

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 mit konkreten Beispielen 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 in den GeoNode Plattform. Ohne Registrierung können die Daten (Karten, Datensätze, Dokumente, etc.) visualisiert und geteilt aber nicht bearbeitet werden. Nachdem sich einen Anwender erfolgreich registriert hat, kann sich gleich anmelden GeoNode im vollen Umfang benutzen.

Anmeldung

Erstellen Sie ein neues lokales Konto

E-Mail

Benutzer

Kennwort

Passwort (wiederholen)

Anmeldung

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

User Profil

Im Profil stehen den user u.a. folgende Funktionen zur Verfügung:

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

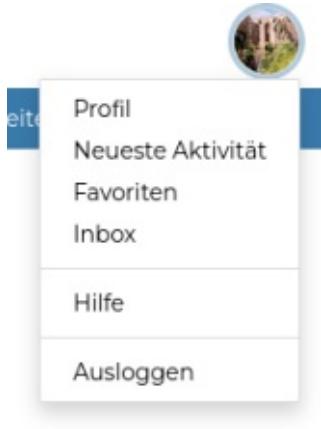
Es lohnt sich diese Optionen in Ruhe zum lesen und hier entscheiden mit welchen Aktionen eine Email an den Anwender gesendet werden soll (standardmäßig sind alle Optionen ausgewählt. Mit einer GeoNode Instance, die von vielen Anwender aktiv benutzt wird, werden Sie dann viele Emails bekommen!)

Benachrichtigungseinstellungen

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

Außerdem können registrierte Benutzer:

- Die letzten Aktivitäten von allen Benutzer sehen und zugreifen (Ebenen, Karten, Dokumenten, Geostories und Dashboards)
- eine Liste mit aller Favoriten Inhalten sehen
- Mailbox abfragen
- Das Hilfe Dokument (mit verlinkten Inhalten) lesen



Weitere empfehlenswert GeoNode Dokumentationen sind:

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

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

Benutzer entdecken

SUCHEN

Total: 3

Suche nach Name 

 joseca Keine Organisation	 jose Keine Organisation	 joseca2 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

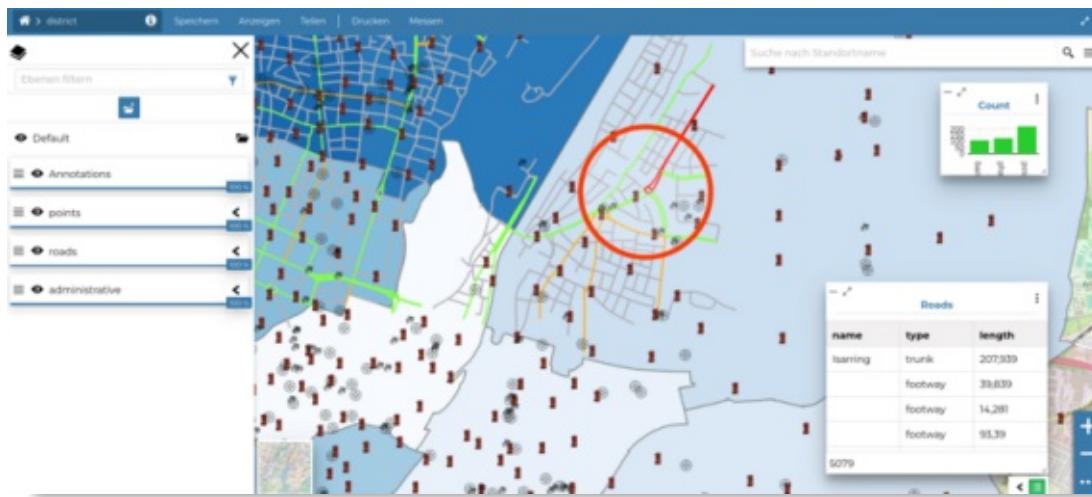
Übung

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

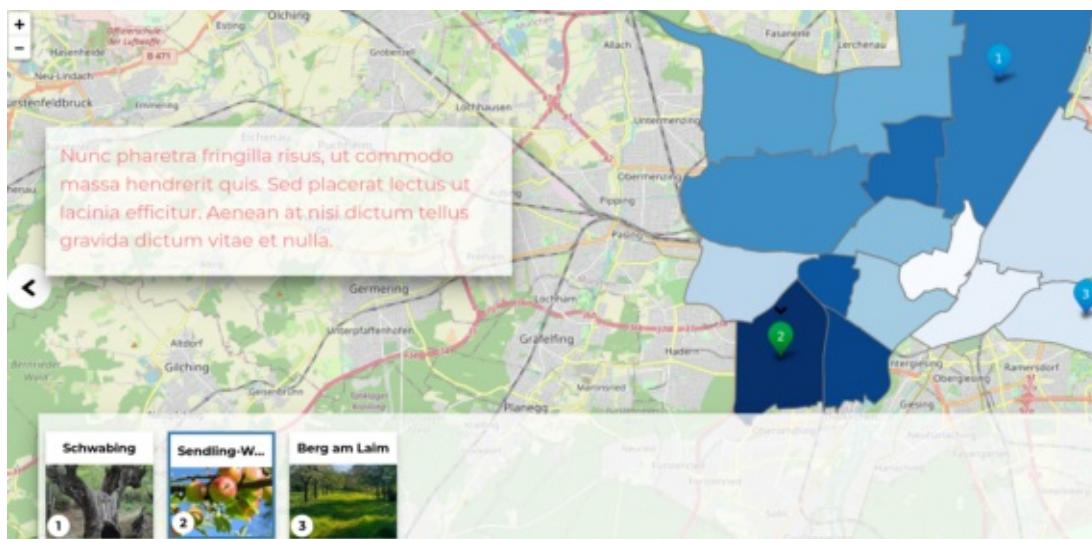
Publikation von Ressourcen

Folgende Ressourcen können in GeoNode publiziert und bearbeitet werden:

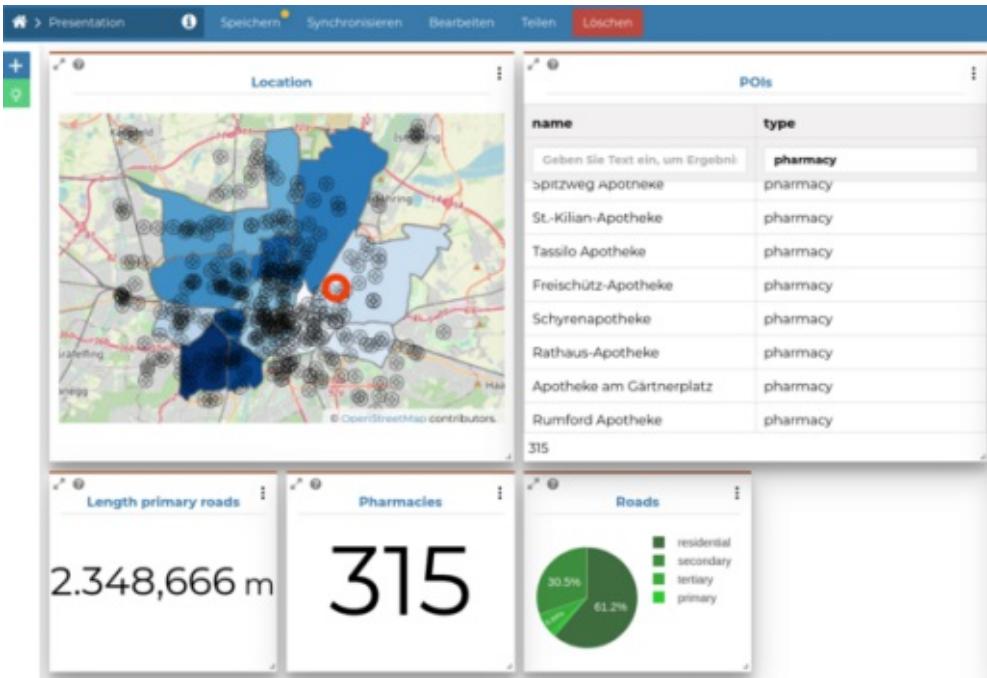
- DATENSÄTZE (Vektor, Raster, Remote oder Zeit Serien). Folgende Dateitypen können hochgeladen werden: ESRI Shapefile, GeoTIFF, Comma Separated Value (CSV), Zip Archive, XML Metadata File, Styled Layer Descriptor (SLD)
- DOKUMENTE (Bilder, Textdateien, Videos, PDF Dokumenten, Tabellen, etc). Diese Dateitypen sind unterstützt: .txt, .log, .doc, .docx, .ods, .odt, .sld, .qml, .xls, .xlsx, .xml, .bm, .bmp, .dwg, .dxf, .fif, .gif, .jpg, .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 beliebigen Datensätze hinzufügen und eigenständig eine Karte mit den verschiedenen Ebenen erstellen.



- GEOSTORIES sind online Berichte die man mit den GeoNode hochgeladenen oder mit externen Ressourcen (videos, webseiten, etc.) erstellen kann



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



- Alle Ressource in GeoNode sind standardmäßig öffentlich. Der Benutzer, der den GeoNode Ressource erstellt hat, kann entscheiden ob diese Daten für alle oder nur für registrierte Benutzer sichtbar sein sollen
- Nur die Person, die den GeoNode Ressource gehört (i.d.R. die Person, die den 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 'Eigentümer' (Owner) settings for a resource. It includes:

- Eigentümer:** Set to 'Jeder' (Everyone).
- Registrierte Mitglieder:** A dropdown menu with options for 'Herunterladen' (Download) and 'Bearbeiten' (Edit).
- + Benutzer / Gruppen hinzufügen:** A button to add users or groups.
- Nach Namen oder Berechtigungen filtern:** A search/filter input field.
- Name:** A table showing existing users: 'joseca' (with edit and delete icons) and 'joseca2' (with edit icon).
- Berechtigungen:** A dropdown menu for each user entry.

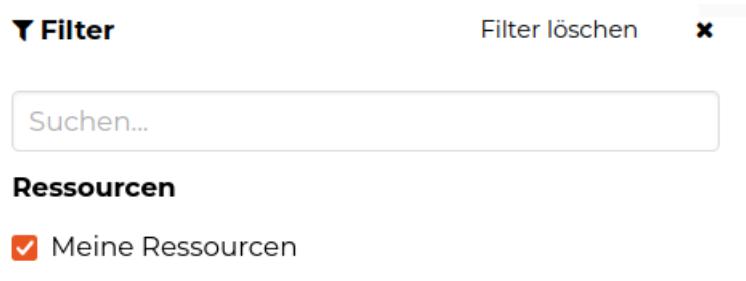
- Der Besitzer des GeoNode Ressources kann entscheiden, ob einen anderen Benutzer die Daten gehören sollen (siehe optionale Metadaten):

The screenshot shows optional metadata fields:

- Verantwortliche Person:** A dropdown menu set to 'joseca'.
- Point of Contact:** A dropdown menu set to 'joseca'.
- Verantwortliche und Berechtigungen:** A dropdown menu set to 'joseca'.
- Owner:** A dropdown menu set to 'joseca'.
- Metadata Author:** A dropdown menu set to 'joseca'.

Übung

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



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, Symbolologie, Berechtigungen,etc. vorgenommen werden, können Sie diese hier dauerhaft speichern. Mit "speichern als" erzeugen Sie einen neuen Dataset.

Im Menü bearbeiten können wir:

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

places
eine dataset von joseca2 / September 16th 2022

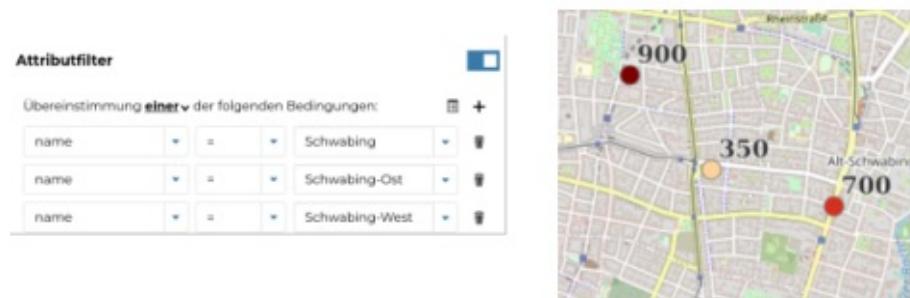
Daten bearbeiten

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

Übung:

Editierungen im Dataset places

1. Starten Sie die Editierung. Wählen Sie Schwabing aus und zoomen Sie zu den 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 Orten.



1. Untersuchen Sie weitere Filter Optionen wie "Interessengeiet"
2. Erzeugen Sie einen Punkt und vergeben Sie die Attribute in die Tabelle
3. Verschieben Sie einen Punkt in die Karte
4. Löschen Sie einen beliebigen Punkt

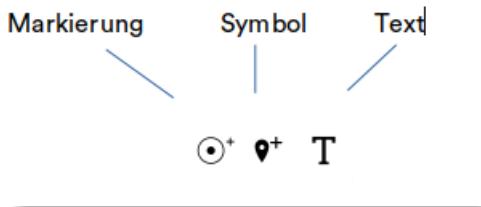
Editierungen im Dataset natural

1. Fügen Sie einen neuen Polygon hinzu
2. Digitalisieren Sie einen neuen benachbarten Polygon (Snap Optionen)

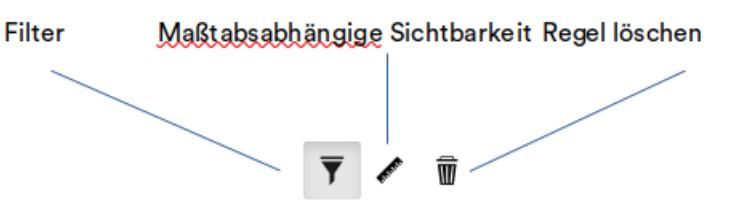
Zum selektieren in die Karte nur ein Mal klicken Selektieren von mehreren Objekten können wir über Filter machen

Stil bearbeiten

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

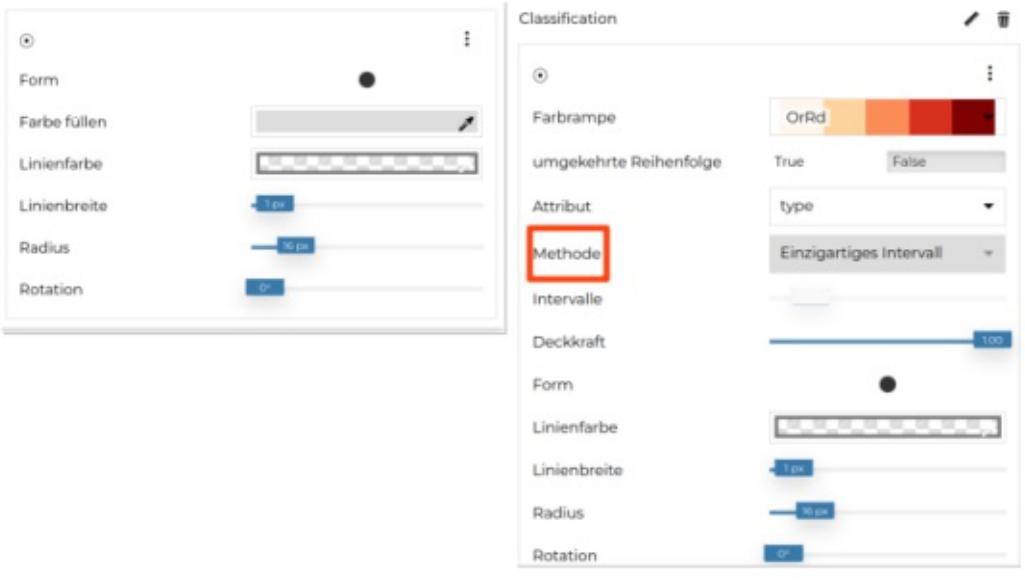


Außerdem gibt es die Möglichkeit für die verschiedene Regel die Daten zu filtern (wenn man zum Beispiel nur bestimmten Daten darstellen möchten – type=forest -) oder nur innerhalb von einem bestimmten Maßstab die Daten darstellen zu lassen.



Innerhalb der Markierung Option haben wir 2 verschiedene Legenden zur Verfügung:

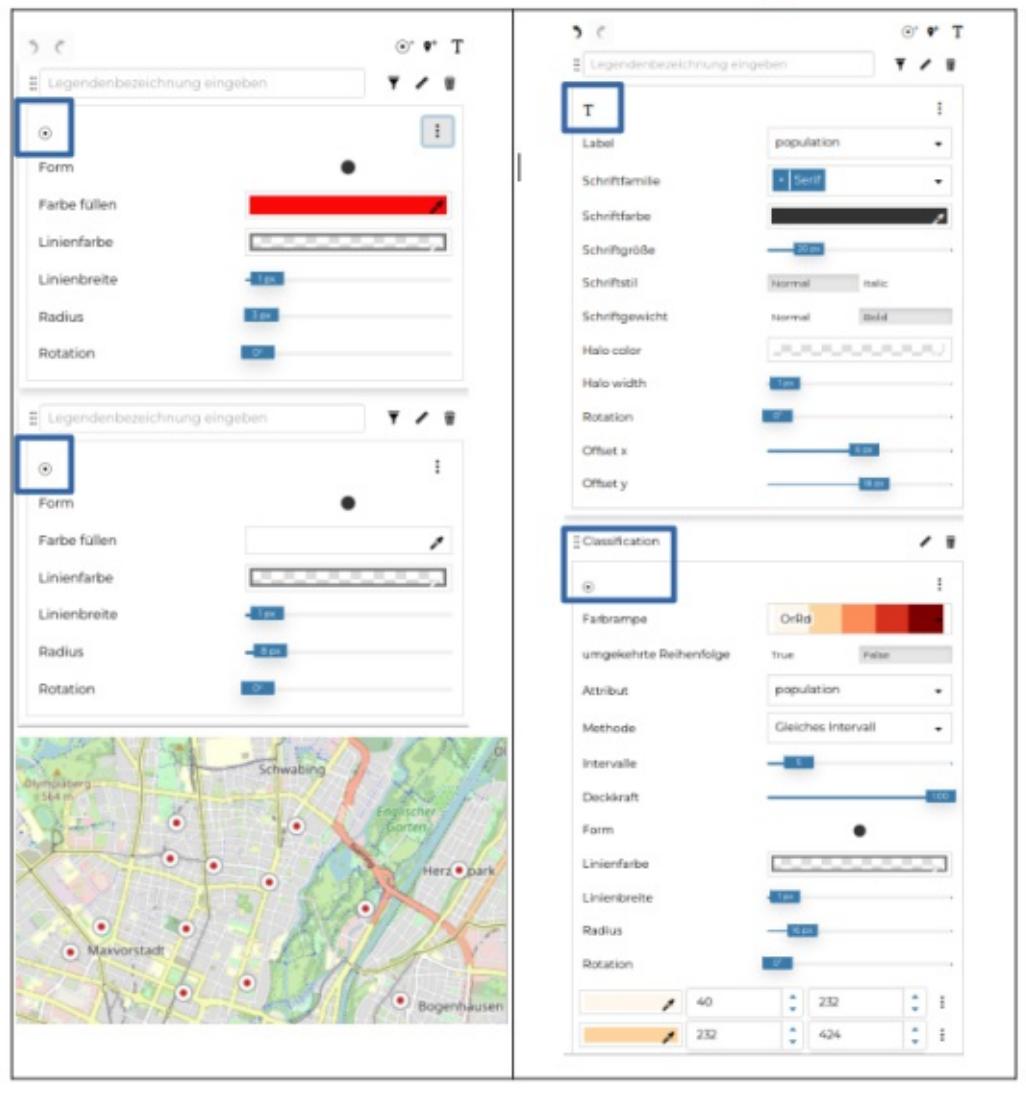
- Einfacher Stil
- Klassifizierungsstil (mit verschiedenen Methoden)



Übung

Symbologie von places

1. Erstellen Sie diese zwei verschiedene Legende für den Ressource *places* und speichern Sie die Ergebnisse (mit



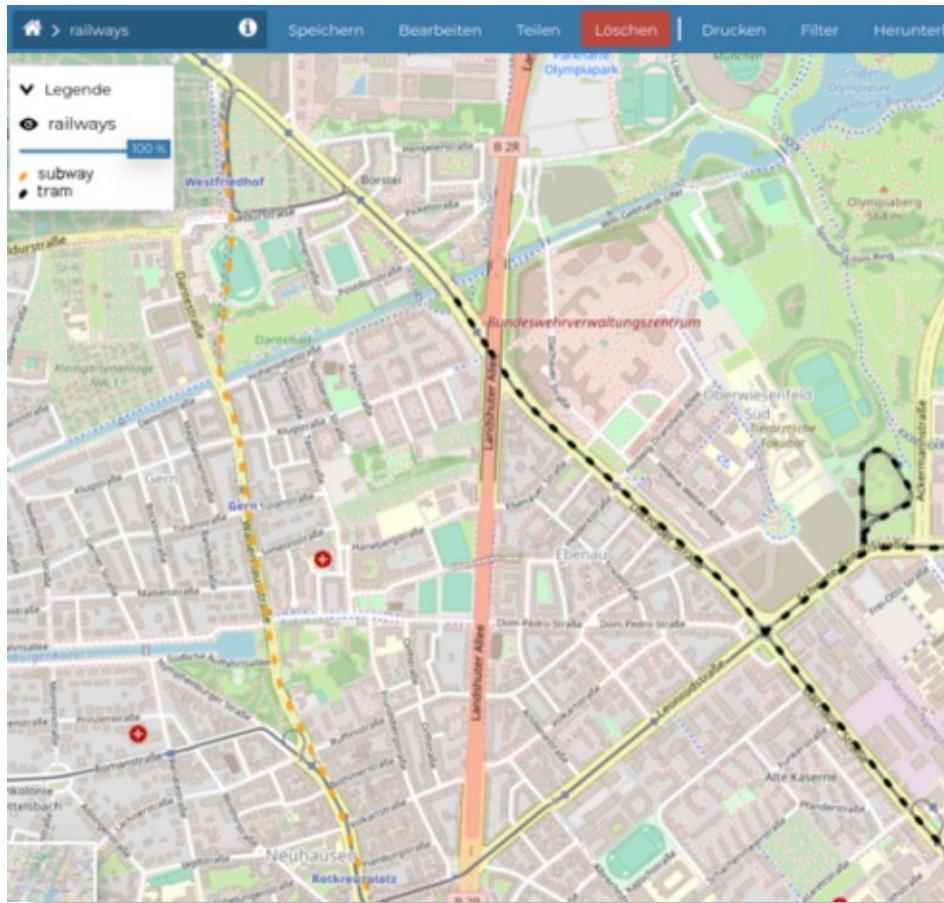
der Änderung .sld):

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

Übung

Symbolologie von railways

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



2. Speichern Sie den Stil als sld und Laden Sie diese Datei als Dokument in GeoNode hoch
3. Benachrichtigen Sie Ihren Kollegen dass es eine neue Legende gibt und schauen Sie Ihren Mailbox die gesendeten/empfangenen Nachrichten an.

Übung

Symbologie von roads

1. Lassen Sie roads klassifiziert darstellen nur mit folgenden Kategorien der Spalte "Type": primary, residential, secundary, tertiary. Roads soll nur ercheinen innerhalb diesen Maßstäbe 1:144448 und 1:9028
2. Wie kann man erreichen dass nur den type residential innerhalb von diesen Maßstäbe erscheint aber dass die andere Typen immer sichtbar bleiben?

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

Metadaten bearbeiten

Die Metadaten in GeoNode haben 2 wesentlichen Zwecken:

- sie geben mehr Auskunft über die Daten
- sie vereinfachen die Suche von Resourcen durch andere Benutzer und damit können die Daten leichter abgerufen werden.

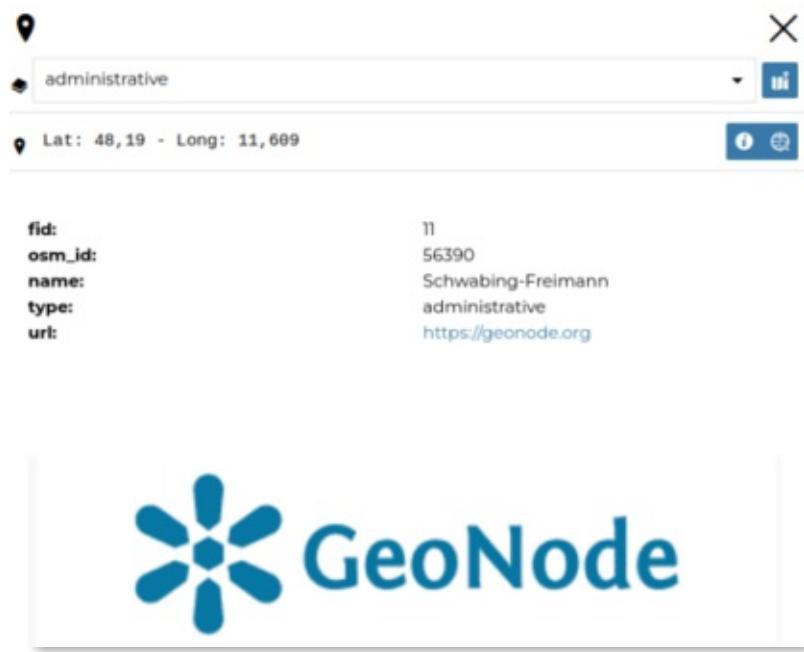
Übung

Metadaten von administrative

1. Thumbnail aktualisieren
2. Kategorie population speichern unter die Grundlegende Metadaten
3. Unter regions soll Germany gespeichert werden
4. Speichern Sie die Metadaten und suchen sie auf der Startseite in GeoNode nach Daten mit zum Beispiel die Kategorie population oder nach der Region Germany
5. Probieren Sie die Suche nach Daten mit dem Filter Option "Ausmaß"

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

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"/>



Ressource teilen

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

verwalten, jeder kann sie ansehen.

Weitere führenden 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 Attributen und Geometrien und des Stils).
- Verwalten (ermöglicht das Bearbeiten, Löschen, Ändern der Freigabeoptionen und Publizieren eines Datensatzes). Datensätze die nicht publiziert sind können nur von Admin gesehen und bearbeitet werden.

Übung

Berechtigungen von places

1. Erstellen Sie 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 Benutzer verwaltet werden können

The screenshot shows the 'Dataset permissions' interface. In the 'Eigentümer:' section, 'joseca2' is listed with 'Ansehen' (View) permission. Under 'Registrierte Mitglieder', 'joseca' is listed with 'Verwalten' (Manage) permission. There is a search bar labeled 'Nach Namen oder Berechtigungen filtern' and a 'Filter' button.

Filtern

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

- Attribut
- Interssengebiet

Beide Optionen können auch gleichzeitig benutzt werden:

Attributfilter

Übereinstimmung **einer** der folgenden Bedingungen:

population >= 600

Interessensgebiet

Filtertyp: Polygon
Geometrische Operation: Schneldet

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 ursprüngliche Projektion (prj Datei die beim Import der Daten benutzt wurde) herunterladen worden oder in WGS84 (EPSG 4326).

Außerdem können Sie definieren ob sie nur die gefilterte Daten oder die Daten des aktuellen Kartenfenster herunterladen möchten.

Daten exportieren

Dateiformat: Shapefile

Räumliches Bezugssystem: Native

Zuschneiden des Datensatzes in das aktuelle Ansichtsfenster

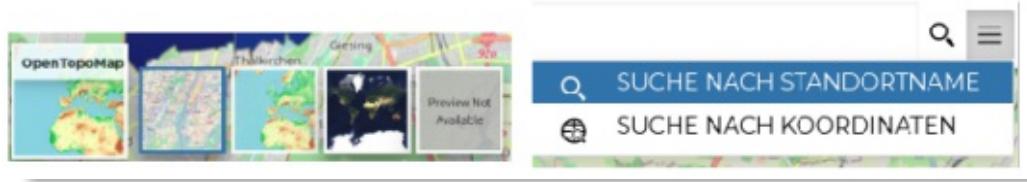
Laden Sie den gefilterten Datensatz herunter

Export

Karten erstellen

In einer Karte können mehreren datasets hinzugefügt werden. Karten sind eigene GeoNode Ressourcen die man auch publizieren oder mit anderen Benutzer bearbeiten/teilen kann.

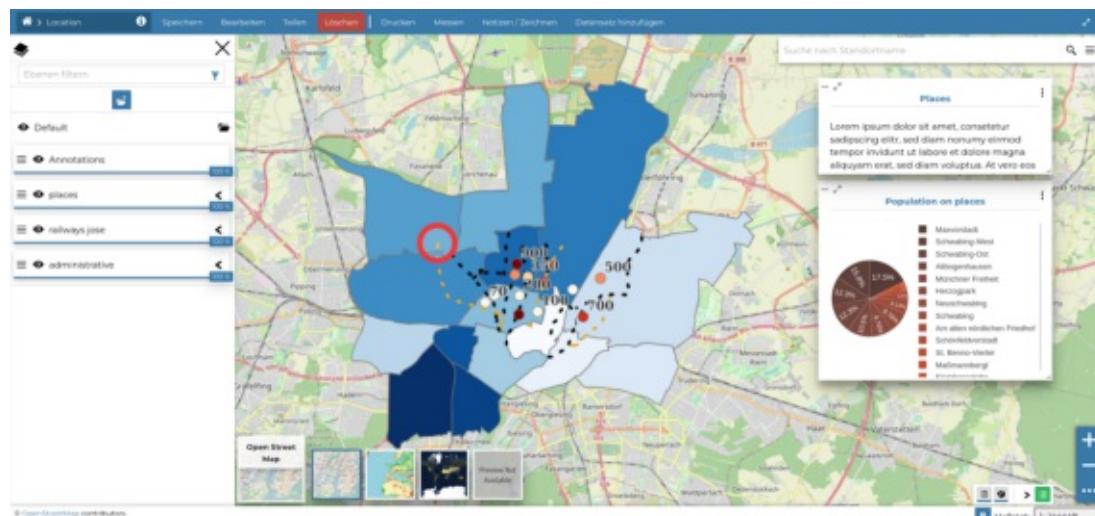
In die Karte stehen auch mehreren Hintergrundskarten oder eine Adresssuche zur Verfügung.



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

Übung

Erstellung folgende Karte

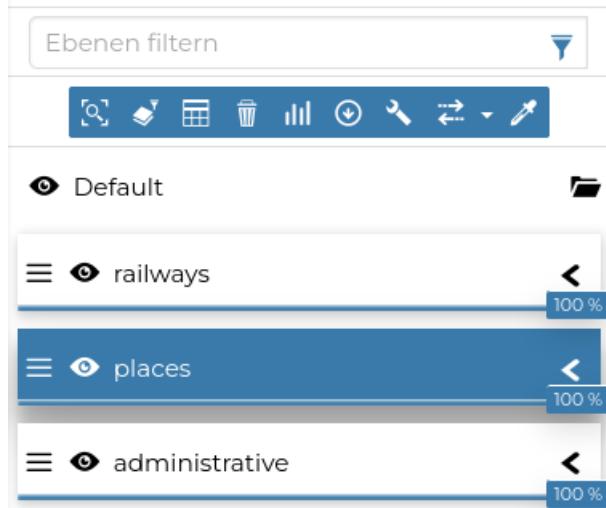


1. Ressource hinzufügen → Karte erstellen → Speichern
2. Datensatz hinzufügen → Im Datensatzkatalog die Ressource administrative, railways und places auswählen.

Two screenshots of the GeoNode dataset catalog. The left screenshot shows a list of datasets: 'points' (represented by a scatter plot icon) and 'administrative' (represented by a map icon). The right screenshot shows the 'Ebenen filtern' (Layers filter) interface, which lists three layers: 'Default', 'railways', 'places', and 'administrative'. Each layer has a slider bar set to 100%. The 'Default' layer is currently selected.

3. Mit klick auf einer Ebenen erscheint eine Reihe von interessanten Funktionalitäten. Wir können hier zum Beispiel:

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



1. 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

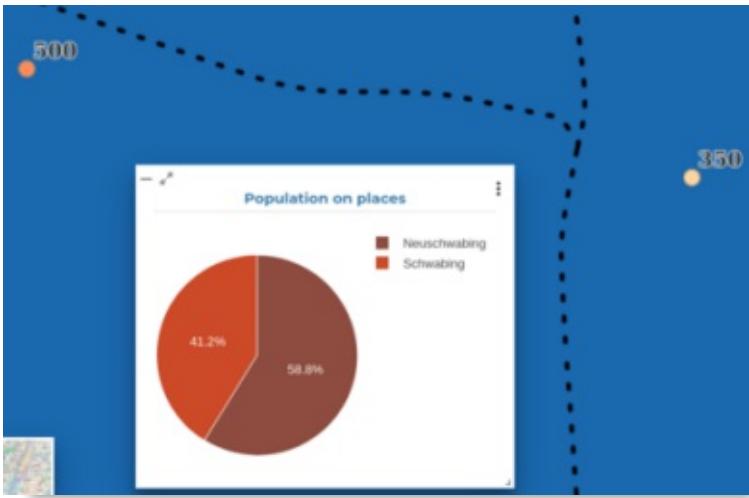
An den Stilen vorgenommene Änderungen gelten nur für den aktuellen Layer in Ihrer Karte und nicht für den ursprünglichen Datensatz.

zu verändern

1. Prüfen Sie dass die Symbologie der GeoNode Ressource *places* sich nicht geändert hat.
2. Begleiten Sie die Karte mit folgenden Widgets des Layers *places*:



1. Beachten Sie dass die Diagramme zeigen immer die Daten die man in den aktuellen Auschnitt der Karte sieht



1. Fügen Sie die Tabelle der Ebene Administrative in die Karte ein
2. Schalten Sie alle Widgets ein und aus und lassen Sie nur den Text und das Diagramm sichtbar
3. Lassen Sie die Ebene administrativ transparent darstellen

administrative X

> Allgemeine Einstellungen

▼ Sichtbarkeitseinstellungen

Deckkraft % X

Sichtbarkeitsgrenzen

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

GeoNode gibt die Informationen von alle Ebenen aus, wo man geklickt und den Hinweis von den Ebenen

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

Annotations

Lat: 48,171 - Long: 11,53

Titel
Roadworks

Beschreibung

1. Lorem ipsum dolor sit amet
2. consetetur sadipscing elitr
3. sed diam nonumy eirmod tempor
4. Stet clita kasd gubergren

2. Wie alle GeoNode Ressource, Sie Können Berechtigungen an die Karte vergeben um zum Beispiel erlauben dass jemand anders diese Karte auch mitgestalten kann

Mit Personen oder Gruppen teilen X

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

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

Eigentümer: joseca2
Ansehen ▾

Jeder

Registrierte Mitglieder

+ Benutzer / Gruppen hinzufügen

Nach Namen oder Berechtigungen filtern

Name ▾	Berechtigungen
joseca	Bearbeiten ▾

3. Speichern Sie Ihre Karte und lassen Sie eine PDF in A4 Format ausdrucken lassen

Drucken

Titel: Location

Auflösung: 96 dpi

Beschreibung: Gib eine Beschreibung ein...

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

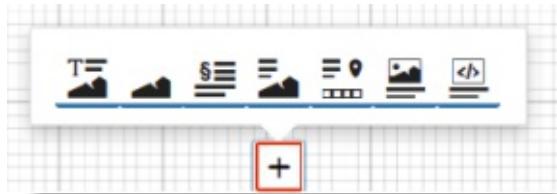
Drucken

Geostories

[testlink](#)

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

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.

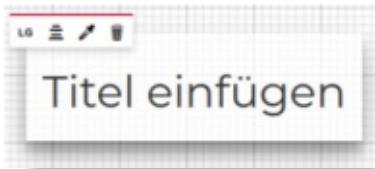
Mit der nächsten Übung bauen wir selber eine Geostory auf.

Übung

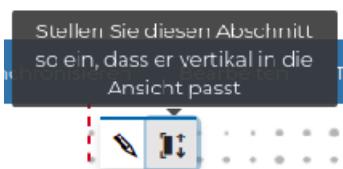
Erstellung einer Geostory

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 einfachen Klicken bearbeiten und formatieren Sie den Titel.

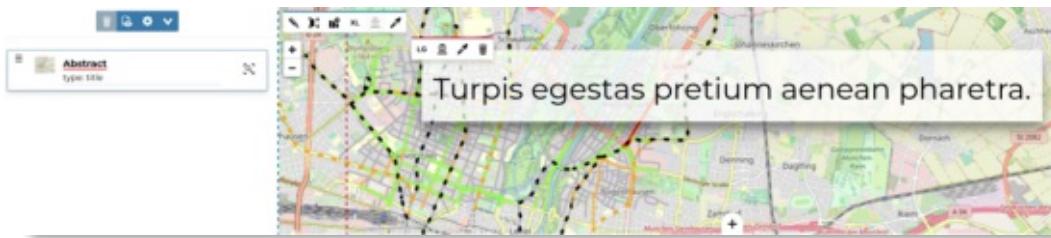


Die Größe des Titelabschnittes können Sie anpassen oder einen Hintergrund hinzufügen mit Bilder, Videos oder Karten.



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.

Absatz Abschnitt
type: paragraph

Text
type: text

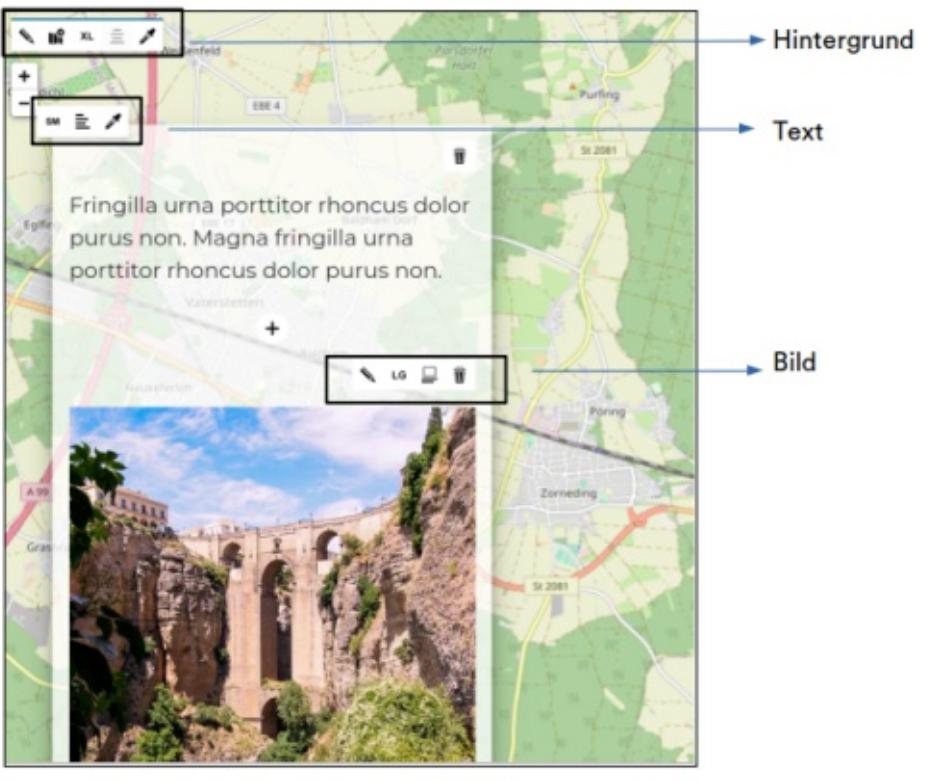
Hier können wir hinzufügen:



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

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

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



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



- Der Hintergrund dient für den gesamten Abschnitt.
- Der Text und das untere kleine Bild werden verknüpft mit einer bestimmten Ort in die Karte.
- Sie können damit beliebigen Anmerkungen an bestimmten Orten hinzufügen.

Dashboards

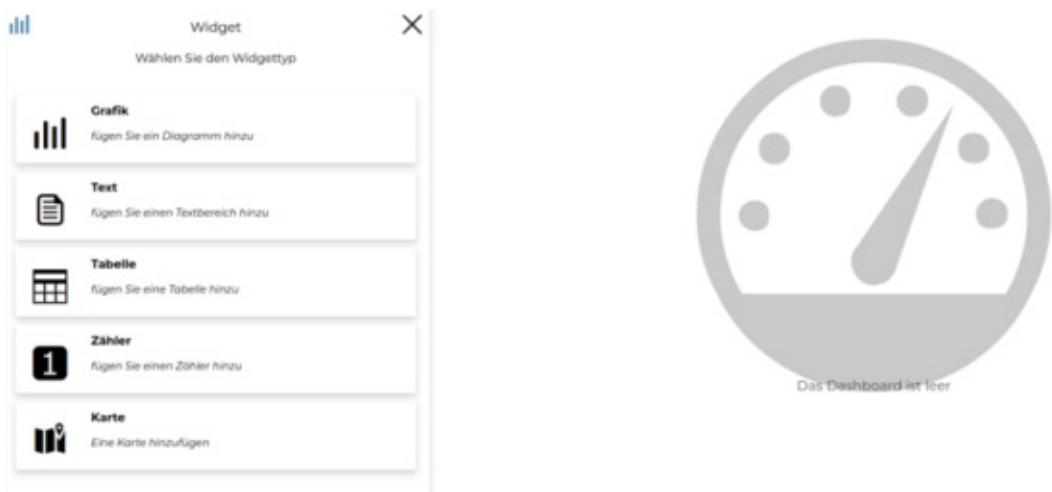
Sie eignen sich gut für die Präsentation von Ergebnissen oder Zusammenfassungen von Daten (siehe [zum Beispiel](#)).

Übung

Erstellung eines Dashboards

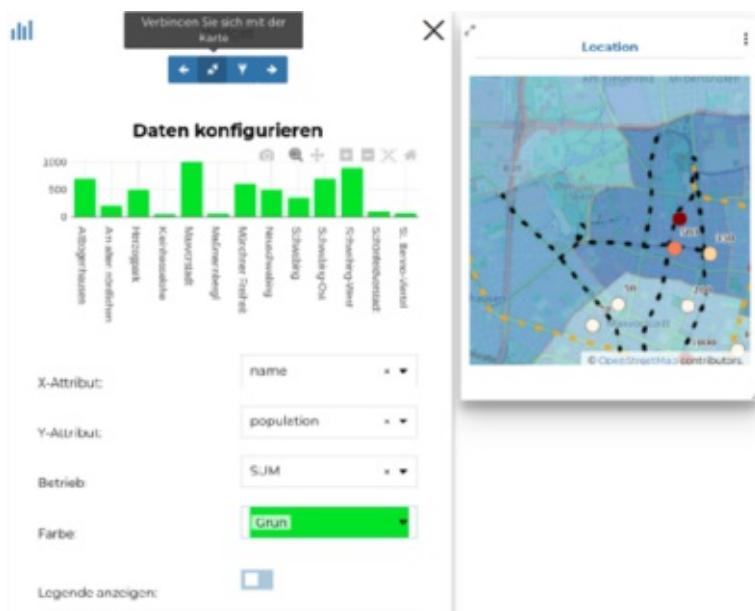
- Um ein Dashboard zu machen müssen Sie wie gewohnt den entsprechenden GeoNode Ressource erstellen:
Ressource hinzufügen → Dashboard erstellen → Speichern

In einem Dashboard können Sie folgende widgets integrieren:



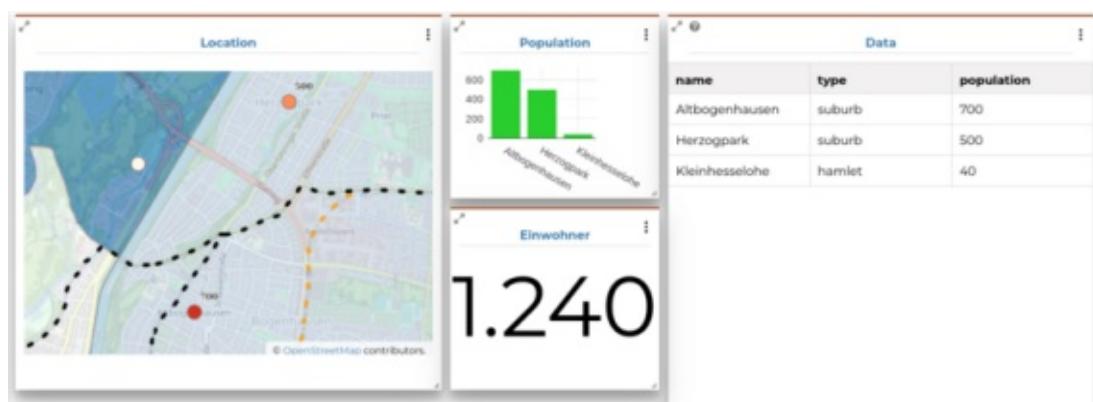
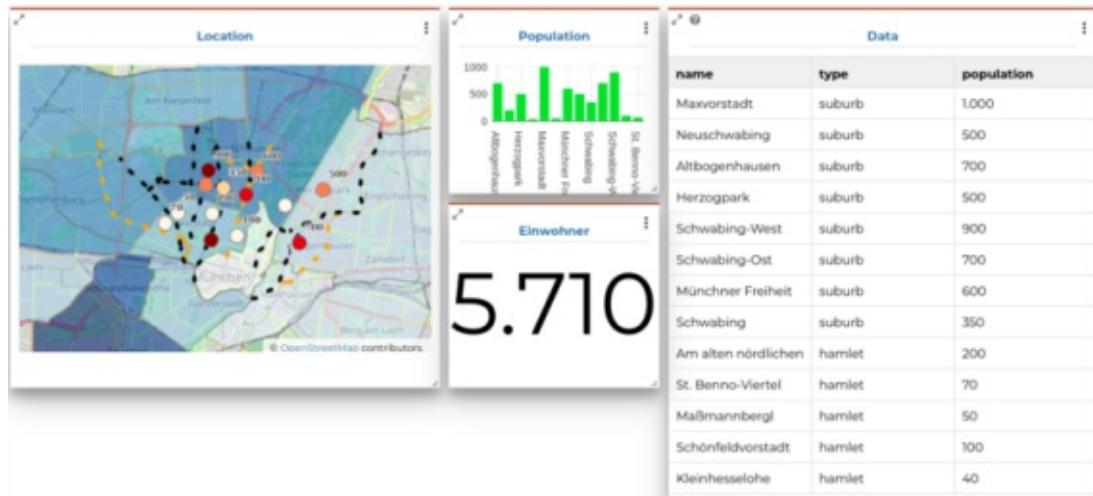
- Fügen Sie zuerst die Karte die sie im Punkt 1.4 erzeugt haben
- Danach bringen Sie in den dashboard folgende Diagramm mit den Einwohnerzahlen.

In Dashboard können die Inhalte dynamisch verknüpft mit anderen Inhalten sein (zum Beispiel wenn man in die Karte zoomt würde das Diagramm nur die Daten anzeigen die man gerade in die Karte sieht)



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

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



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
- Was sind 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

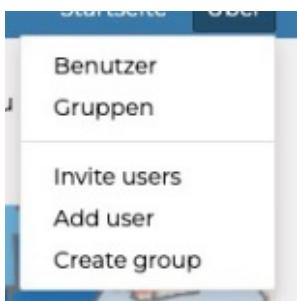
Frontend meint den für die Öffentlichkeit sichtbaren Bereich. Ihm gegenüber steht das Backend, welches das System im Hintergrund bezeichnet.

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. Die E-Mail bittet um Registrierung.

The screenshot shows a web browser window with the URL `geonode-training.cgiis.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 blue 'Absenden' button. At the bottom of the page, there are links to 'geonode.org', 'Developers', and 'About'.

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.cgiis.de`. The title bar says 'Django-Verwaltung' and 'GesNode'. The top navigation bar includes 'admin', 'Auf der Website anzeigen', and user icons. Below the navigation, the breadcrumb trail shows 'Startseite > People > Benutzer > Hinzufügen Benutzer'. The main content is a form titled 'Benutzer hinzufügen' with three fields: 'Benutzername' (username), 'Kennwort' (password), and 'Passwort bestätigen' (confirm password). A note below the password fields says 'Bitte das selbe Passwort zur Bestätigung erneut eingeben.' (Please enter the same password again for confirmation). At the bottom of the form are three buttons: 'Sichern und weiter bearbeiten' (Save and continue editing), 'Sichern und neu hinzufügen' (Save and add another), and a blue 'Speichern' (Save) button.

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 main content area is titled 'Erstellen Sie eine Gruppe' (Create a Group). It contains three input fields: 'Titel' (Title), 'Logo' (Logo), and 'Beschreibung' (Description). The 'Logo' field has a placeholder text 'Datei auswählen. Keine Datei ausgewählt.' (Select file. No file selected.). At the bottom right of the page, there are links to '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

The screenshot shows the 'Gruppen betrachten' (View Groups) page of the GeoNode application. At the top, there is a search bar and a navigation menu with links like 'Daten', 'Karten', 'GeoStories', 'Dashboards', 'Beachtenswert', 'Registrieren', 'Anmelden', 'Startseite', 'Über', 'Test menu 01', and 'Deutsch'. On the left, there is a sidebar with a volcano icon and a large gray square placeholder. The main content area displays a list of groups. The first group is 'test farfagallo group' with the ID 'sfdfdsfgfsdggdsgd'. It has 3 Members and 2 Managers. The second group is 'Test group 1' with the ID 'test'. It has 1 Member and 1 Manager. At the bottom, there is a footer with links to 'geonode.org', 'Developers', and 'About'.

Weiterhin können Administratoren User hinzufügen sowie Details der Gruppe ändern.

In der Pause haben wir zu diesem Zweck Ihren Accounts Administratoren Rechte gegeben.

The screenshot shows the 'Gruppendetails bearbeiten' (Edit Group Details) page for the 'test' group. The page includes a sidebar with several options: 'Gruppendetails bearbeiten', 'Gruppenmitglieder verwalten', 'Diese Gruppe löschen', 'Gruppenaktivitäten', 'Berechtigungen', and 'Manager'. The 'Berechtigungen' section notes that the group is 'Öffentlichkeit' (Public). The 'Manager' section shows 'admin' with a power icon and 'Keine Gruppe' (No Group). The page also displays the group's name 'test' and email 'ts@arc-gestaltung.de'. At the bottom, there are links to 'geonode.org', 'Developers', and 'About'.

Übungen

Neue Gruppe

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

Weiterführende Links

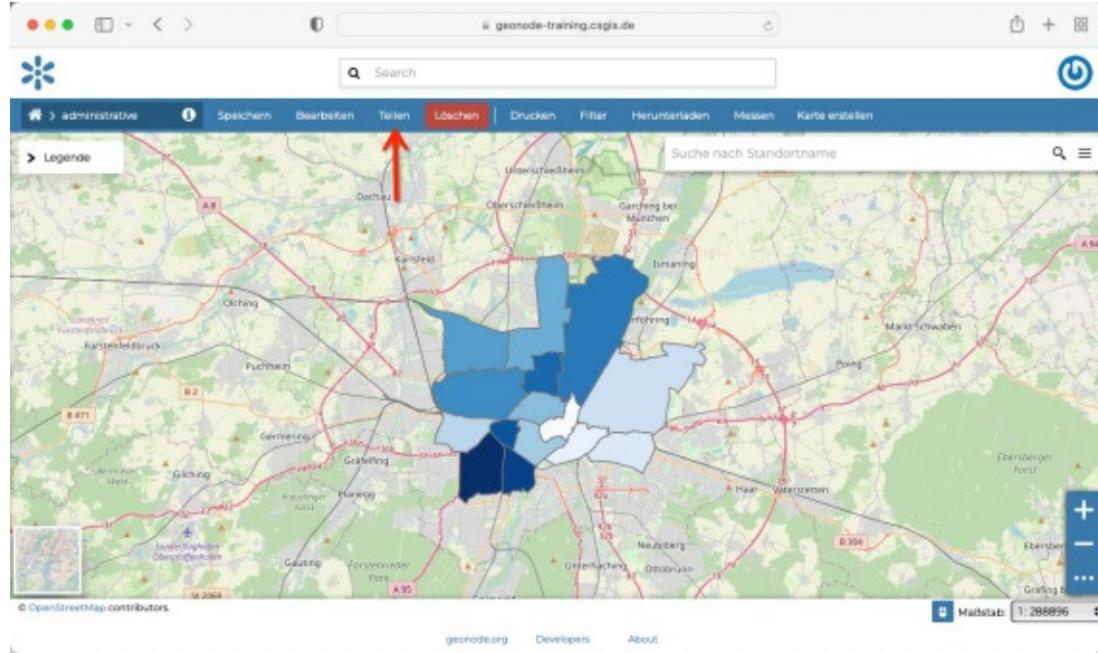
- [GeoNode Docs – Gruppen anlegen](#)

Gruppenrechte

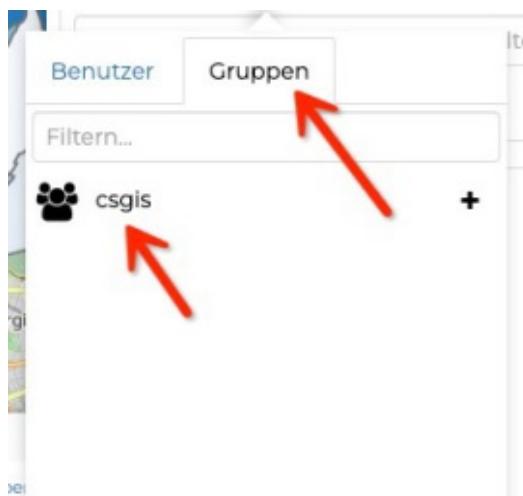
Um Ihren Datensätze-Gruppenberechtigungen zuzuweisen, müssen Sie kein Administrator sein. Jedoch, wie im vorherigen Kapitel gezeigt, um diese zu erstellen.

Rechte zuweisen

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



Im bekannten Panel zur Rechtevergabe wählen Sie die jeweiligen Gruppen:



! [Ebenen Berechtigungen einstellen]

Und vergeben hiernach die gewünschte Sichtbarkeit:

+ Benutzer / Gruppen hinzufügen

Nach Namen oder Berechtigungen filtern

Name ▾	Berechtigungen
 csgis	  <ul style="list-style-type: none">AnsehenAnsehenHerunterladenBearbeitenVerwalten



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 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.

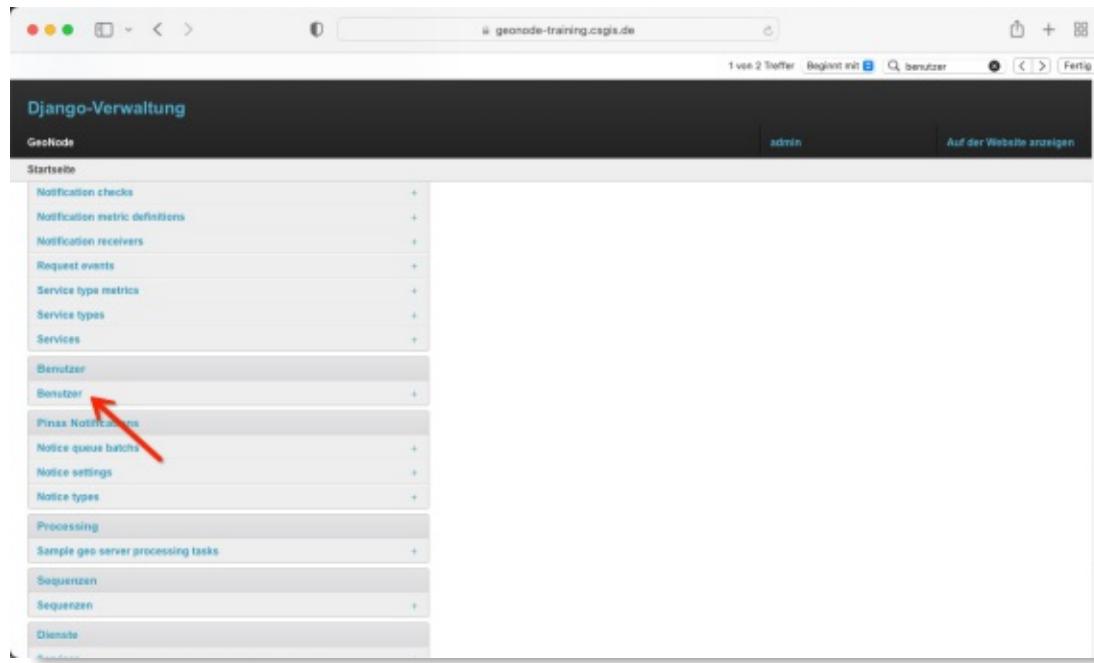


Die hierauf folgende Übersicht listet alle “Django Apps”, die GeoNode besitzt.

A screenshot of the Django Admin interface. The top navigation bar shows 'Django-Verwaltung', '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 of 'Neueste Aktionen' (Recent actions) with entries like 'Ankündigung' (Geo node theme customization), 'Announcement object (3)', 'MyMenu Menu', 'XXX Menu item', 'MyMenu Menu', 'MyMenu Menu', 'MyMenu Menu', 'Ankündigung' (Geo node theme customization), 'Ankündigung' (Geo node theme customization), and 'MyMenu Menu'.

Benutzer verwalten

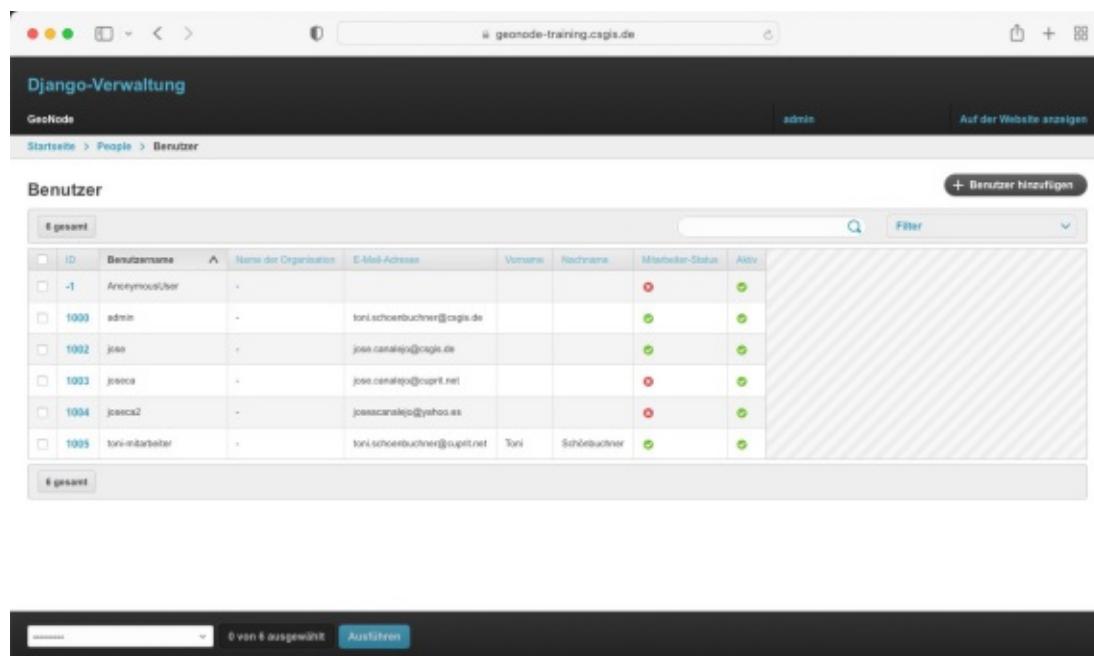
Betrachten wir zunächst einen sehr einfachen, jedoch wesentlichen Abschnitt des Django Admins. Die Benutzerverwaltung.



The screenshot shows the Django Admin interface for a '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 red arrow pointing to the 'Benutzer' link. Below it, under 'IPmax Notifications', are links for 'Notice queue batchs', 'Notice settings', and 'Notice types'. The right side of the screen is currently empty.

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

aufllisten



The screenshot shows the 'Benutzer' list view in the Django Admin. The URL in the address bar is 'geonode-training.csgis.de/admin/people/benutzer/'. The page title is 'Django-Verwaltung' under 'GeoNode'. The breadcrumb navigation shows 'Startseite > People > Benutzer'. At the top right, there is a button '+ Benutzer hinzufügen'. The main area is a table with columns: ID, Benutzername, Name der Organisation, E-Mail-Adresse, Vorname, Nachname, Mitarbeiter-Status, and Aktiv. The table contains 6 rows of data. At the bottom of the table, there is a note '6 present' and a footer with a dropdown menu, a status bar showing '0 von 6 ausgewählt', and a 'Auszählen' button.

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

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

Über die Checkbox und das Menü am Seitenfuß können Sie User löschen. Sowie über den Button in der rechten oberen Ecke neue User anlegen.

löschen

The screenshot shows the 'Benutzer' (User) list page in the GeoNode Django Administration. The table lists users with columns: ID, Benutzername, Name der Organisation, E-Mail-Adresse, Vorname, Nachname, Mitarbeiter-Status, and Aktiv. A red arrow points to the 'Benutzer hinzufügen' (Add User) button in the top right. Another red arrow points to the ID column, specifically the row for user ID 1092.

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

editieren

The screenshot shows the 'Benutzer ändern' (Edit User) detail page for user 'jose'. It includes sections for 'Persönliche Informationen' (Personal Information) and 'Berechtigungen' (Permissions). The 'Berechtigungen' section contains checkboxes for 'Aktiv', 'Mitarbeiter-Status', and 'Administrator-Status'. A red box highlights the 'Aktiv' checkbox. Red arrows point to the 'Löschen' (Delete) button at the bottom left and the 'Speichern' (Save) button at the bottom right.

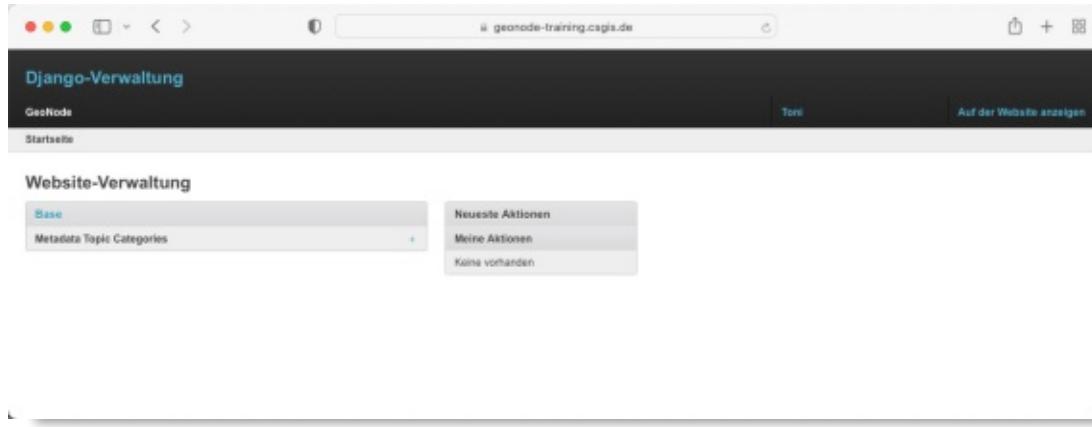
Wie auch in der Übersicht können Sie den User löschen oder Änderungen speichern (Seitenfuß). Das Passwort ändern, oder die Berechtigungen ändern:

- Aktiv
 - Legt fest, ob dieser Benutzer aktiv ist. Kann deaktiviert werden, anstatt Benutzer zu löschen.
- 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

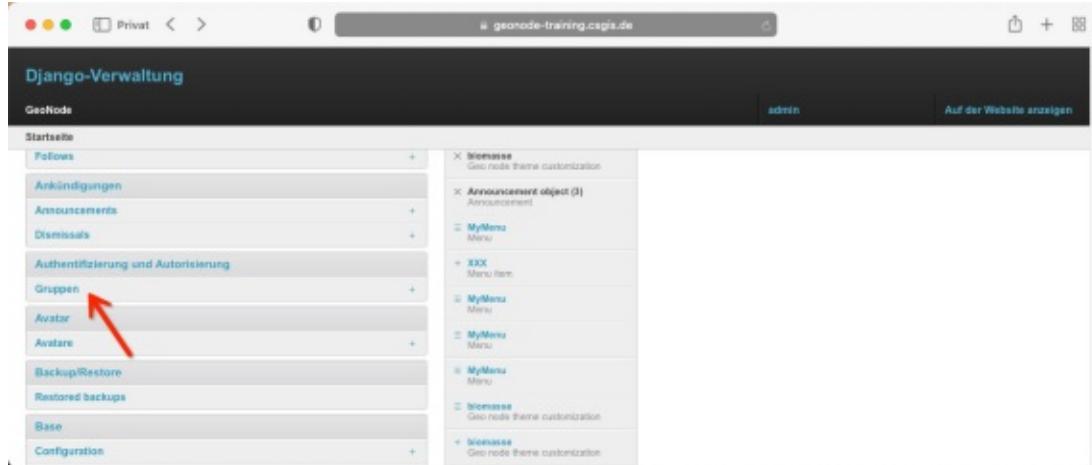
Um Mitarbeiter mit weiteren Rechten auszustatten, ist es nötig

- 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.

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

Mitarbeiter hinzufügen

In der Übersicht gehen wir hierauf auf Benutzer,

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

Alle auswählen Alle entfernen

Die Gruppen, denen der Benutzer angehört. Ein Benutzer bekommt alle Berechtigungen dieser Gruppen. Halten Sie die Strg-Taste (⌘ für Mac) während des Klickens gedrückt, um mehrere Einträge auszuwählen.

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 über 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 Benutzer und Gruppen näher angesehen. Gehen wir eine Stufe weiter und werfen einen Blick auf die Dataset Verwaltung.

Datasets hießen vor Version 3.3 Layer

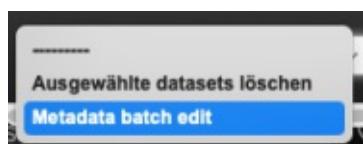
The screenshot shows the Django Admin interface for a 'GeoNode' application. The left sidebar has a tree structure with nodes like 'Task results', 'Dataset', 'Attributes', 'Datasets' (which is highlighted with a red arrow), and 'Styles'. The main content area is currently empty, showing the title 'Django-Verwaltung' and the user 'admin'.

Die Tabelle gibt uns wieder einen kurzen Überblick über nützliche Informationen. Zum Beispiel wann der Datensatz erzeugt wurde, den Titel, oder die ID.

Stapelverarbeitung

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.

The screenshot shows the 'Datasets' list page in the Django Admin interface. It displays a table with columns: ID, Ansicht, Titel [en], Datum, Category, and Group. A red box highlights the 'Ansicht' column. A red arrow points from the 'dataset hinzufügen' button in the top right to a small blue modal dialog at the bottom left. The dialog contains the text 'Ausgewählte datasets löschen' and 'Metadata batch edit'.



Detailansicht

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

The screenshot shows the Django Admin interface for a dataset named 'unser_test2'. The 'Kurzbeschreibung [en]' field is highlighted with a red box. Below the text area, there is a note: 'Kurze Beschreibung des Inhalts der Quelle(n)'. At the bottom right, there are buttons for 'Sichern und weiter bearbeiten' (Save and continue editing), 'Sichern und neu hinzufügen' (Save and add another), and 'Speichern' (Save).

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

The screenshot shows the GeoNode dataset editor for 'unser_test2'. The 'Abstract' field is highlighted with a red box. The interface includes tabs for 'Bearbeiten', 'Voransicht', and 'Einstellungen'. It shows sections for 'Grundlegende Metadaten', 'Ort und Lizenzen', 'Optional', and 'Datensatz Attribute'. There is also a 'Thumbnail' section with a blue triangle image.

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

The screenshot shows a CSW search result for a dataset. The 'pointOfContact' field is highlighted with a red box and contains the following technical metadata:

```
846c490f-c245-49fb-989c-77811374d6c9 eng utf8 dataset
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 -
geonode:unser_test2 unser_test2
toni.schoenbuchner@csgis.de originator None Thumbnail for 'unser_test2' image/png 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 dashboard for GeoNode. On the left, there is a sidebar with the following items: 'GeoNode-Themenbibliothek' (highlighted with a red box), 'Jumbotron theme slides', 'Themes' (also highlighted with a red box), and 'Gruppen'. The main content area shows a table with one row, labeled '1 gesamt'. The table has columns for 'ID', 'Name', 'Date', and 'Description'. The first row contains the value '2' under 'ID', 'Test Theme' under 'Name', and '30. September 2022 12:57' under 'Date'. The 'Description' column is empty. At the bottom 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".

This screenshot shows the 'Themes' list view in the Django Admin. The URL in the browser bar is 'geonode-training.djangos.de/admin/geo_node/theme/'. The page title is 'Django-Verwaltung' and the sub-section is 'Themes'. The table header includes 'ID', 'Name', 'Date', and 'Description'. The first row is highlighted with a red box, showing 'ID: 2', 'Name: Test Theme', 'Date: 30. September 2022 12:57', and 'Description:'. Below the table, there is a footer with a search bar, a selection status '0 von 1 ausgewählt', and a 'Auszählen' button.

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.

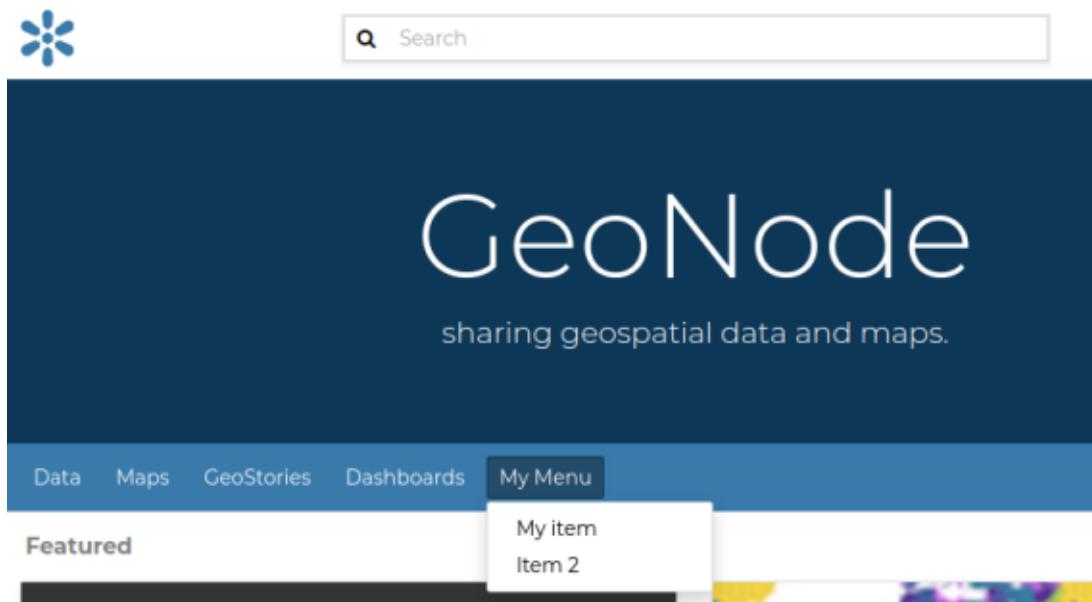
zeigt sich auf der Startseite

Weiterführende Links

- [GeoNode Docs Theming](#)

Eigene Menüpunkte erstellen

GeoNode bietet einige integrierte Funktionen, mit denen Sie das Menü in der oberen Leiste schnell und einfach anpassen können (siehe das Beispiel unten).



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: "Base", "Configuration", "Contact roles", "Hierarchical keywords", "Licenses", "Links", and "Metadata Regions". Under the "Links" section, there are three items: "Menu items", "Menu placeholders", and "Menus". The "Menus" item is highlighted with a red rectangular border.

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 are four buttons: 'Löschen' (Delete), 'Sichern und weiter bearbeiten' (Save and continue editing), 'Sichern und neu hinzufügen' (Save and add another), and 'Speichern' (Save). A red arrow points to the 'Speichern' button.

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	https://geonode-training.csgis.de/about/

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 standard browser controls (back, forward, search, etc.), the URL 'geonode-training.cgiis.de', and user account links ('Registrieren' and 'Anmelden'). Below the URL is a search bar with placeholder text 'Search'. The main menu items are 'Daten', 'Karten', 'GeoStories', 'Dashboards', 'Beachtenswert', and 'Unsere Forschungsdaten'. A red arrow points from the text 'Unsere Vektordaten' in a dropdown menu to the 'Unsere Forschungsdaten' menu item. Other visible elements include a 'Filter' button, a message '13 Ressourcen gefunden', and a map visualization at the bottom.

Ein Klick auf About bringt die User zur About Seite:

The screenshot shows the 'Über GeoNode' (About GeoNode) page. The top navigation bar is identical to the previous one. The main content area has a heading 'Über GeoNode'. Below it is a paragraph of text about GeoNode's purpose and features, followed by two smaller paragraphs. At the bottom of the page, there are 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 Vektordaten' (Our Vector Data) page. The top navigation bar is identical. The main content area displays a list of vector datasets. On the left is a large blue polygon dataset labeled 'unser_test2'. In the center is a dataset labeled 'education' with a database icon. On the right is a map-based dataset labeled 'administrative'. Each dataset entry includes a checkbox, a name, and a 'Ansehen' (View) button.

Mit dieser Technik ließen sich natürlich auch Links speziell einer Gruppe oder in einer Region denken.

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.

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.

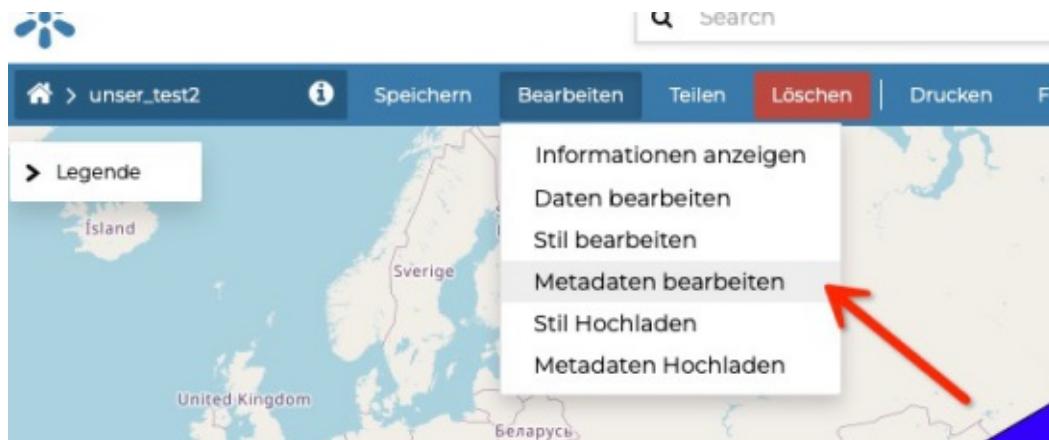
Benutzer können auch ein XML-Metadaten-Dokument (ISO-, FGDC- und Dublin Core-Format) hochladen, um wichtige GeoNode-Metadatenelemente automatisch zu füllen.

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 GeoNode 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 setzen. Dies erfolgt im Frontend auf Ebene Datensatz über “Metadaten bearbeiten”.



Administratoren können in beschränkten Umfang Einfluss auf die angegebenen Daten nehmen. Zwei einfache Beispiele:

Lizenzen anpassen

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

A screenshot of the Django Admin interface. On the left, there's a sidebar with sections like 'Hierarchical keywords', 'Licenses' (which is highlighted with a red box), 'Links', and 'Menu items'. Each section has a '+' button to add new items.

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). In the top right corner, there is a button labeled '+ Lizenz hinzufügen'. A red arrow points to this button.

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'.

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 part

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 in the address bar is 'topic category hinzufügen | Django-Systemverwaltung'. The top navigation bar includes 'admin' and 'Auf der Website anzeigen'. The main content area is titled 'topic category hinzufügen'. It contains several input fields: 'Identifier' (value: 'Soil'), 'Description [en]' (value: 'Soil data'), 'GeoNode description [en]' (value: 'Soil data'), 'Is choice' (checkbox checked), and 'Fa class' (value: 'fa-times'). At the bottom right 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', and 'Soil data'. The 'Soil data' option is selected, indicated by a red border around its input field. Below the dropdown, a note reads: '* Feld ist als verpflichtend im Metadatenschema gekennzeichnet'. The 'Gruppe' tab is also visible at the bottom left.

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 sammeln.

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

Standardmäßig unterstützt GeoNode das Harvesting von:

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

Weiterhin besteht die Möglichkeit, eigene Harvester zu programmieren.

Einmaliges Harvestern

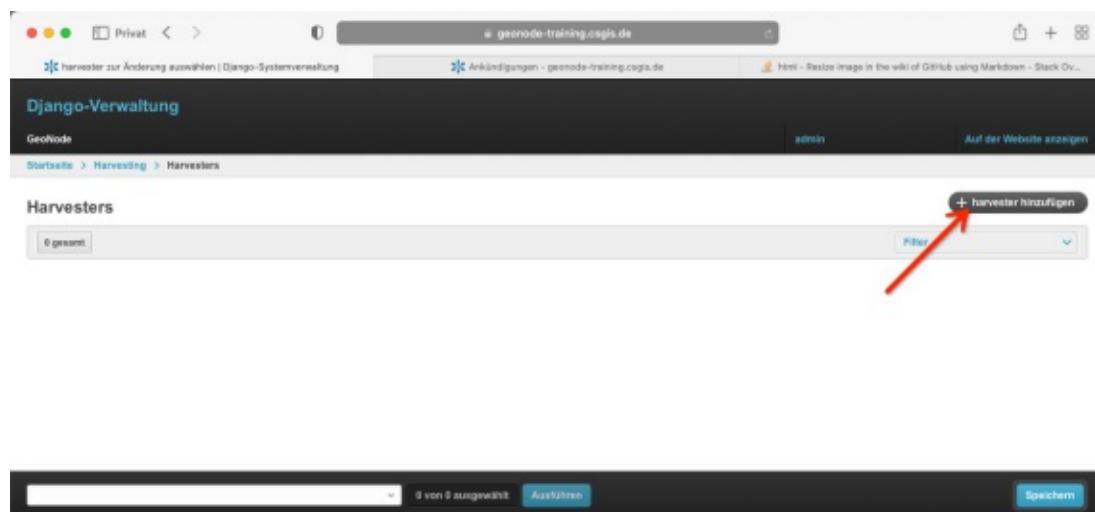
Dieser Workflow wird meist manuell vom Benutzer ausgeführt.

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	https://master.demo.geonode.org/W 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

Harvesting
Asynchronous harvesting sessions
Harvestable resources
Harvesters

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 Django Admin interface for 'Harvestable resources'. A red box highlights the first row in the table, which contains the ID 823, Last harvested (1. Oktober 2022 09:30), Last harvesting succeeded (red circle), Last harvested (1. Oktober 2022 09:30), Unique identifier (879), Harvester (x_2018_02_anrcd), Should be harvested (unchecked), Previous resource type (dataset). Below the table, a red box highlights the 'Auswählen' button in the bottom navigation bar, which is part of a dropdown menu labeled 'Harvest selected resources'.

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

Unser neuer Datensatz ist hiernach im GeoNode Portal sichtbar:

The screenshot shows the GeoNode portal homepage with two new datasets visible: 'Humedal' and 'unser_testz'. Below the portal, a screenshot of the harvested resources table shows the first row with a green checkmark in the 'Last harvesting succeeded' column, indicating successful harvesting.

ID	Last refreshed	Last harvesting succeeded	Last harvested
823	1. Oktober 2022 09:30		1. Oktober 2022 09:30

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”.

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:

Django-Verwaltung

GeoNode

Startseite > Harvesting > Harvestable resources

Harvestable resources

ID	Last refreshed	Last harvesting succeeded	Last harvested	Unique Identifier	Title	Harvester	Should be harvested	Remote resource type
823	1. Oktober 2022 09:30	grün	1. Oktober 2022 09:33	879	t_2018_02_anviD	Manueller Harvester	<input type="checkbox"/>	dataset

Ausgewählte harvestable resources löschen

1 von 1 ausgewählt

Ausführen

Speichern

Zurück in der Übersicht aller Harvester editieren wir den vorab angelegten Harvester "Manueller Harvester".

Django-Verwaltung

GeoNode

Startseite > Harvesting > Harvester

Harvesters

ID	Status	Name	Scheduling enabled	Workers available	Harvestable resources	Selected harvestable resources	Lastest harvesting session	Last refresh date	Worker-specific configurations
2	ready	Manueller Harvester	<input checked="" type="checkbox"/>	grün	0	-	-	-	harvest_documents: True copy_documents: False harvest_datasets: True copy_datasets: False resource_type_filter: t_2018_02_anviD start_date_filter: None end_date_filter: None keywords_filter: None categories_filter: None

Ausgewählte harvester löschen

0 von 1 ausgewählt

Ausführen

Speichern

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

The screenshot shows the 'Django-Verwaltung' (Django Administration) interface for 'Harvester ändern'. The harvester is named 'Automatischer Harvester' and its URL is set to 'https://linmaster.demo.geonode.org'. The 'Scheduling enabled' checkbox is checked. Harvesting session update frequency is set to 2 minutes, and refresh harvestable resources update frequency is also set to 2 minutes. At the bottom, there are buttons for 'Löschen' (Delete), 'Sichern und weiter bearbeiten' (Save and continue editing), 'Sichern und neu hinzufügen' (Save and add another), and 'Speichern' (Save).

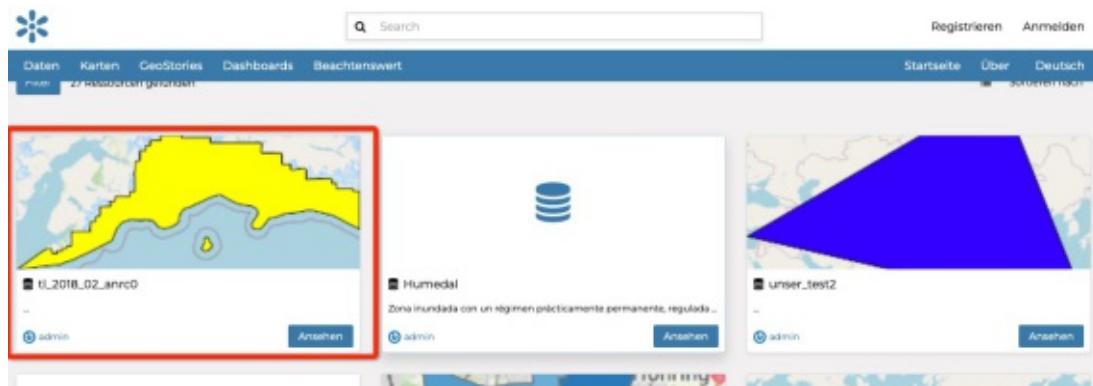
Und speichern die Konfiguration ab.

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

Harvestable resources

A table titled '1 gesamt' showing harvested resources. One entry is listed: ID #24, Last refreshed 1. Oktober 2022 10:07, Last harvesting succeeded 1. Oktober 2022 10:05, Unique identifier 879, Title tl_2018_02_anrc0, Harvester Automatischer Harvester, Should be harvested checked, and Remote resource type dataset.

Sowie die Ebene im Portal sichtbar werden.



Weiterführende Links

- [GeoNode Docs: Harvesting](#)

Ankündigungen (announcements)

Als Administrator müssen Sie von Zeit zu Zeit Ankündigungen über Ihr Portal veröffentlichen.

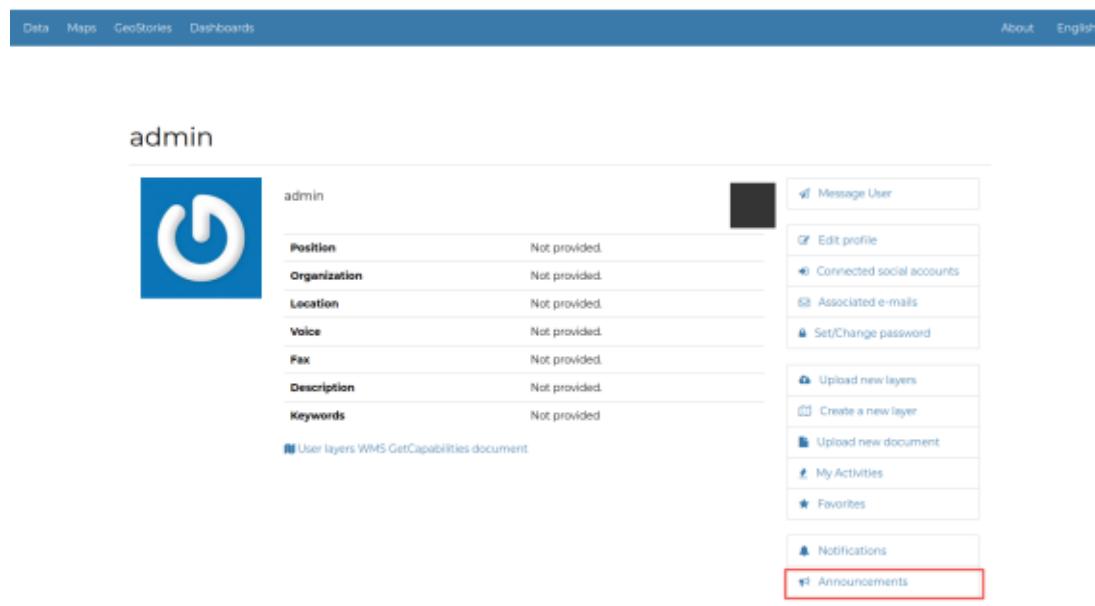
GeoNode “announcements” ermöglichen genau das; ein Administrator hat die Möglichkeit, drei Arten von Nachrichten zu erstellen, entsprechend ihrem Schweregrad, ihre Gültigkeit in Bezug auf den Zeitraum zu bestimmen (Start- und Ablaufdatum der Ankündigung), wer sie sehen kann oder nicht (jeder oder nur die registrierten Mitglieder) und wann ein Benutzer die Nachricht ausblenden.

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:

Über die GeoNode-GUI, aus dem Profil-Panel



The screenshot shows the GeoNode user profile interface. 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' with a blue icon. To the right is a sidebar with various user management and system links. The 'Announcements' link is located at the bottom of this sidebar and is highlighted with a red border.

Über den Django Admin

Die Funktionalitäten sind für beide Oberflächen fast gleich, außer dass es über das Admin-Panel möglich ist, auch die "Ablehnungen" (dismissal) zu verwalten.

"Ablehnungen" sind Aufzeichnungen von Mitgliedern, die die Ankündigung gelesen und das Nachrichtenfeld geschlossen haben. Eine Ankündigung kann eine der drei folgenden "Ablehnungsgründen" haben:

- 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.

Announcement im Frontend über Profil anlegen

Klicken Sie im Profilbereich auf den Link Ankündigungen

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 GeoNode interface with a blue header bar. On the left is a navigation menu with icons for Data, Maps, GeoStories, and Dashboards. On the right are links for About and English. A search bar is at the top center. Below the header is a section titled "Announcements". It contains a table with one row: "Test announcement" (Title), "General" (Level), "Just testing" (Announcement), and "Published from November 29, 2021 to November 29, 2021" (Published From). A blue button labeled "New Announcement" is located in the top right corner of this section. The entire screenshot is framed by a red border.

Füllen Sie hierauf alle Felder des Formulars aus.

Ankündigung über den Django Admin

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

The screenshot shows the Django Admin interface with a dark header bar. The title is "Django administration" and the user is "admin". Below the header is a breadcrumb trail: Home > Announcements > Announcements > Add announcement. The main content area is titled "Add announcement". It contains fields for "Title" (a text input field), "Level" (a dropdown menu set to "General"), "Content" (a large text area), and "Dismissal type" (a dropdown menu set to "Session Only Dismissal"). Below these are checkboxes for "Site wide" and "Members only", and date/time pickers for "Published start" (2021-11-29 13:58:51) and "Published end". The entire screenshot is framed by a red border.

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.

The screenshot shows the Django Admin interface with a dark header bar. The title is "Django administration" and the user is "GeoNode". Below the header is a breadcrumb trail: Home > Announcements > Dismissals > Add dismissal. The main content area is titled "Add dismissal". It contains three rows of fields: "User" (a dropdown menu with a placeholder "-----"), "Announcement" (a dropdown menu set to "Announcement object (1)"), and "Dismissed at" (a date/time picker set to 2021-11-29 14:00:08). The entire screenshot is framed by a red border.

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, showing the contents of the `geonode` directory. A red box highlights the `geonode` folder. The page lists various commits, issues, and pull requests. On the right side, there is an 'About' section with a brief description of GeoNode as an open source platform for geospatial data, along with links to `geonode.org/`, Python, Django, GIS, SDI, and spatial-data-infrastructure. There are also sections for Readme, License, Code of conduct, Star count, Watchers, Forks, and Releases.

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, and the `geonode` folder is expanded. The sub-directories listed are `api`, `base`, `br`, `catalogue`, `client`, `documents`, `favorite`, and `media`. Each sub-directory has its own commit history displayed below it.

Dem/der aufmerksamen Zuhörer*in wird nun auffallen, dass sich die einzelnen Ordner, im “Django Admin” des vorhergehenden Artikels wiederfinden. Grund hierfür ist, dass Django die Administrationsoberfläche automatisiert aus den einzelnen Apps erstellen kann.

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. 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.

fixtures	GeoNode test case refactoring	5 years ago	
management	[Flying] Replace “+” concatenated strings with f{strings}	2 years ago	
migrations	[Fixes #7984] Remove Registered Members group profile (#7985)	13 months ago	
static/people/img	Added the icons for people profile in correct path. Added the necessa...	10 years ago	
templates/people	[Fixes #9748] Cleanup of stale code (#9693)	2 months ago	
templatetags	[Fixes #6804] Map Thumbnails Broken	3 years ago	
__init__.py	GNIP 89: Architecture Design - Resource and Storage Manager Modules (R...	15 months ago	
admin.py	[Fixes #7842] Rename layer to dataset (#7841)	15 months ago	
languages.py	GNIP 89: Architecture Design - Resource and Storage Manager Modules (R...	15 months ago	
models.py	[Hardening] Few consistency checks on some models properties	11 months ago	
search_indexes.py	GNIP 89: Architecture Design - Resource and Storage Manager Modules (R...	15 months ago	
urls.py	[Fixes #9381] Hide users to anonymous users and restrict visibility f...	4 months ago	
views.py	[Fixes #9381] Hide users to anonymous users and restrict visibility f...	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 Datenbanen 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 Abhängigkeiten 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 in der Terminal-Ausgabe folgende Meldung sehen:

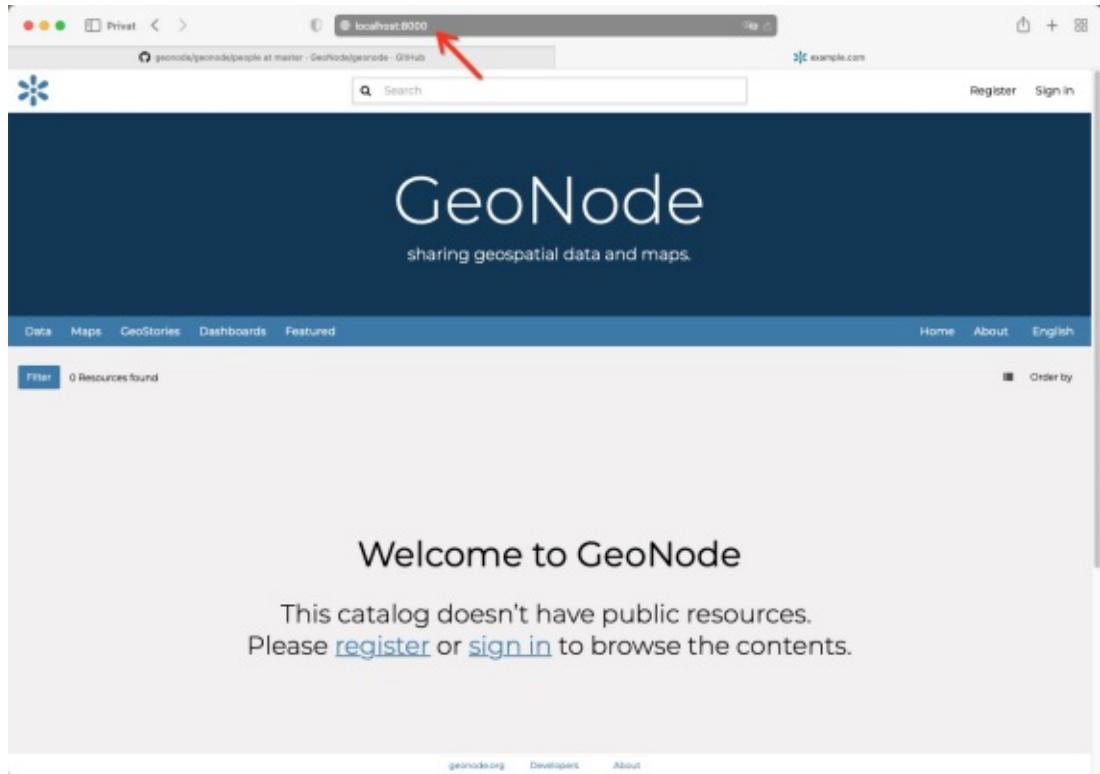
```

WARNING: Illegal reflection access by Javassist.util.proxy.SecurityActions (file:/Users/tonischenbuehner/dev/github/gr-sauth3/ils/geoserver/geoserver/WEB-INF/lib/javassist-3.12.0.GA.jar)
        to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,javax.security.PrivilegedDomain)
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@80aaefab@server@/geoserver
2022-08-03 12:15:14.095:INFO[oejs]ServerConnector.main: Started o.e.j.w.WebAppContext@80aaefab@server@/geoserver
2022-08-03 12:15:14.095:INFO[oejs]AbstractConnector.main: Started ServerConnector@22a71001[HTTP/1.1, (http/1.1)]@0.0.0.0:8000
2022-08-03 12:15:14.095:INFO[oejs]ServerMain: Started @1999ms
2022-08-03 12:15:14.244:INFO[oejs]Geoserver:cp1586270994-21: Initializing Spring DispatcherServlet 'dispatcher'
The logs are available at /Users/tonischenbuehner/dev/github/gr-sauth3/ils/geoserver/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.

http://0.0.0.0:8000/
```

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



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 'localhost:8000/geonode/people/1/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: 'Diese Plattform dient der Publikation von Daten zum Thema Weltraum. 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.' At the bottom of the page are 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.

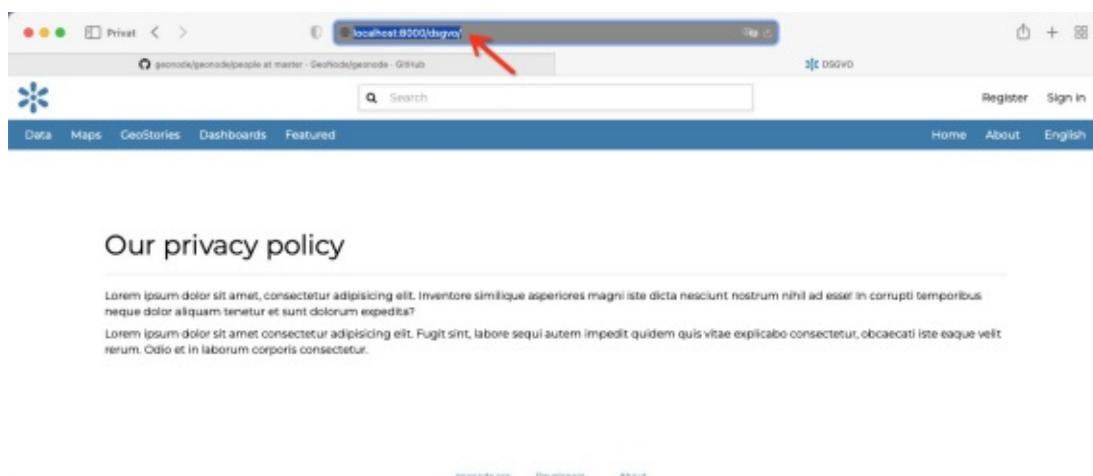
```
{% extends "geonode_base.html" %}  
{% load i18n %}  
{% block title %}DSGVO{% endblock %}  
  
{% block body_outer %}  
    <div class="page-header">  
        <h2>{% trans "Our privacy policy" %}</h2>  
    </div>  
    <p>  
        Lorem ipsum dolor sit amet, consectetur adipisicing elit. Inventore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

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

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

4 directories, 2 files
```

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

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

```
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 · GitHub". The header includes a search bar, a language switcher ("de DE"), and navigation links for "Register" and "Sign in". Below the header is a navigation menu with links for "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 · GitHub". The header includes a search bar, a language switcher ("de DE"), and navigation links for "Registrieren" and "Anmelden". Below the header is a navigation menu with links for "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  
└── __pycache__  
    └── __init__.cpython-39.pyc  
    ├── admin.cpython-39.pyc  
    ├── apps.cpython-39.pyc  
    ├── helper.cpython-39.pyc  
    ├── models.cpython-39.pyc  
    └── views.cpython-39.pyc  
└── admin.py  
└── apps.py  
└── helper.py  
└── migrations  
    └── __init__.py  
        └── __pycache__  
            └── __init__.cpython-39.pyc  
└── 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 Hilfunktionen 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>
    <ows:DataInputs>

```

```

<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/gpk/1.0.0">
                        {partials}
                    </geopackage>
                ]]>
            </wps:ComplexData>
        </wps:Data>
    </wps:Input>
</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

Abschließend 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>


| Auswahl                                                                  | Vector Dataset     |
|--------------------------------------------------------------------------|--------------------|
| <input type="checkbox" name="layerselect" value="{{ object.alternate }}> | {{ object.title }} |


</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")
.element.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

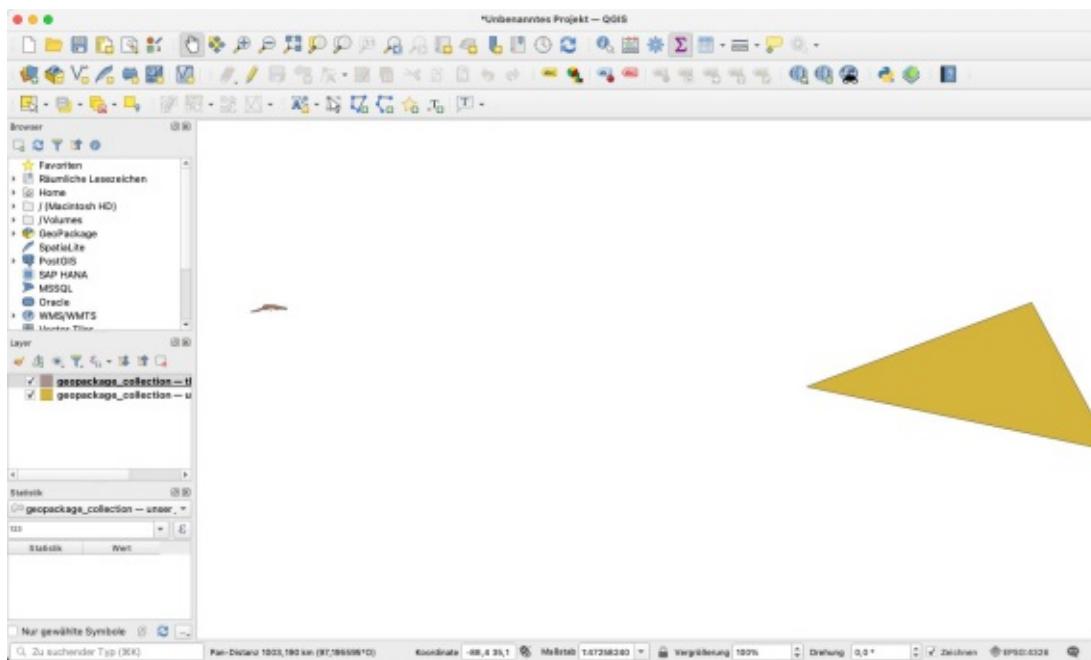
Unter unserer neuen URL `http://localhost:8000/gpkg` sollten wir alle Vektor Datensätze aufgelistet bekommen.

The screenshot shows a web browser window with the address bar set to "localhost". The page content is titled "GeoPackage erzeugen". On the left, there's a table with two columns: "Auswahl" and "Vector Dataset". Under "Auswahl", there are two checked checkboxes corresponding to the datasets "tL_2018_02_anrc" and "unser_test2". At the bottom left of the page, there is a button labeled "Generiere Geopackage von Auswahl!". The footer of the page includes links to "geonode.org", "Developers", and "About".

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. At the bottom of the page, there are links to 'geonode.org', 'Developers', and 'About'.

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.

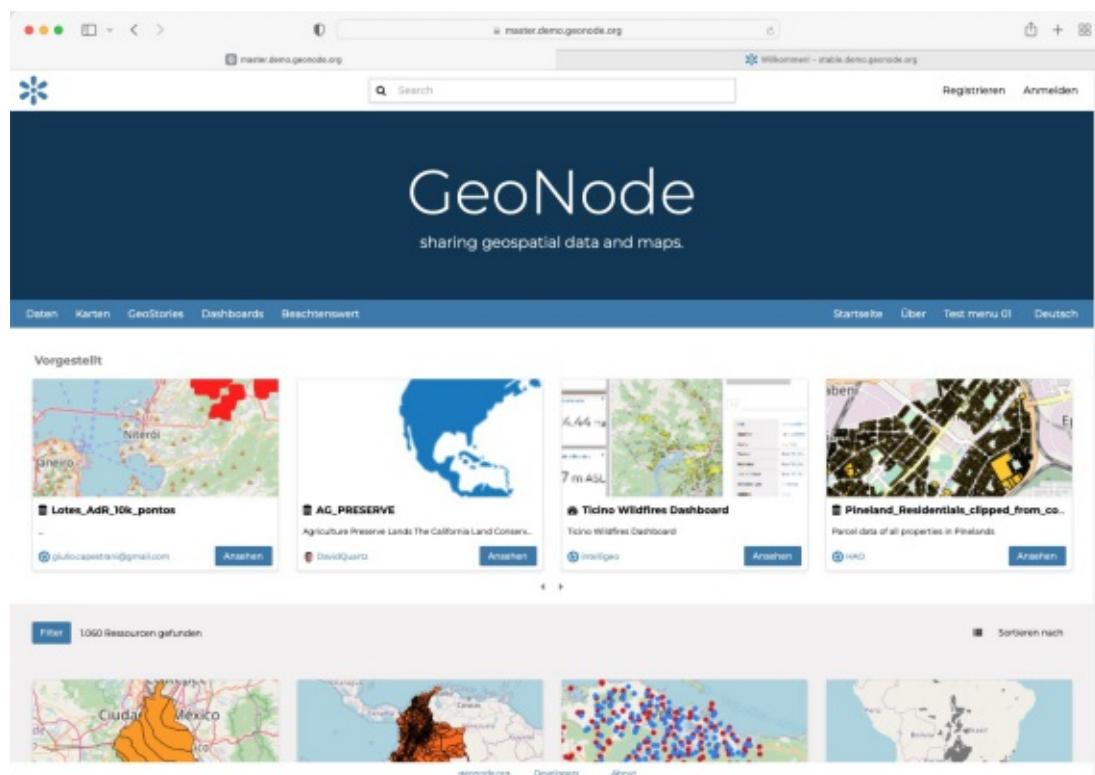
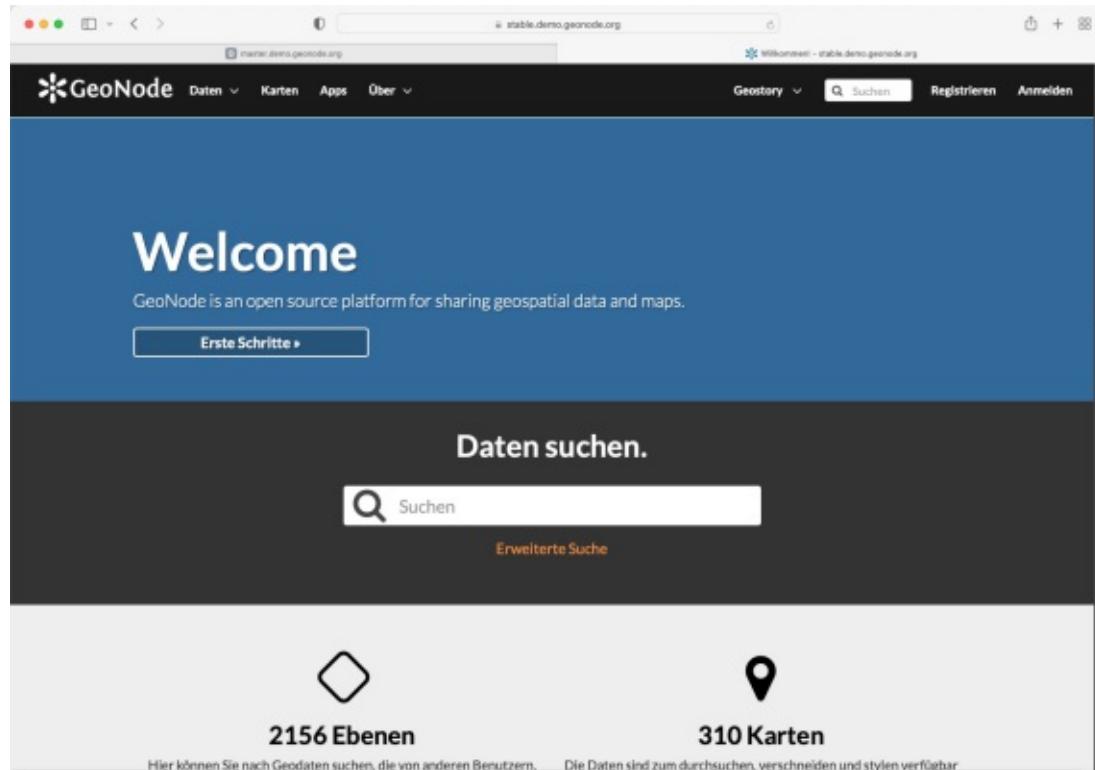
Selbstverständlich ließe sich dieses Beispiel noch weiter ausarbeiten. Zum Beispiel ein eigenes Datenbankmodell erstellen, um Daten zu speichern und die Django Administrationsoberfläche zu verwenden.

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.



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](#), basiert wurde entschieden weitere Komponenten (wie die Startseite) ebenfalls hierüber abwickeln zu lassen.

React ist eine JavaScript-Programmbibliothek[3] 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 agieren und wie diese 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

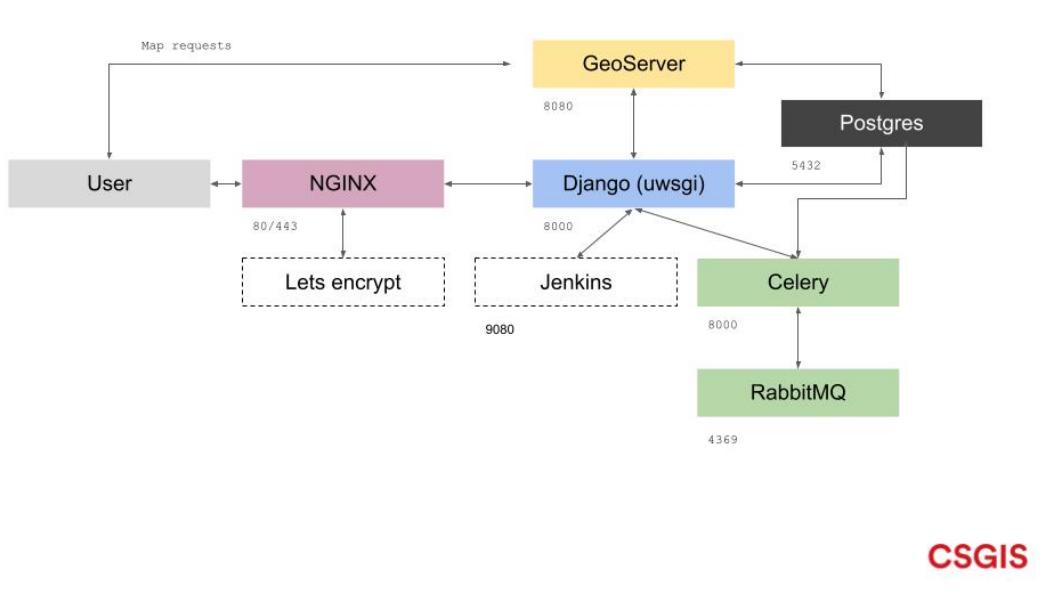
besitzen.

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](#).

Spezifisches für GeoNode

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

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 zu 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=postgres://{{project_name}}:{dbpwd}@db:5432/{{project_name}}
 - Die Verbindungsparameter zur Datenbank
- GEODATABASE_URL=postgres://{{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

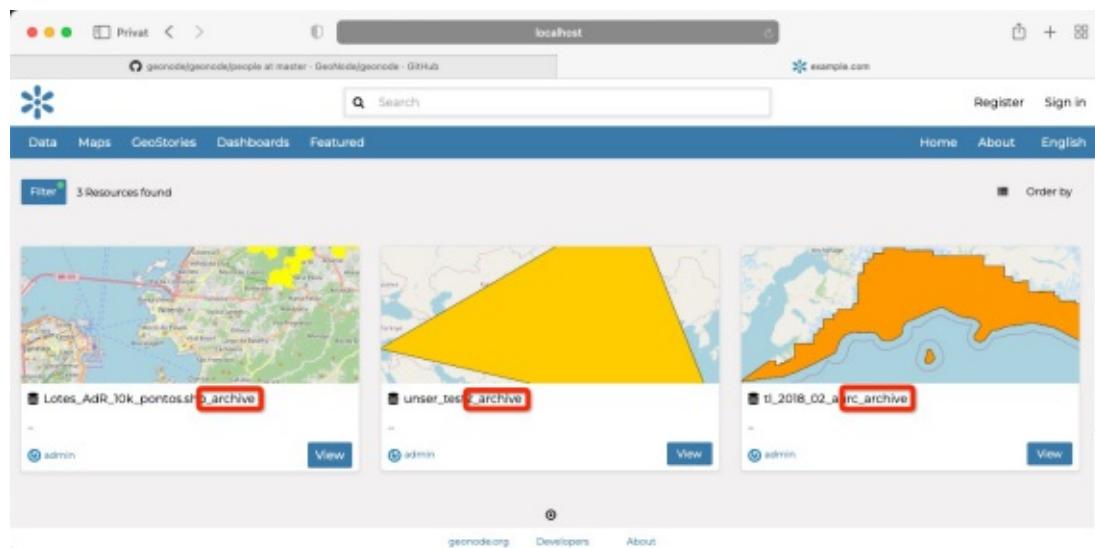
```
python ./manage.py shell
```

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

Beispiel: Ändern aller Titel bestehender Datensätze

```
In [1]: from geonode.base.models import ResourceBase
In [2]: all_datasets = ResourceBase.objects.all()
In [3]: for dataset in all_datasets:
...:     dataset.title += '_archive'
...:     dataset.save()
...:
```

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



Weiterführende Links

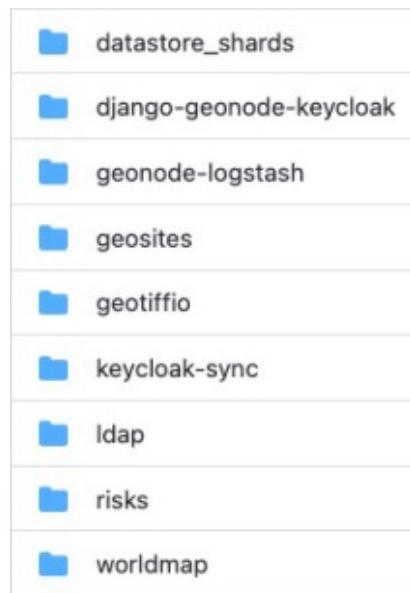
- [Django Dokumentation - manage.py](#)

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. This node contains a list of tables, each preceded by a blue table icon and a right-pointing arrow. The tables listed are: 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, and 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. This node contains a list of tables, each preceded by a blue table icon and a right-pointing arrow. The tables listed are: 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), and 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
```

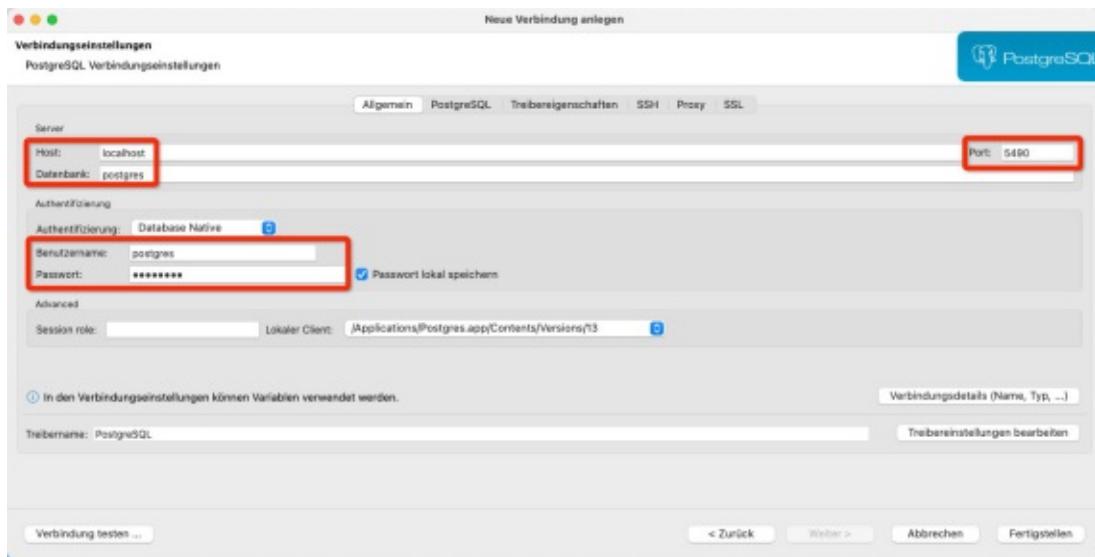
Über die -L Angabe binden wir den Port 5490 an den Port 5432 des Servers. (In diesem Fall geonode-training.csgis.de)

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
e4200e848c28 geonode/geoserver:2.20.5 "/usr/local/tomcat/t..." 7 days ago Up 7 days (healthy) 8.0.0.0:8080->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 8.0.0.0:80->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 8.0.0.0:9080->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 etnweder 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. Beachte den erforderlichen Zugang zu
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 files
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"

# create archive
tar cvfj /backups/bba-geonode.tar_${NOW}.bz2 ${BPTH} && rm -R ${BPTH}

# Delete old
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 besitzen

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.

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](#)