
Collective geo Documentation

Release 2.1

Giorgio Borelli

June 21, 2014

1	Installation	1
1.1	Dependencies	1
1.2	Buildout	1
2	How to	3
2.1	...configure c.geo	3
2.2	...georeference a content type	5
2.3	...aggregate locations in a map	11
2.4	...override a map layer template	12
2.5	...override KML file	13
3	Core Packages	15
3.1	collective.geo.geographer	15
3.2	collective.geo.openlayers	15
3.3	collective.geo.settings	16
3.4	collective.z3cform.mapwidget	17
4	Extra Packages	19
4.1	collective.geo.usersmap	19
4.2	collective.geo.flexitopic	21
5	Overview	23
6	Indices and tables	25

Installation

1.1 Dependencies

Collective geo and collective.geo.contentlocation in particular, depend on Shapely, a library for the manipulation of planar geometric objects and Shapely itself depends on libgeos_c > 3.1

Before installing collective.geo you have to install libgeos_c library on your machine.

1.1.1 Debian/Ubuntu

On Linux systems based on Debian, you have to install the following packages:

- libgeos-c1
- libgeos-dev

1.1.2 Mac OS X

To install libgeos on Mac OS X system you can use macports and install this package:

- geos

1.2 Buildout

The best way to install collective geo is using zc.buildout and adding the collective.geo.bundle package.

This package contains all core packages of c.geo that are suitable for common use.

Below an example of configuration:

```
[buildout]
...
eggs =
    ...
    collective.geo.bundle
...
```

You can also use each package separately to better suit your needs.

For instance, you can use only c.geo.geographer if you need to annotate geographical information on Plone content types or you can use only c.geo.openlayers to integrate OpenLayers javascript in Plone. See above for package details.

After running the buildout and started zope you have to install the product Plone Maps (collective.geo) from Addons Control Panel.

For development purpose you can use the buildout configuration included in collective.geo.bundle.

In this package you'll find two different configurations:

- buildout.cfg: a generic buildout configuration that includes collective.geo.bundle and its dependencies
- devel.cfg a specific development configuration that includes all c.geo source packages from github

2.1 ...configure c.geo

After installing `collective.geo` on your Plone site you should configure it using its control panel. Go to the Plone control panel and click on Collective geo link.

In the collective geo settings form you can:

- choose which content types can be georeferenced;
- choose which map layers you need, for instance Open Street Map, Google Maps, Bing maps;
- set the default center of the map to be used when no other coordinates are provided;
- select the default styles of the geographical shapes on the map;
- choose the position where the map will appears when you display a georeferenced content type;
- use the “advanced tab” to setup new viewletmanagers.

Take a look to these screenshots:

Geo Settings

[Up to Site Setup](#)

Collective Geo Default Settings

Base settings ■ Style ■ Advanced

Georeferenceable content types

Choose which content types (e.g. Page, News Item, etc) can be georeferenced. Be aware that, when you change this setting, the system might take quite some time to apply it to the whole site, especially if you have many objects within it.

Collection	<input type="checkbox"/>	<input type="checkbox"/>
Collection (old-style)	<input type="checkbox"/>	<input type="checkbox"/>
Comment	<input type="checkbox"/>	<input type="checkbox"/>
Event	<input type="checkbox"/>	<input type="checkbox"/>
File	<input type="checkbox"/>	<input type="checkbox"/>

Default map layers

Choose which layers will be activated on the map layout.

Yahoo Hybrid	<input type="checkbox"/>	OpenStreetMap	<input type="checkbox"/>
Yahoo Street	<input type="checkbox"/>	Terrain (Google)	<input type="checkbox"/>
Yahoo Satellite	<input type="checkbox"/>	Hybrid (Google)	<input type="checkbox"/>
Bing Aerial	<input type="checkbox"/>	Satellite (Google)	<input type="checkbox"/>
Bing Roads	<input type="checkbox"/>	Google	<input type="checkbox"/>

Map position ■

Choose if and where the map will be displayed within the georeferenced content view.

Above Content ▾

Default map position

Set the centre point and the zoom level of the map used when the system cannot use any other point to center the map (e.g. when displaying the map to georeference an item for the first time)

Map
Manual settings

Find places

Geo Settings

[Up to Site Setup](#)

Collective Geo Default Settings

Base settings ■ Style ■ **Advanced**

Set default styles for geographical shapes

Marker image ■

The path to the image used as marker in maps. You can use either an absolute URL or a TAL expression.

`string:${portal_url}/img/marker.png`

Marker image size ■

Choose the scaling factor of the marker shown on maps.


Line color ■

Choose the color for the line feature



Polygon color ■

Choose the color for the polygon feature



Line width ■

Choose the width of the line feature

OpenLayers theme

Provide the base path of the OpenLayers theme. You can use an absolute URL or a TAL expression.

`string:${portal_url}/img/`

Map width

Choose the width of the displayed map, specified as an absolute value (e.g. '450px' or '15em'), or relative (e.g. '100%') size.

Map height

Choose the height of the displayed map, specified as an absolute value (e.g. '450px' or '15em'), or relative (e.g. '100%') size.

Balloon details

Choose which properties of the georeferenced content will be displayed in the additional informations section of the balloon.

ID	Title
Type	Description
Subject	
Content Location	
Last Modified Date	

Geo Settings

[Up to Site Setup](#)

Collective Geo Default Settings

Base settings ■ **Style** ■ Advanced

Advanced configurations

Viewlet managers

Specify which viewlet manager can contain the map displayed on the georeferenced content view. Insert them one per line, where each line has the 'dotted name|title' format. If you want to insert a new viewlet manager, you first have to make sure that the map viewlet ('collective.geo.kml.browser.viewlets.ContentViewlet') is registered for that manager (via ZCML).

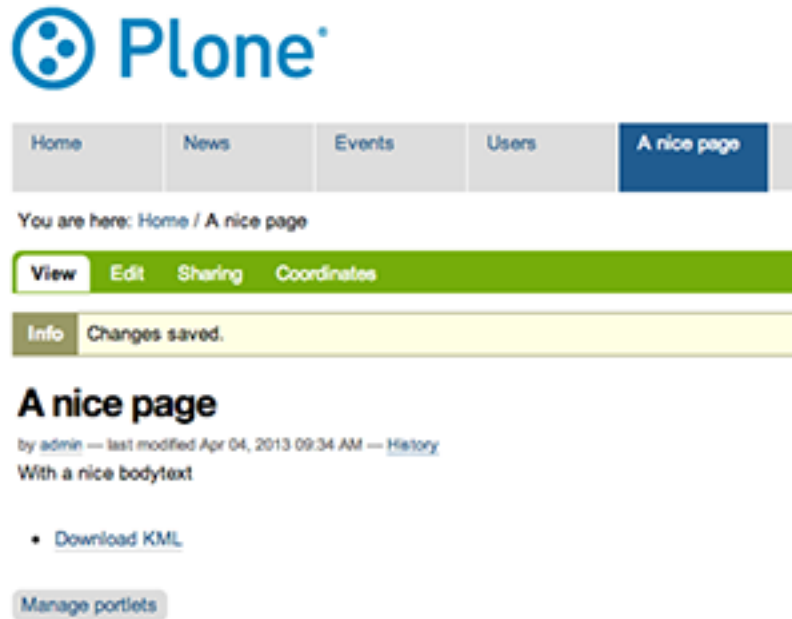
`plone.abovecontentbody|Above Content`
`plone.belowcontentbody|Below Content`

Bing API Code

Provide the Bing API code if you want to use the Bing Maps layer.

2.2 ... georeference a content type

After configuring collective geo you can georeference a content type by adding a new content type and click on Coordinates tab.



The Coordinates link will appear on the edit bar on content types that you've chosen as geo referenceable on collective geo settings form.

In the form that will load you can use the toolbar on the map to add a geographical shape like: a point, a line or a polygon.

Coordinates

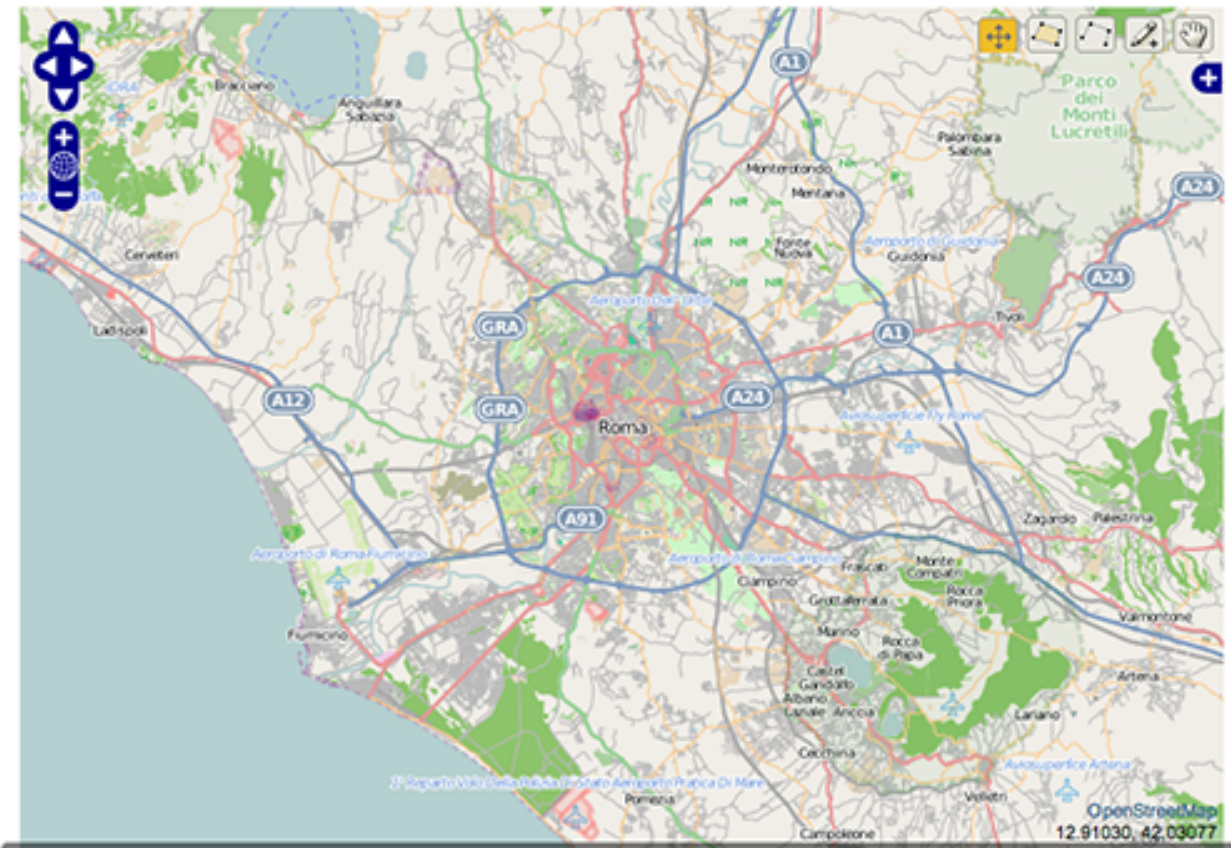
Specify the geometry for this content

Default CSV Import Custom styles

Find places

Coordinates

Choose a tool and draw the shape on the map



Manual coordinates

Save Cancel

Remove georeference

(polygon example)

Coordinates

Specify the geometry for this content

Default CSV Import Custom styles

Find places

Search

Coordinates

Choose a tool and draw the shape on the map



You can also use the text input on top of the map to search for a specific location and its coordinates.

View
Edit
Sharing
Coordinates

Coordinates

Specify the geometry for this content

Default
CSV Import
Custom styles

Find places

Select a location from the list below to set the coordinates.

- **Ciampino Province of Rome, Italy**

Coordinates

Choose a tool and draw the shape on the map

At the bottom of the map you'll find a textarea where you can manually put the coordinate in WKT format.

On the second tab of this form you can also override the default map styles for this content type.

View
Edit
Sharing
Coordinates

Coordinates

Specify the geometry for this content

Default
CSV Import
Custom styles

☐ Use custom styles?
Check this option if you want to customize the style of the geo features for this content

Marker image

The path to the image used as marker in maps. You can use either an absolute URL, or a TAL expression.

Marker image size

Choose the scaling factor of the marker shown on maps.

Line color

Choose the color for the line feature

Polygon color

Choose the color for the polygon feature

Line width

Choose the width of the line feature

Map position

Choose if and where the map will be displayed within the georeferenced content view.

Map width

Choose the width of the displayed map, specified as an absolute value (e.g. '450px' or '15em'), or relative (e.g. '100%') size.

Map height

Choose the height of the displayed map, specified as an absolute value (e.g. '450px' or '15em'), or relative (e.g. '100%') size.

Balloon details

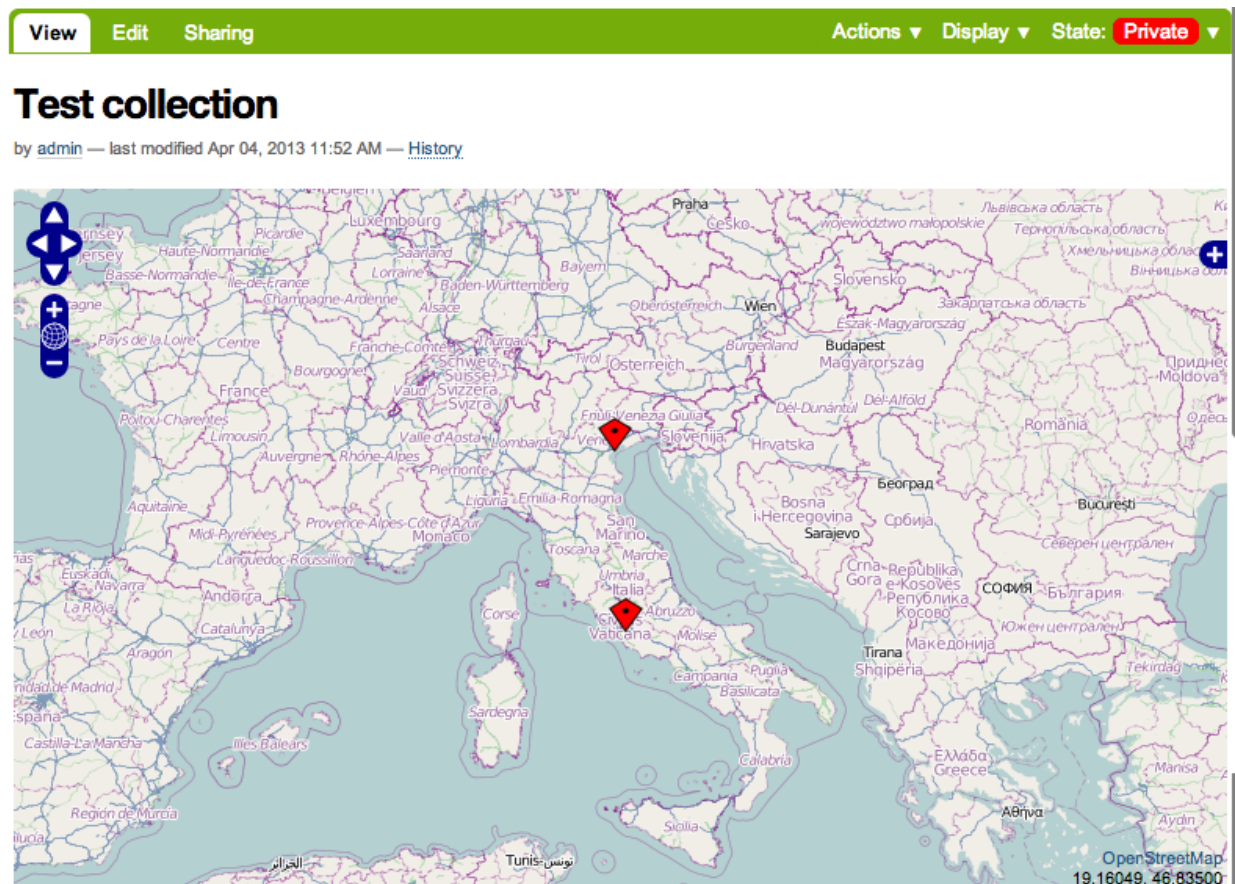
Choose which properties of the georeferenced content will be displayed in the additional informations section of the balloon.

<input type="checkbox"/> ID <input type="checkbox"/> Type <input type="checkbox"/> Subject <input type="checkbox"/> Content Location <input type="checkbox"/> Last Modified Date	<input type="checkbox"/> Title <input type="checkbox"/> Description
--	--

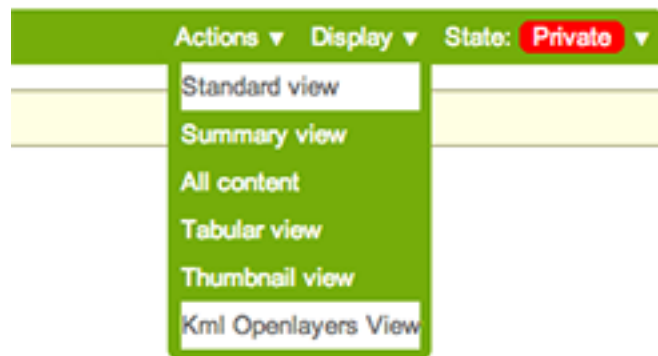
Save
Cancel
Remove georeference

After saving the form, the map will be displayed in the position you've chosen in the coordinates form or in the default position.

To remove the coordinates from a content type just click on the Coordinates tab and then click on the button Remove coordinates



After saving them you have to choose the item KML Openlayers view from the Display menu.



2.4 ...override a map layer template

All the maps in collective geo are composed by one or more map layers and each layer returns a javascript snippet by a page template which contains all information about the layer itself.

These page template are stored in Zope and can be customized by the portal view customization tool on the ZMI.

For example, the layer that defines the Google map layer is defined by this snippet:


```
function() {  
    return new OpenLayers.Layer.Google(  
        '%(title)s',  
        {  
            'sphericalMercator':true,numZoomLevels:20  
        }  
    );  
}
```

Into this javascript function you can use the OpenLayers API to create the layer correctly.

You can use the %(title)s variable to include the layer's title.

2.5 ...override KML file

Collective geo mainly uses a KML file and a Vector OpenLayers layer to display the maps.

This's useful for dividing javascript machinery that renders the map from geographical data stored in Zope.

This way we can change the data in KML file without having to modify the javascript or the map layers.

The KML file is defined by a Browser view class and a page template. This page template uses some macros to render many parts of it.

To override the default behaviour you can choose to override entirely the main KML template or only the macro template.

You can find all templates on portal view customization tool on the ZMI.

Core Packages

3.1 collective.geo.geographer

collective.geo.geographer has been documented on readthedocs.org: <http://collectivegeo.readthedocs.org/projects/collectivegeogeographer>

3.2 collective.geo.openlayers

This package includes `OpenLayers` framework in Plone.

You can include an `Openlayers` map in a page using a `BrowserView` or a `Page` template.

There is a simple integration example:

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:metal="http://xml.zope.org/namespaces/metal"
      xmlns:tal="http://xml.zope.org/namespaces/tal"
      metal:use-macro="here/main_template/macros/master">
<head>
  <metal:block metal:fill-slot="style_slot">
    <style type="text/css">
      #map {
        width: 100%;
        height: 500px;
        border: 1px solid black;
        position: relative
      }
    </style>
  </metal:block>

  <metal:block metal:fill-slot="javascript_head_slot">
    <script type="text/javascript">
      <!--
      (function ($) {
        var map;
        jq(window).load(function() {
          map = new OpenLayers.Map('map');
          var wms = new OpenLayers.Layer.WMS(
            "OpenLayers WMS",
            "http://labs.metacarta.com/wms/vmap0",
            {layers: 'basic'}
          );

```

```
map.addLayer(wms);
mousecontrol = new OpenLayers.Control.MousePosition(),
map.addControl(mousecontrol);

map.setCenter(new OpenLayers.LonLat(7, 45), 3);

});
}(jQuery));
// -->
</script>
</metal:block>
</head>
<body>
  <metal:content fill-slot="content-core">
    <metal:content define-macro="content-core">
      <div id="map" class="olMap">
        <!-- openlayers map -->
      </div>
    </metal:content>
  </metal:content>
</body>
</html>
```

For more informations about [OpenLayers](#) see:

- [Openlayers examples](#)
- [Openlayers User documentation](#)
- [Openlayers API](#)

3.3 collective.geo.settings

This package provides some utilities to store settings used in `collective.geo` packages.

Settings are stored in Plone registry and they provide default values to map widgets.

They are split in two different interfaces:

- `IGeoSettings` for map settings, like default map layers, center, zoom etc.
- `IGeoFeatureStyle` for map style like lines and polygon colors, marker image etc.

For more details see: [collective.geo.settings.interfaces](#)

You can retrieve settings by `plone.app.registry` API:

```
from zope.component import getUtility

from plone.registry.interfaces import IRegistry
from collective.geo.settings.interfaces import IGeoSettings
registry = getUtility(IRegistry)
settings = registry.forInterface(IGeoSettings)

from collective.geo.settings.interfaces import IGeoFeatureStyle
styles = registry.forInterface(IGeoSettings)
```

or using some utilities:

```
from collective.geo.settings.utils import geo_settings
settings = geo_settings()

from collective.geo.settings.utils import geo_styles
styles = geo_settings()
```

3.3.1 Coordinate field

collective.geo.settings defines a field type 'Coordinate' useful to define a coordinate attribute in Zope Interface and use it in forms.

You can define an interface in this way:

```
import decimal
from zope.interface import Interface
from collective.geo.settings.schema import Coordinate

class IGeoSettings(Interface):
    ...
    longitude = Coordinate(
        title=_(u'Longitude'),
        default=decimal.Decimal("0.0"),
        required=True
    )
```

3.4 collective.z3cform.mapwidget

Example:

```
from zope.interface import Interface
from zope import schema

from z3c.form import form, field
from collective.z3cform.mapwidget.widget import MapFieldWidget

class IMyForm(Interface):
    wkt = schema.Text(
        title=u"Shape in WKT format"
    )

class GeoShapeForm(form.Form):
    fields = field.Fields(IMyManager)
    fields['wkt'].widgetFactory = MapFieldWidget

    ...
```

Extra Packages

4.1 collective.geo.usersmap

This package provides a map showing the registered users of the portal by the location attributes of each user.

4.1.1 Dependencies

- Plone >= 4.2
- collective.geo.mapwidget > 2.0

4.1.2 Installation

This addon can be installed has any other addons, please follow official [documentation](#).

Add to buildouts configuration

```
[buildout]
...
eggs =
    collective.geo.usersmap
```

Re-run buildout, e.g. with

```
$ ./bin/buildout
```

Restart Plone and activate the product in Plones Add-on configuration section.

4.1.3 How it works

The “location” attribute in the user’s profile is automatically transformed in its corresponding coordinates, and then registered in a specific utility.

The Plone registry contains keys to manage the title and description showing on the map, and the list of user’s attributes to show in the map in the specified order.

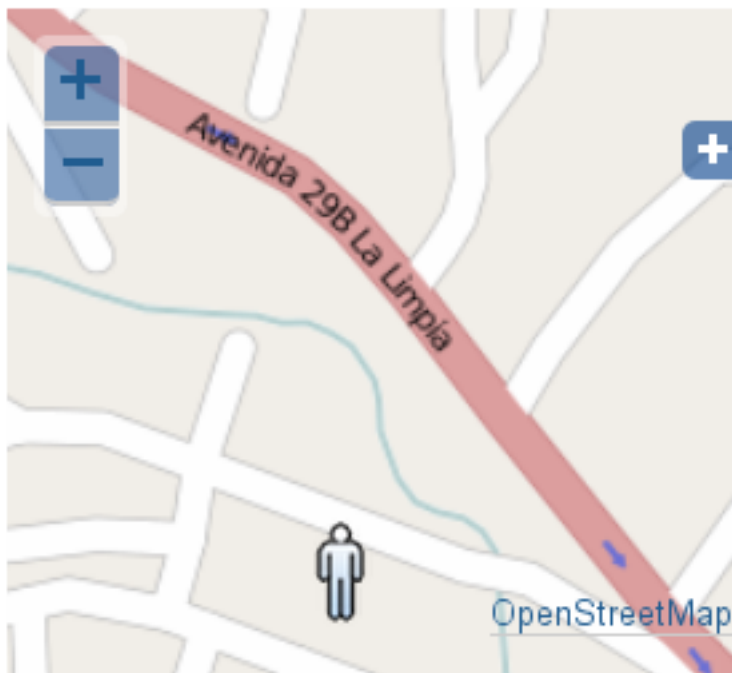
4.1.4 How to use the map

This package provides both a browser view at http://localhost:8080/Plone/@@usersmap_view.



Also this package provides a portlet named **usersmap portlet**.

Users' Map Portlet



4.1.5 Utils

A browser view at http://localhost:8080/Plone/@@usersmap_reindex is provided to let the admin reindex all the users' locations in case of any misalignment.

Optionally, you can define a crontab job to eventually run the update of users index.

4.1.6 Known bugs

You can set just one **usersmap portlet** per page.

4.1.7 Resources

- Code repository: <https://github.com/collective/collective.geo.usersmap/>
- Questions and comments to <http://www.coactivate.org/projects/collectivegeo/lists/collectivegeo-discussion/>
- Report bugs at <https://github.com/collective/collective.geo.usersmap/issues>

4.2 collective.geo.flexitopic

Use `collective.geo.flexitopic` to easily build interactive maps out of plone collections. It combines plone maps (`collective.geo`) with `collective.flexitopic`.

4.2.1 collective.flexitopic

Flexitopic integrates the easy use of plone collections with a `Flexigrid` AJAX view. The criteria from the topic are taken to construct a simple query form to narrow down a search inside a collection. For old style collections subtopics are displayed inside tabs of the collection.

- Flexitopic does not install a new content type but just adds an additional view to the collection type.
- it degrades for non javascript browsers to a simple table - (almost) same usability, no information loss.
- it requires JQuery only (built into plone 4) no JQueryUI
- **lightweight JS**
 - `Flexigrid`: 24 KB packed
 - `JSlider`: 15 KB packed

4.2.2 Usage

Add a collection. The Criteria of the collection will be used to build a form to narrow down your search inside the collection. If the criterion (the index in `portal_catalog`) is sortable you can sort this column. Not all criteria types can be used as input for `Flexicollection` so beware.

- Search Text: full text search inside the collections. If you leave the criterion value empty users can search for content containing that text, if the value is not empty it will search for this text plus the text the user supplied.
- Title: search or sort by title (see above)
- Description: search description only (search see above, no sorting here!)

- Dates (effective, created, ...): will be converted to date ranges and can be selected with the JQuery [JSlider Plugin](#))
- Location (path index): will not be displayed in the search form and always be applied to the query
- Keyword Indices (like tags): a drop down list will be generated to narrow the search down
 - if the criterion operator is AND the list will contain all unique values of the index minus the ones you selected, the query will search for all terms that match your criteria plus the user input (this only applies to old style collections)
 - if the criterion is OR the terms you selected will be display in the selection list. The search will be for the user supplied input only. This is the only available behaviour for new style collections

The output is always a table with the fields you supplied in the ‘Table Columns’ of the collection, no matter if ‘Display as Table’ is checked or not.

4.2.3 Subtopics (old style Collections only):

Flexitopic will display subtopics as tabs on top of the page. The first tab is the description of the topic, subtopics will occupy the following tabs. Subtopics will always be displayed as (plain html) tables defined by the criteria, ‘Table Columns’ and the ‘Number of Items’ of the subtopic.

4.2.4 Installation

This addon can be installed has any other addons, please follow official [documentation](#).

Add to buildouts configuration

```
[buildout]
...
eggs =
    collective.geo.flexitopic
```

Re-run buildout, e.g. with

```
$ ./bin/buildout
```

Restart Plone and activate the product in Plones Add-on configuration section.

4.2.5 Resources

- Code repository: <https://github.com/collective/collective.geo.flexitopic/>
- Questions and comments to <http://www.coactivate.org/projects/collectivegeo/lists/collectivegeo-discussion/>
- Report bugs at <https://github.com/collective/collective.geo.flexitopic/issues>

Overview

collective.geo allows to geo-reference Plone content types and to display this information over a map.

The core of collective geo is composed by the following packages:

collective.geo.geographer provides geo annotation for Plone. ([repository](#))

collective.geo.openlayers enables openlayers machinery into Plone. ([repository](#))

collective.geo.settings provides some utilities where settings of collective.geo packages can be stored ([repository](#))

collective.geo.mapwidget provides some handy page macros and adapters to easily manage multiple maps on one page. ([repository](#))

collective.z3cform.mapwidget provides a mapwidget for z3c.form framework. ([repository](#))

collective.geo.contentlocations provides a GUI for collective.geo.geographer. It provides some simple forms to add geographical coordinates to Plone content types. ([repository](#))

collective.geo.kml provides KML views for georeferenced objects, allowing Plone content types to be visualized in Google Earth. ([repository](#))

To display the maps, **collective.geo** takes advantage of [Openlayers](#), a JavaScript library for displaying map data in web browsers, with no server-side dependencies.

As a default map source we can select [OpenStreetMap](#), [Google Maps](#) or [Bing Maps](#).

Indices and tables

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