Collective geo Documentation

Release 2.0

Giorgio Borelli

May 02, 2015

Contents

`collective.geo.geographer` provides geo-annotation for Plone.

This package is based on Sean Gillies's idea (zgeo.geographer) and integrates its functionalities in collective.geo project. Found a bug? Please, use the issue tracker.

Table of contents

CHAPTER 1

Requirements

- Plone >= 4

CHAPTER 2

Installation

This addon can be installed like any other addon, please follow the official documentation.

CHAPTER 3

How it works

Any object that implements `IAttributeAnnotatable` and `IGeoreferenceable` can be adapted and geo-referenced.

All Zope content objects provide the former, and the latter can be easily configured via ZCML.

Let's test with an example placemark, which provides both of the marker interfaces mentioned above:

```
>>> from zope.interface import implements
>>> from zope.annotation.interfaces import IAttributeAnnotatable
>>> from collective.geo.geographer.interfaces import IGeoreferenceable

>>> class Placemark(object):
...     implements(IGeoreferenceable, IAttributeAnnotatable)

>>> placemark = Placemark()
```

Adapt it to `IGeoreferenced`:

```
>>> from collective.geo.geographer.interfaces import IGeoreferenced
>>> geo = IGeoreferenced(placemark)
```

Its properties should all be `None`:

```
>>> geo.type is None
True
>>> geo.coordinates is None
True
>>> geo.crs is None
True
```

Check whether the geo-referenceable object has coordinates or not:

```
>>> geo.hasCoordinates()
False
```

Now set the location geometry to type *Point* and coordinates *105.08 degrees West, 40.59 degrees North* using `setGeoInterface`:

```
>>> geo.setGeoInterface('Point', (-105.08, 40.59))
```

A georeferenced object has `type` and `coordinates` attributes which should give us back what we put in:

```
>>> geo.type
'Point'
>>> tuple(['%.2f' % x for x in geo.coordinates])
```

```
('-105.08', '40.59')
>>> geo.crs is None
True
```

Now the `hasCoordinates` method returns True:

```
>>> geo.hasCoordinates()
True
```

An event should have been sent:

```
>>> from zope.component.eventtesting import getEvents
>>> from collective.geo.geographer.event import IObjectGeoreferencedEvent
>>> events = getEvents(IObjectGeoreferencedEvent)
>>> events[-1].object is placemark
True
```

To remove the coordinate from a georeferenced object, we can use the `removeGeoInterface` method:

```
>>> geo.removeGeoInterface()
>>> geo.type is None
True
>>> geo.coordinates is None
True
>>> geo.crs is None
True
```

3.1 Plone integration

Add geo-referenced content:

```
>>> from plone.app.testing import setRoles
>>> from plone.app.testing import TEST_USER_ID
>>> portal = layer['portal']
>>> setRoles(portal, TEST_USER_ID, ['Manager'])

>>> oid = portal.invokeFactory('Document', 'doc')
>>> doc = portal[oid]
```

If the content type doesn't implement `IGeoreferenceable` interfaces, we need to provide it:

```
>>> from zope.interface import alsoProvides
>>> alsoProvides(doc, IGeoreferenceable)
```

Now we can set the coordinates:

```
>>> from collective.geo.geographer.interfaces import IWriteGeoreferenced
>>> geo = IWriteGeoreferenced(doc)
>>> geo.setGeoInterface('Point', (-100, 40))
```

and reindex the document:

```
>>> doc.reindexObject(idxs=['zgeo_geometry'])
```

We can create a subscriber for `IObjectGeoreferencedEvent` to do that automatically.

Check the catalog results:

```
>>> from Products.CMFCore.utils import getToolByName
>>> catalog = getToolByName(portal, 'portal_catalog')
>>> brain = [b for b in catalog({'getId': 'doc'})][0]
>>> brain.zgeo_geometry['type']
'Point'
>>> brain.zgeo_geometry['coordinates']
(-100, 40)
```

A simple view (`geoview`) notifies us if a context is geo-referenceable:

```
>>> view = doc.restrictedTraverse('@@geoview')
>>> view.isGeoreferenceable()
True
```

and allows us to find its coordinates:

```
>>> view.getCoordinates()
('Point', (-100, 40))
```

When we remove the coordinates, the corresponding index will return `None`:

```
>>> geo.removeGeoInterface()
>>> doc.reindexObject(idxs=['zgeo_geometry'])
>>> brain = [b for b in catalog({'getId': 'doc'})][0]
>>> brain.zgeo_geometry
Missing.Value
```

CHAPTER 4

Package documentation

4.1 Interfaces

4.2 Classes

CHAPTER 5

Indices and tables

- *genindex*
- *modindex*
- *search*