datenbanken von GeoNode in einer Docker Umgebung zu finden.

Standardmäßig ist von außen keine Verbindung zu der im Postres-Container laufenden Datenbankserver möglich. Wir ändern dies, indem wir in unserer `docker-compose.yml` die Ports verlinken.

```
db:
  # use geonode official postgis 13 image
  image: geonode/postgis:13
  command: postgres -c "max_connections=${POSTGRESQL_MAX_CONNECTIONS}"
  container_name: db4${COMPOSE_PROJECT_NAME}
  env_file:
    - .env
  volumes:
    - dbdata:/var/lib/postgresql/data
    - dbbackups:/pg_backups
  restart: on-failure
  healthcheck:
    test: "pg_isready -d postgres -U postgres"
  # Hier die Verlinkung zwischen den Ports zwischen Host und Container
  ports:
    - "127.0.0.1:5432:5432"
```

Achtung: wir verwenden auf Host Seite nicht 5432 , sondern 127.0.0.1:5432 . Bei Verwendung von 5432 würden wir den Port komplett nach außen hin öffnen, was wir nicht wollen.

Um uns nun mit den Datenbanken mit einem lokalen Datenbank-Klienten wie “Dbeaver” oder “pgadmin” verbinden zu können, müssen wir einen sogenannten SSH Tunnel herstellen.

```
ssh toni@geonode-training.csgis.de -L 5490:127.0.0.1:5432
```

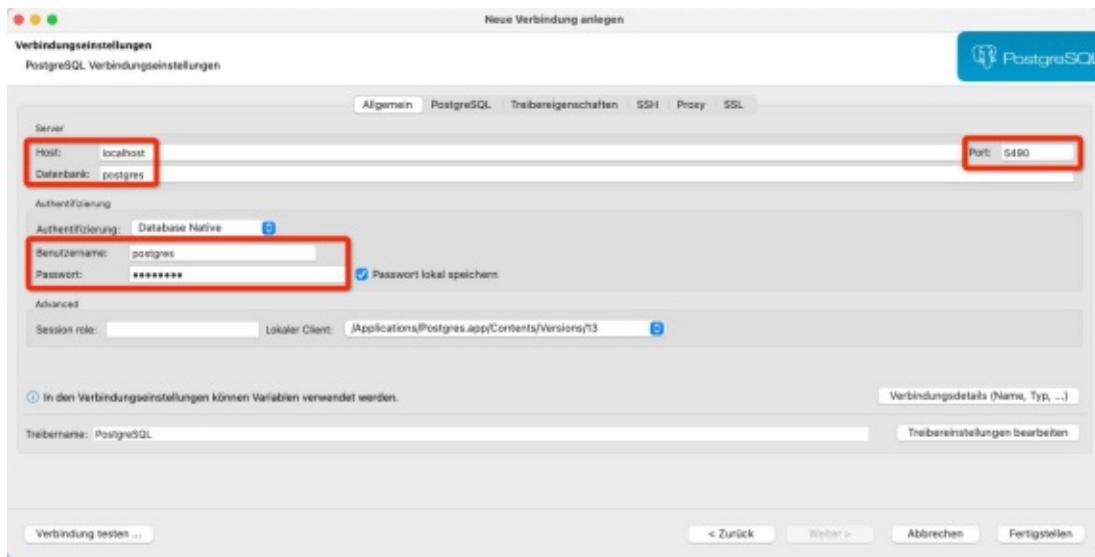
Über die -L Angabe binden wir den Port 5490 an den Port 5432 des Servers. (In diesem Fall geonode-training.csgis.de)

Eine letzte Überprüfung der Docker Container stellt sicher, dass der Port für den Host sichtbar ist:

```
docker ps
```

```
toni@localhost:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
6bac79a254ca geonode_training_django:4.0 "/usr/src/geonode_tr..." 6 days ago Up 6 days (healthy) 8080/tcp
891f3fc135a geonode_training_django:4.0 "/usr/src/geonode_tr..." 7 days ago Up 7 days 8080/tcp
e4200e048c28 geonode/geoserver:2.20.5 "/usr/local/tomcat/t..." 7 days ago Up 7 days (healthy) 0.0.0.0:8080->8080/tcp, 0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp
4c71c2b4fc23 geonode/nginx:4.0 "/docke...r-entrypoint..." 7 days ago Up 7 days 0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp
cf3031ec07a geonode/letsencrypt:4.0 "/docke...r-entrypoint..." 7 days ago Up 7 days
58e0a5209c49 geonode/geoserver_date:2.20.5 "sleep infinity" 7 days ago Up 7 days (healthy)
1bd229624048 jenkins/jenkins:slts "/sbin/tini -- /usr/..." 7 days ago Up 7 days 0.0.0.0:9080->9080/tcp, 0.0.0.0:9443->9443/tcp,
eac5b8df4b3 rabbitmq:3.7-alpine "docke...r-entrypoint.s..." 7 days ago Up 7 days 127.0.0.1:25672/tcp
6a8f1dd77ce6 geonode/postgis:13 "docke...r-entrypoint.s..." 7 days ago Up 7 days (healthy) 127.0.0.1:5432->5432/tcp
```

Hiernach können wir uns mit unserem lokalen Datenbank-Klienten über den Tunnel verbinden:



- Wir verwenden den Tunnelport 5490
- Als Host wählen wir localhost
- Als Datenbank die gewünschte Datenbank: {projectname}\_geonode oder {projectname}\_geonode
  - Da wir hier den User postgres verwenden, belasse ich in diesem Beispiel die Datenbank auf postgres
- Als Benutzername und Passwort haben wir drei Möglichkeiten. Je nachdem mit, wem wir uns verbinden, müssen wir ebenso die gewählte Datenbank-Definition verändern.
  - Den User der {projectname}\_geonode Datenbank
  - Den User der {projectname}\_geonode
  - Den Postgres User

Hiernach können wir uns mit den Datenbanken verbinden / mit diesen arbeiten.



# Backups

GeoNode besitzt ein Management Kommando, um alle erforderlichen Daten einer Instanz zu sichern sowie diese wieder einzuspielen. Der Prozess lässt sich weiterhin wie in der [Dokumentation](#) über Jenkins automatisieren.

Wir raten von der Verwendung der Funktion ab, da sie in der Vergangenheit in der User Community zu Problemen geführt hat.

## Backup Script

Eine alternative zur in GeoNode integrierten Backup-Funktionalität, stellt die Verwendung eines eigenen Backups Scripts dar. Dieses kann entweder über einen Cronjob zu festgelegten Zeiten ausgeführt werden. Oder in Docker Umgebungen über einen eigenen Docker Service automatisiert werden.

Ein sehr einfaches Backups Script könnte wie folgt aussehen:

backup.sh

```
#!/bin/sh

NOW=$( date '+%d-%m-%y' )

# Create directories
BPTH=/backups/backup_${NOW}
mkdir -p ${BPTH}

DPTH=${BPTH}/databases
mkdir -p ${DPTH}

SPTH=${BPTH}/statics
mkdir -p ${SPTH}

GPTH=${BPTH}/geoserver-data-dir
mkdir -p ${GPTH}

# Datenbanken sichern
pg_dump -h db -U geonode -C -d geonode_training > ${DPTH}/${DATABASE}_daily.pgdump && echo "${DATABASE} backed up"
pg_dump -h db -U geonode -C -d geonode_training > ${DPTH}/${DATABASE_GEO}_daily.pgdump && echo "${DATABASE_GEO} backed up"

# Backup der statischen Dateien und der GeoServer Dateien
rclone copy /geonode_statics ${SPTH} --log-level ERROR && echo "geonode_statics copied successfully"
rclone copy /geoserver-data-dir ${GPTH} --log-level ERROR && echo "geoserver-data-dir copied successfully"

# Daten in Archiv packen
tar cvfj /backups/bba-geonode.tar_${NOW}.bz2 ${BPTH} && rm -R ${BPTH}

# Alte Backups löschen
find /backups -maxdepth 1 -mtime +${DAYS_TO_KEEP} -exec rm -rf {} \; && echo "Clean of /backups completed"
```

Achten Sie auf die richtige Setzung der Pfade.

Die Daten, die gesichert werden sollen:

- Die Datenbanken
- Das Geoserver Daten Verzeichnis
- Die statischen Dateien von Django, die Thumbnails und hochgeladene Dokumente beinhalten

# Backup als Docker Service

---

Einen Beispiel Service dieses Scripts finden sie[hier](#).

## Weiterführende Links

---

- [GeoNode Docs - Backup und Restore](#)
- [Beispiel backup Service über Docker](#)

# Erste Hilfe bei Problemen

Die erste Handlung, die Sie bei Problemen unternehmen, ist die Betrachtung der Log-Files.

In einer Docker Umgebung setzen wir hierfür das Log-Level von GeoServer auf ein höheres als “production”.

The screenshot shows the GeoServer Admin interface. On the left, there's a sidebar with links like Global, JAI, Raster, Importer, Kartenkacheln-Cache, Sicherheit, and Daten. The Global link has a red arrow pointing to it. On the right, the 'Internal Settings' section is open, specifically the 'Logging Settings' tab. It shows a dropdown menu for 'Profil für die Protokollierung' with several options: DEFAULT\_LOGGING.properties, GEOSERVER\_DEVELOPER\_LOGGING.properties, GEOTOOLS\_DEVELOPER\_LOGGING.properties, PRODUCTION\_LOGGING.properties (which is highlighted and has a red arrow pointing to it), QUIET\_LOGGING.properties, TEST\_LOGGING.properties, and VERBOSE\_LOGGING.properties.

Und betrachten hiernach die Log-Ausgabe der Container

```
docker-compose logs -f
```

Um nur die Logs von einem Service zu bekommen, vermerken wir diesen wie folgt:

```
docker-compose logs -f django
```

## Die häufigsten Fehler

### Upload schlägt fehl

Kontrollieren Sie das Admin-Passwort des Geoservers in `.env` und setzen Sie dieses in der GeoServer Admin GUI erneut.

Ein weiterer Punkt ist dass eine Einstellung die Upload Größe limitiert. [Übersicht über Limits](#)

### Datensätze werden nicht angezeigt, obwohl die Berechtigungen stimmen

GeoServer regelt die Regeln über “Geofence” Regeln. Um diese mit Django/GeoNode abzulegen, verbindet sich GeoServer über oauth2 mit Django. Prüfen Sie die oauth2 Einstellungen der “geoserver app” im Django Admin (Abschnitt oauth2). Sowie die oauth2 Einstellungen auf Seiten von Geoserver.

- Siehe: <https://docs.geonode.org/en/master/advanced/components/index.html?highlight=oauth2>

Weitere kurze Problembeschreibungen finden Sie in folgendem Wiki:

- <https://github.com/GeoNode/geonode/wiki/Good-to-know>

## Orte um Hilfe zu finden

GeoNode ist ein Open-Source-Projekt, deren Mitwirkende in unterschiedlichen Foren versammeln:

- Mailingliste für Benutzer: <https://lists.osgeo.org/cgi-bin/mailman/listinfo/geonode-users>
- Mailing-Liste für Entwickler: <https://lists.osgeo.org/cgi-bin/mailman/listinfo/geonode-devel>
- Gitter Chat: <https://gitter.im/GeoNode/general>

Wir empfehlen den Gitter-Chat, um nach Hilfe zu fragen. Erfahrungsgemäß können Fragen hier am schnellsten beantwortet werden.

Der [Issue-Tracker auf Github](#) sollte nur für Bugs oder “Feature Requests” verwendet werden

## Weiterführende links

- [GeoNode Docs - oauth2](#)
- [Cithub Wiki – Good to know](#)