
geonode*custom_metadataappDocumentation*

Release latest

Mar 10, 2023

CONTENTS

- 1 About 3**
 - 1.1 What can this app do for you? 3
 - 1.2 Currently available fields 4
- 2 Installation 5**
- 3 Contribution 7**
- 4 Translations 9**
- 5 Development 11**
 - 5.1 URLs 11
 - 5.2 View decorators 11
 - 5.3 Helpers 11
 - 5.4 Templates 12
 - 5.5 Form defintions 12
- 6 Management Commands 13**



Custom metadata

Attention this application is under development and not released for live operation.

ABOUT

GeoNode's metadata model is based on the ISO 19115 standard. This standard defines the structure and content of metadata for geographic information and is widely used in the spatial data community. The standard includes information such as resource title, summary, keywords, spatial and temporal extent and contact information.

However, there is often a need to add custom metadata to layers or maps.

1.1 What can this app do for you?

This app extends

- the metadata wizard with freely definable form fields
- the metadata detail view of a resource
- the XML export of a dataset
- creates form fields for data added by the REST API

This is how the Wizard looks like with extra fields

The screenshot displays the GeoNode metadata wizard interface. At the top, there are three tabs: 'Bearbeiten' (active), 'Voransicht', and 'Einstellungen'. Below the tabs is a progress bar with five steps, each represented by a numbered circle (1-5) and a label below it. The progress bar is divided into two sections: 'Verpflichtend' (red) for steps 1-3 and 'Optional' (blue) for steps 4-5. The steps are: 1. Grundlegende Metadaten, 2. Ort und Lizenzen, 3. Optionale Metadaten, 4. Datensatz Attribute, and 5. Extra Metadaten. A blue button labeled 'Advanced Metadata' is located at the top right. The form below the progress bar contains several fields: 'Age:' with a text input '4,6 Milliarden Jahre'; 'Planet:' with a text input 'Earth'; 'Tage:' with a date input '19.01.2023' and a calendar icon; 'My Radio:' with three radio button options: 'Option 1' (selected), 'Option 2', and 'Option 3'; 'Is Checked?' with an unchecked checkbox; 'Color:' with a text input 'Mostly blue X'; and 'MyCategory:' with a dropdown menu showing 'Food'.

1.2 Currently available fields

The app currently offers following input fields

- NumberField
- CharField
- DateField (with a calendar picker)
- RadioSelectField
- BooleanField

and following widgets:

- TextInput
- Select
- CheckboxInput
- TextArea
- HiddenInput
- DatePicker
- RadioSelect

INSTALLATION

The app is currently still under development and should not be installed. It can be installed later via pypi or github. Instructions will follow.

CONTRIBUTION

Participation is most welcome. The following link shows the things that are currently pending

[TODO](#)

TRANSLATIONS

By adding a file called `custom_metadata/metadata_fields/translations.json` one can add language specific translations. An example is shown below. The object key is the name of the field, the value the translated label.

DEVELOPMENT

This document explains the program flow in case you plan to contribute.

5.1 URLs

The app overwrites the route defined in geonode depending on the resource/app. Responsible for this are the url definitions in custom_metadata/urls

5.2 View decorators

The views defined here can be found in custom_metadata/views.py Currently there are two generic views that “decorate” the original views in geonode.

A deocrator view responsible for feeding the matdata detail view with its own template.

```
custom_metadata.views.handle_generic_metadata_form
```

A decorator view responsible for feeding a custom Django form into the context for GET requests. It also updates the m2m relation of the resource base model for POST requests.

5.3 Helpers

```
custom_metadata.dynamic_form.CreateExtraMetadataForm
```

Generates the Django form enriched with data from the database.

```
custom_metadata.get_item_config.GetItemConfig
```

Sets the appropriate configuration based on the called URL (defines the app).

```
custom_metadata.resolve_resource.resolve_resource_type
```

Reads the matching resource from the database based on the type. (used by handle_generic_metadata_form)

```
custom_metadata.get_item_config.GetItemConfig
```

Selects the right “return” object matching the app (decorated by the handle_generic_metadata_form).

5.4 Templates

The customized templates can be found in
`custom_metadata/templates`

5.5 Form definitions

The json based Django Form definitions can be found in
`custom_metadata/metadata_fields`

MANAGEMENT COMMANDS

The App currently defines three basic management commands

```
delete_all_custom_metadata
```

Delete all ManyToMany relations from `geonode.models.ResourceBase`

```
delete_resource_custom_metadata
```

Delete the metadata associated with a `ResourceBase` instance

```
show_resource_custom_metadata
```

Print the metadata associated with a `ResourceBase` instance