

Deliverable

Project Acronym: LoCloud

Grant Agreement number: 325099

Project Title: Local content in a Europeana cloud

D3.2 Geocoding Enrichment Services

- Geolocation API (LoGeo API)
- Geocoding application

Revision: Version 1

Authors:

Franc J.Zakrajsek, IPCHS
Vlasta Vodeb, UIRS
Jurij Stare and Andrej Grilc, Grangeo
Stein Runar Bergheim, Asplan Viak Internet AS (AVINET)

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	x
C	Confidential, only for members of the consortium and the Commission Services	

Revision History

Revision	Date	Author	Organisation	Description
0.0	03.02.2014	Franc J.Zakrajsek	IPCHS	Draft
0.1	19.08.2014	Franc J.Zakrajsek Vlasta Vodeb Jurij Stare Andrej Grilc	IPCHS UIRS Grangeo Grangeo	Complete section on geocoding API
0.2	22.08.2014	S.R. Bergheim	AVINET	Incorporated geocoding tool documentation
0.3	26.08.2014	Franc J.Zakrajsek Vlasta Vodeb Jurij Stare Andrej Grilc S.R. Bergheim	IPCHS UIRS Grangeo Grangeo AVINET	For internal review
0.4	28.08.2014	Franc J.Zakrajsek Vlasta Vodeb Jurij Stare Andrej Grilc S.R. Bergheim	IPCHS UIRS Grangeo Grangeo AVINET	Revised draft, complete for internal review
1.0	03.09.2014	Franc J.Zakrajsek S.R. Bergheim	IPCHS AVINET	Final Version 1

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Contents

1. Executive Summary	4
2. Introduction	5
Overview of geolocation enrichment tools.....	5
Overview of the development methodology	6
3. Getting started	9
Geolocation API	9
Geocoding application	10
4. Geolocation API reference	12
Request.....	12
Response	13
HTML Status Codes	14
Invoking the geolocation API programmatically	15
5. Geocoding application user documentation	17
Overview of the user interface	17
Executing a geocoding project, step by step	18
6. How to install the geolocation enrichment tools	32
Geolocation API	32
Geocoding application	32
7. How the the tools are installed in LoCloud	34
Geolocation API	34
Geocoding application	34
8. References	35
9. Glossary	37

1. Executive Summary

Geolocation is an important piece of information that makes it possible to facilitate both search/retrieval and exploration of cultural heritage content. The majority of heritage metadata contains some form of implicit or explicit geographical reference but more commonly than coordinates are indirect textual or formal references such as addresses or geographical names. In order to achieve the full benefits of spatial metadata it is necessary to have access to map coordinates that can be used to visualize the location of a resource on a map – or that may be used to infer meaningful relationships between two or more resources based on the proximity of their spatial metadata.

This deliverable, the LoCloud Geolocation Enrichment Services addresses this weakness in existing cultural heritage data and introduces two different interoperable tools to facilitate enrichment of existing cultural heritage metadata as well as enabling geographical names resolution into coordinates from literally any third-party application through a geolocation API.

Geolocation API

LoGeo is a geolocation API (Application Programming Interface), one of the Geolocation Enrichment Tools and has been developed within WP3 in the LoCloud project. The purpose of the LoGeo API is to resolve a given search term into one or more recognized place names candidates accompanied by geographical coordinates.

The LoGeo API is specially designed for the recognition of the place name from metadata of the cultural heritage and could be simply integrated and used in the other API-s, microservices, and applications. The usability of the LoGeo API may easily be tested online: http://locloudgeo.eculturelab.eu/Console_LoGeo_1_1_m/

Geocoding application

The purpose of the geocoding application is to enable local heritage professionals to execute crowd-sourcing projects to enrich non-spatial metadata records with geographical locations.

Source data to be geocoded can be imported from either the MORE repository, on the way from the local institution to Europa, or it can be imported as CSV files from any local collection management system that is capable of representing content as records. Records can be geocoded by means of manual location in the map or by searching several main-stream geolocation APIs including, optionally, the LoGeo geolocation API. Geocoded data can be exported back into a variety of popular usage formats that will facilitate their uptake and use in web applications as well as loading enriched data back into the authoritative collection management system.

The geocoding application can be tested by pointing your web browser to the following URL: <http://locloud.avinet.no/demo>

2. Introduction

Overview of geolocation enrichment tools

In recognition of the importance of geography in creating meaningful relationships between independent pieces of cultural heritage content, LoCloud has developed a set of tools and services that helps owners and custodians of cultural heritage collections to add geographical metadata, in the form of spatial coordinates, to their existing content.

These tools fill a void in existing software infrastructures for geocoding that are either (1) high-end commercial offerings targeted at professional GIS users or (2) open source desktop/command-line/terminal/shell applications that are best suitable for expert users with very particular software environment requirements.

The LoCloud geocoding enrichment tools consists of two separate components that optionally can be configured to work together, i.e. the geocoding application can consume the geolocation API services:

1. LoGeo: Geolocation API
2. Geocoding application

LoGeo: geolocation API

LoGeo is a geolocation API (Application Programming Interface), one of the Geolocation Enrichment tools. The API was developed within WP3 in the LoCloud project by IPCHS (Institute for Protection of Cultural Heritage of Slovenia) and Grangeo Ltd.

The purpose of LoGeo API is to recognize (NER-Name Entity Recognition) a given place name and return one or more recognized place names candidates accompanied by geographical coordinates.

The LoGeo API is specifically designed for the use of the place name recognition of cultural heritage metadata. The efficiency of LoGeo API is tested with Europeana EDM collections.

The LoGeo API may be invoked directly by any of the other microservices (WP3) including the Metadata Enrichment (Task 3.3) and Historic place names (Task 3.4). The LoGeo API could be also simply implemented in the cultural heritage management systems/repositories used by museums, libraries, archives and other cultural institutions.

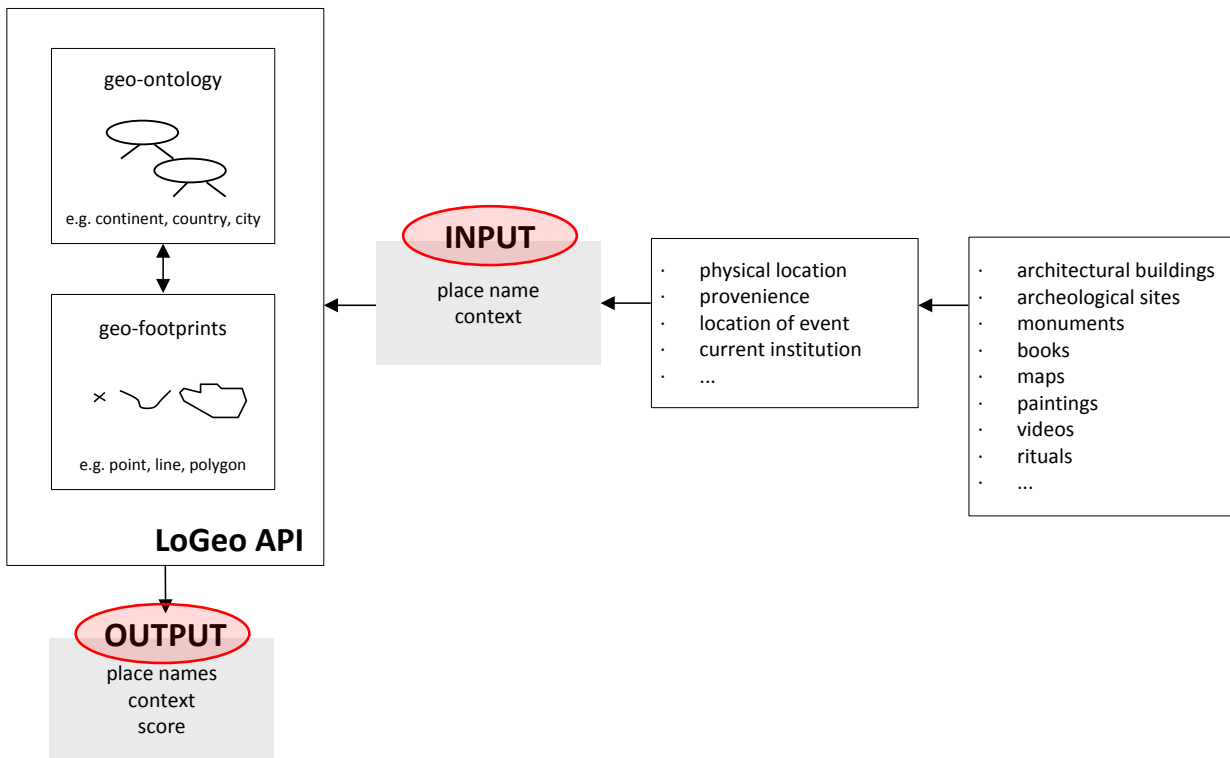


Figure 1: LoCloud API scheme

Geocoding application

The geocoding application offers a simple, map-centric user interface that permits geocoding of any records-based content, or content that may be represented as records, requiring no other software tools installed on their computer than a free, simple, mainstream web browser.

The application may be configured to act as a front-end to the LoGeo geolocation API but can equally well act as a stand-alone software installation.

While a number of user-friendly end-user applications for geocoding tasks exists, for an example the Google Maps Composer, no existing solutions permits users to take existing data as a starting point for the geocoding work. Instead, they require users to build new datasets from scratch based on a simple, but limited, point of interest data model.

The geocoding microservice client is not meant to replace inhouse collection management systems but to integrate flexibly with other software designed to fit into existing business process IT applications. For this reason, the user interface is kept very clean and simple. It offers a wide range of external interfaces through import of CSV data that permits access to data held literally any external system - as well as export of geocoded data back into popular and flexible usage formats such as CSV, JSON, KML and RDF.

Overview of the development methodology

Geolocation API

LoGeo API is developed on the basis of up-to date methods of spatial reasoning and several applications as are Stanford NER (21), Europeana Geoparser (25), Google API, Geonames (9) and mostly on authors experiences when developing GIS tools and applications gained in the EU project

as are Athena (28), Carare, Indicate (12) and other projects (14, 15). In the process of developing the LoGeo API also the actual state of the LoCloud collections has been taken into account (see Appendix).

The effectiveness of the place name recognition depends on the use of geospatial reasoning methods and on used place names databases. LoGeo API 1_1 is currently searching among more than 12 million place names. The gazettters are three types:

- Global gazetter (geonames)
- National gazettters of settlemets (including also small settlements) and other geographical places
- Cultural gazettters of architectural and archeological sites.

LC_Geonames:	No. of places:
Geonames	9.034.306
LC_National:	No. of places:
Slovenia	14.302
Norway	1.027.824
Finland	808.258
Spain	1.089.091
Poland	161
LC_Cultural:	No. of places:
LC_Cultural	344.702

Geocoding application

The system requirements specification, system design documents and practical implementation of the geocoding microservice client has been the responsibility of LoCloud partner Asplan Viak Internet AS (AVINET).

The main input to the system requirements specification was the outputs from a focus-workshop held during the LoCloud kick-off meeting in Oslo (March 2013). The outcomes of this workshop were translated into a formal specification list that identified functional requirements, assigned their priority and added them to a work-log in preparation for the practical implementation.

The critical success factors in identifying functional requirements were to achieve ease of use from the perspective of LoCloud content providers, i.e. small and medium-sized local and regional cultural heritage institutions with limited capacity for learning and spending time using advanced software applications.

The absolute technical requirement that were dimensioning for the work was that the end-user application must be capable of running as a SaaS cloud service from any common hosting provider.

It was determined that an application capable of running on any WIMP/WAMP/LAMP¹ platform would be preferable. This will permit users to run the geocoding microservice on any web host after the project is over, thus contributing to the long-term sustainability of project results.

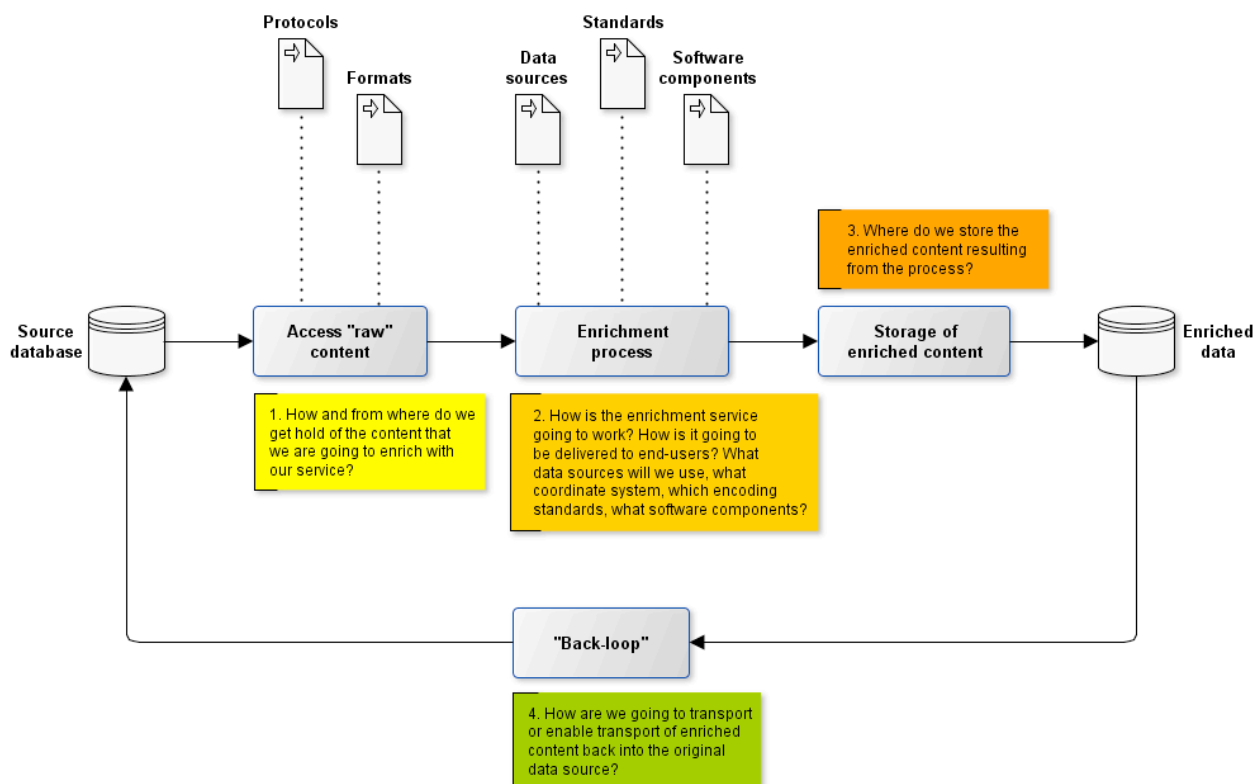


Figure 2: High-level data flow specification for geocoding application

Having identified the requirements, a simple system-design document was created, and subsequently presented to and discussed with the LoCloud consortium during the project plenary meeting in London (December 2013). Along with a discussion of the system design, a working prototype was presented to the same audience in order to make it easier for partners to contextualize and visualize what type of microservice might result from the project. Based on the practical feedback from this meeting, the implementation process commenced.

The implementation process followed the SCRUM software development methodology where the requirements from the SRS were grouped into so called “sprints”, or iterations, each of which resulted in an incrementally feature rich release of the software application.

The SCRUM process defines a number of roles and responsibilities that were assigned and divided among AVINET development staff. The two most important roles are those of “SCRUM master” and “product owner”. Together, these two roles are responsible for making sure that the output from the implementation process satisfies the requirements in the functional specifications as well as ensuring that the development process itself runs smoothly and that all resulting software components are made subject to rigorous testing in order to identify bugs and performance issues.

¹ W=Windows, A=Apache, M=MySQL, P=PHP, L=Linux, I=Internet Information Server

3. Getting started

Geolocation API

The LoGeo API is accompanied by the LoGeo API console:
http://locloudgeo.eculturelab.eu/Console_LoGeo_1_1_m/

When a user points their web browser to the URL of the LoGeo API console, he or she may simply press the submit button and immediately gain an insight into how the geolocation API works. Invoking the default query will return the geographical representation of the placenames matching the query “Paris”.

The console is intuitive and simple to use. The user submits *Input text* (place name, e.g. Paris), *Context* (spatial limit of the place name as country, region, continent, e.g. Europe), *Country* (limit results on Country, e.g. France), *MaxOutput* (limit the number of results) and chooses the *PreferableSource* (Geonames, National or Google).

The user can unlimitedly use the console and submits queries with the place name on her/his choices. Fig. 2 displays the example of small town “Silo de Cadillo”.

The Lo GeoAPI console is not only a “getting started tool”, but also an excellent tool for testing and designing the implementation of LoGeo API in a specific user environment - and for learning geo-spatial reasoning rules.

The screenshot shows the LoGeo API console interface. On the left, there is a 'Parameters' section with the following fields: 'InputText' containing 'Silo de Cadillo', 'Context' set to 'Spain', 'Country' set to 'Spain', 'MaxOutput' set to '10', and 'PreferableSource' set to 'Geonames'. A 'submit' button is located below these fields. The main area displays a map of central Spain with a red pin marking the location of Silo de Cadillo near Villacañas. Below the map, the 'Request URL' is shown as: `http://locloudgeo.eculturelab.eu/LoGeo_1_1/loGeo.aspx?InputText=Silo%20de%20Cadillo&ContextPlace=Spain&Country=Spain&MaxOutput=10&PreferableSource=Geonames&Key=xxxxxxx`. The 'JSON response' is displayed as a JSON object:

```
{
  "LoGeo version 1.1 07.02.2014": {
    "InputText": "Silo de Cadillo",
    "PlaceName": "Silo de Cadillo",
    "AlternativeNames": "",
    "ContextOut": "Villacañas, n.a. (254), Toledo, Castilla-La Mancha, Spain",
    "ContextTime": "",
    "CoordinateSystem": "EPSG:4326",
    "PlaceX": -3.4332283,
    "PlaceY": 39.605286,
    "PlaceZ": 0.0,
    "AccuracySpatial": "",
    "Confidelity": "15.283072",
    "Rights": "GeoLoOnto",
    "Source": "GeoLoOnto",
    "Date": "2014-06-23"
  }
}
```

Figure 3: LoGeo API console

Geocoding application

The geocoding application is a SaaS service, installed into several cloud based web hosting environments.

To get started you can simply visit the URL: <http://locloud.avinet.no/demo> where the latest development version of the service always is available for testing.

If you would like to install the application locally on your own (virtual) hosting environment you can follow the step-by-step installation instructions in chapter 5 below.

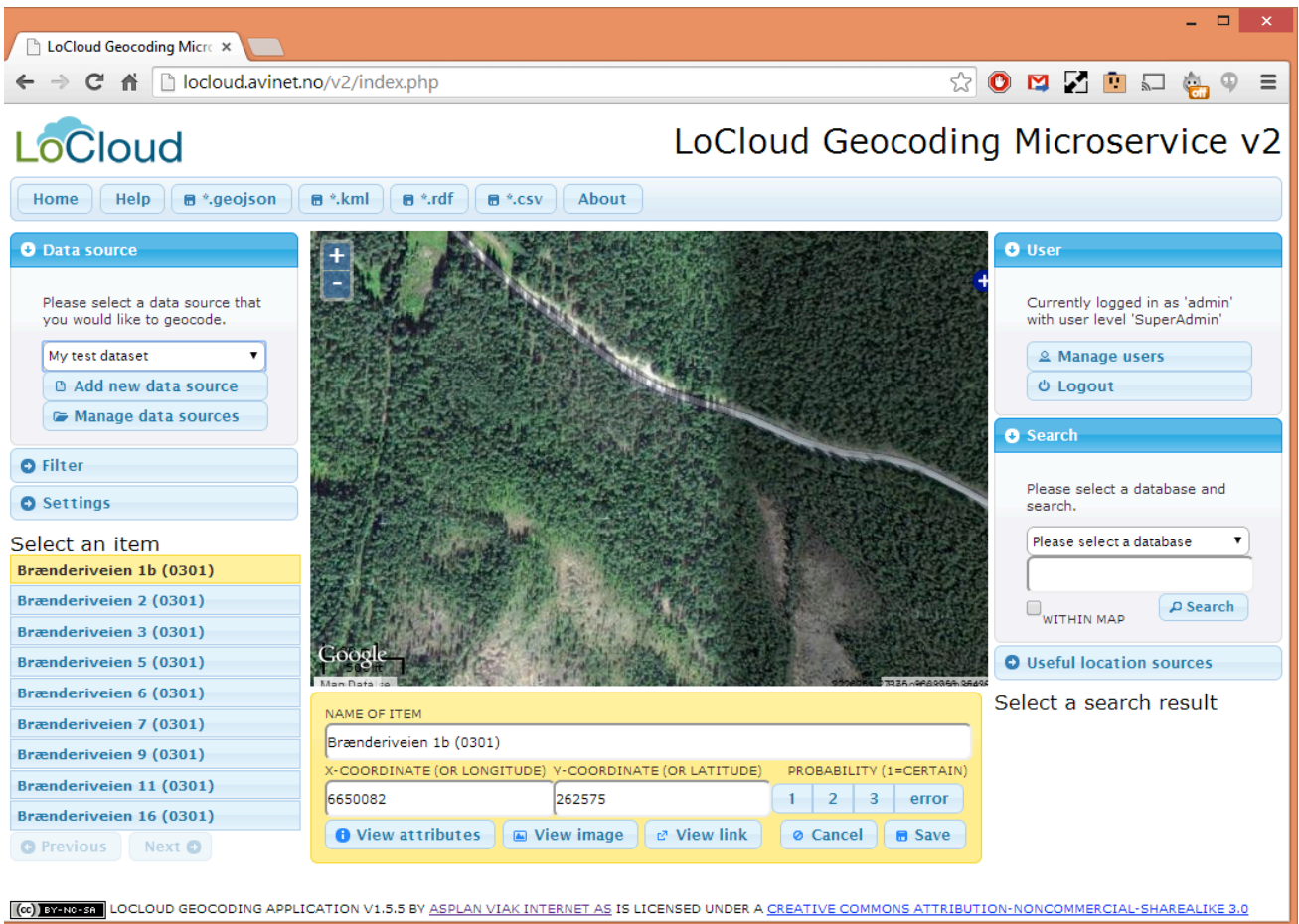


Figure 4: Geocoding application user interface

4. Geolocation API reference

The geocoding microservice API consists of a single generic geographical names search method that returns a simple response format that is common to all geographical names source that are accessible through the API.

Request

Method	URL
GET	Example request: <code>http://locloudgeo.eculturelab.eu/LoGeo_1_1/loGeo.aspx?InputText=Ljubljana&ContextPlace=Slovenia&Country=Slovenia&PreferableSource=Geonames&MaxOutput=10&Key=xxxxxxx</code>

Parameter	Datatype	Description
InputText	String	Place name (i.e. Ljubljana). Required.
ContextPlace	String	Geographic context (i.e. Slovenia). Not Required.
Country	String	Country (i.e. Slovenia). Not Required.
MaxOutput	Integer	Output will contain maximum of specified results. If not specified then MaxOutput=1. Not Required.
PreferableSource	String	Preferable lookup source will be used as priority. If not specified then PreferableSource=Geonames. Not Required. Possible values: 1. »Geonames« - lookup using Geonames API 2. »Google« - lookup using Google Places API 3. »National« - lookup using the National database
Key	String	Key for LoGeo API usage. Required.

Response

Status	Response
200	<p>Example response (standard JSON):</p> <pre>{ "LoGeo version 1.1 07.02.2014" : [{ "InputText" : "Ljubljana", "PlaceName" : "Ljubljana", "AlternativeNames" : "Aemona,Gorad Ljubljana,Jubaljana,LJU,Laibach,Laiboch,Lioumpliana,Liubliana,Liubliane,Liublijana", "ContextOut" : "Osrednjeslovenska,Slovenia", "ContextTime" : "", "CoordinateSystem" : "EPSG:4326", "PlaceX" : 14.50513, "PlaceY" : 46.05108, "PlaceZ" : 295.0, "AccuracySpatial" : "", "Confidelity" : "185.93762", "Rights" : "Creative Commons Attribution 3.0", "Source" : "Geonames", "Date" : "2014-04-27", "Remarks" : "" }] }</pre>

Parameter	Datatype	Description
InputText	String	Place name (i.e. Ljubljana) input
PlaceName	String	Place name (i.e. Ljubljana) output
AlternativeNames	String	Alternative place names separated by comma (i.e. Ljubljana, Lyublyana)
ContextOut	String	Geographic context (i.e. Osrednjeslovenska,Slovenia)
ContextTime	String	Time context (i.e. 19th century)
CoordinateSystem	String	Coordinate system (i.e. EPSG:4326)
PlaceX	Double	X-axis coordinate

PlaceY	Double	Y-axis coordinate
PlaceZ	Double	Z-axis coordinate
AccuracySpatial	Integer	Spatial accuracy of the location
Confidelity	Double	Record rank among results
Rights	String	Copyright notice
Source	String	Source of the lookup
Date	Date	Date associated with a record
Remarks	String	Remarks associated with a record

HTML Status Codes

All status codes are standard HTTP status codes. The geocoding microservice API uses the following status code for all requests. Any information about the status of the search operation is embedded in the valid JSON object that is returned from the Web Service.

Status Code	Description
200	OK
201	Created
202	Accepted (Request accepted, and queued for execution)
400	Bad request
401	Authentication failure
403	Forbidden
404	Resource not found
405	Method Not Allowed
409	Conflict
412	Precondition Failed
413	Request Entity Too Large
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

Invoking the geolocation API programmatically

Below are the programming snippets to help the implementing of LoGeo API in different programming environments.

LoGeo API may be invoked using any common programming environments including but not limited to C#, VB.NET, JAVASCRIPT, JAVA, PHP.

C#

```
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;
using System.Net;
using System.Text;

...

using (WebClient webClient = new WebClient())
{
    webClient.Encoding = Encoding.UTF8;
    string JSONString =
    webClient.DownloadString("http://locloudgeo.eculturelab.eu/LoGeo_1_1/loGeo.aspx?InputText=Paris&ContextPlace=France&Country=France&MaxOutput=10&PreferableSource=Geonames&Key=xxxxxxx");
    JObject JSONObject = JObject.Parse(JSONString);
}
```

VB.NET

```
Imports Newtonsoft.Json
Imports Newtonsoft.Json.Linq
Imports System.Net
Imports System.Text

...

Using webClient As New WebClient()
    webClient.Encoding = Encoding.UTF8
    Dim JSONString As String =
    webClient.DownloadString("http://locloudgeo.eculturelab.eu/LoGeo_1_1/loGeo.aspx?InputText=Paris&ContextPlace=France&Country=France&MaxOutput=10&PreferableSource=Geonames&Key=xxxxxxx")
    Dim JSONObject As JObject = JObject.Parse(JSONString)
End Using
```

JAVASCRIPT

```
var jsonReq = new XMLHttpRequest();
jsonReq.open("GET",
"http://locloudgeo.eculturelab.eu/LoGeo_1_1/loGeo.aspx?InputText=Paris&ContextPlace=France&Country=France&MaxOutput=10&PreferableSource=Geonames&Key=xxxxxxx",
false);
jsonReq.send();
var JSONString = jsonReq.responseText;
var JSONObject = JSON.parse(JSONString);
```

JAVA

```
import org.json.simple.JSONObject;
import org.apache.commons.io.IOUtils;

...

String JSONString = IOUtils.toString(new
URL("http://locloudgeo.eculturelab.eu/LoGeo_1_1/loGeo.aspx?InputText=Paris&ContextPlace=France&Country=France&MaxOutput=10&PreferableSource=Geonames&Key=xxxxxxx"), "UTF-8");
JSONObject json = new JSONObject(JSONString);
```

PHP

```
$contents =
file_get_contents('http://locloudgeo.eculturelab.eu/LoGeo_1_1/loGeo.aspx?InputText=Paris&ContextPlace=France&Country=France&MaxOutput=10&PreferableSource=Geonames&Key=xxxxxxx');
$results = json_decode($contents, true);
```


5. Geocoding application user documentation

Overview of the user interface

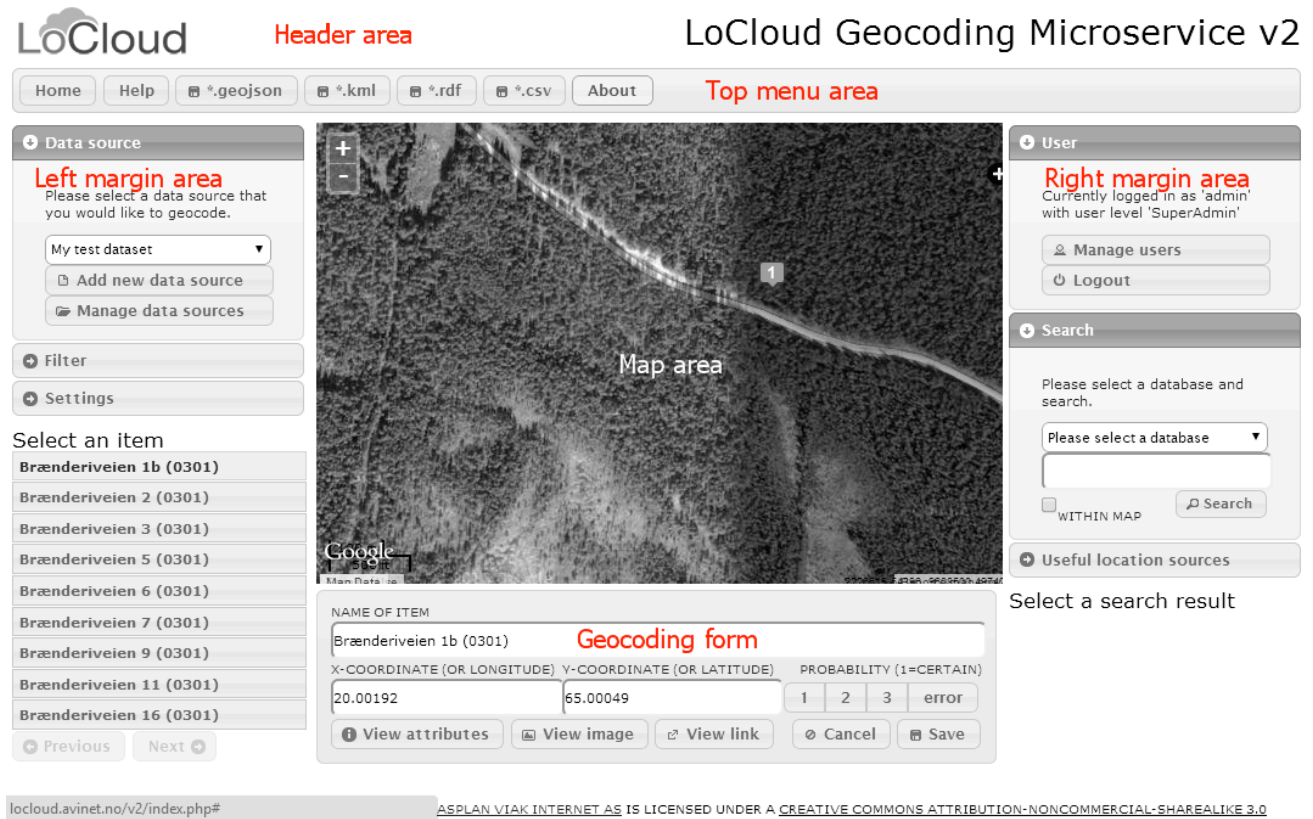


Figure 5: Overview of the geocoding application user interface

The following are the main components of the user interface. The screen is divided into seven *areas*. Some of these are further subdivided into collapsible sections called *panels*. The areas and panels are as follows:

- The *header area* contains a logo and the site title
- The *top menu area* contains menu buttons for navigation, information, download of data - and for logging out
- The *left margin area* is located on the left side of the screen
 - The *settings panel* contains a slider that permits us to set the default zoom
 - The *data source panel* contains functions to select and filter data sources to be geocoded
- The *map area* occupies the upper part of the main content area
- The "geocoding form" occupies the lower part of the main content area
- The *view and edit attribute form* is a modal window that can be activated from the *geocoding form*
- The *right margin area* is located on the right side of the screen

- The *search panel* contains fields to select a database, to limit the search and to enter a search expression
- The *useful location sources panel* allows you to open various external applications zoomed in on the same area to see if they contain hints as to where your *item* is located.

Executing a geocoding project, step by step

This section contains step-by-step instructions on how to execute a geocoding project using the LoCloud geocoding application.

Step 1: Authentication

This section describes the functions of the authentication module of the geocoding-microservice client application.

Login

When entering the application via a web browser, it will automatically challenge the user for username and password before loading the application.

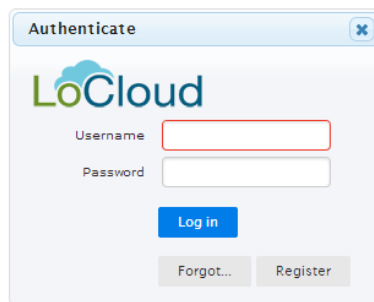


Figure 6: Geocoding application login prompt

Register new user

New users can be registered either from:

- the login prompt by clicking the “Register” button or by users who are members of the “editor” role or higher.
- the user panel in the right-margin area of the user interface by clicking the “Manage users” button.

Self-registered users will always be given the role “user” and must be added to a geocoding project by an editor – or promoted to a “higher” role by an authorized user. Users registered by existing users can be given the same or lower roles as the user doing the registration.

Register new user

Add user details

Username

Password

Here you can register for a new user account.

By default you will be assigned rights as a viewer, that means you can contribute to existing crowd-sourcing datasets that are already in the system and made available to you.

If you would like to upload your own data you must contact the system administrator to make her or him elevate your account to editor rights

Figure 7: Self-registration for new users

Manage existing users

In order to use the application it is necessary to have a user account. At this time, the application is not configured to permit public registration of new users. Users have to be created by existing users who are assigned the role of either “SuperAdmin”, “Admin” or “Editor”.

Manage users

1a) Create a new user

...or...

1b) Select an existing user to edit

2. Add or modify user data

User name

Password

Level

ID#	User name	User level
1	admin	Super administrator
3	guest	Guest user
9	gunnar	Administrator
25	h-gaarder	User
28	h-gaarder/2014	User
24	hoseth	User
26	o-nokleby	User
27	t-gaarder	User
2	user	User

Server Response 0.299 sec 1-9 of 9 (buffered 9)

Figure 8: Interface for management of existing users

Step 2: Working with data sources

Creating a new data source

In order to do geocoding, you first have to upload a file that you would like to add coordinates to. This is a very simple procedure that involves selecting a CSV file from your computer and specifying some simple metadata.

New datasource

Select a file to upload Add metadata Finish

You can upload any CSV text files including files created in Microsoft Excel by choosing "File-->Save as..." and then select one of the formats starting with "CSV..." in the dropdown box "Save as type".

Select file to upload...

Delimiter character Semicolon

Character encoding ISO-8859-1

Field names in first row

100%

Successfully uploaded file: 20140506 - Demo Dataset 2.csv.
Please wait while parsing the file and inserting into database. Once completed, the wizard will automatically proceed to the next step.

Next

Figure 9: Step 1 of the new datasource wizard

New datasource

Select a file to upload Add metadata Finish

Data source name LoCloud demo dataset v2

Unique identifier id

Persistent URI <not set>

Display field name

Spatial reference system LatLong, WCS 84

Coordinate precision * 6

X-coordinate x

Y-coordinate y

Category category

Area division level 1 country

Area division level 2 area

Web link field url

Figure 10: Step 2 of the new datasource wizard

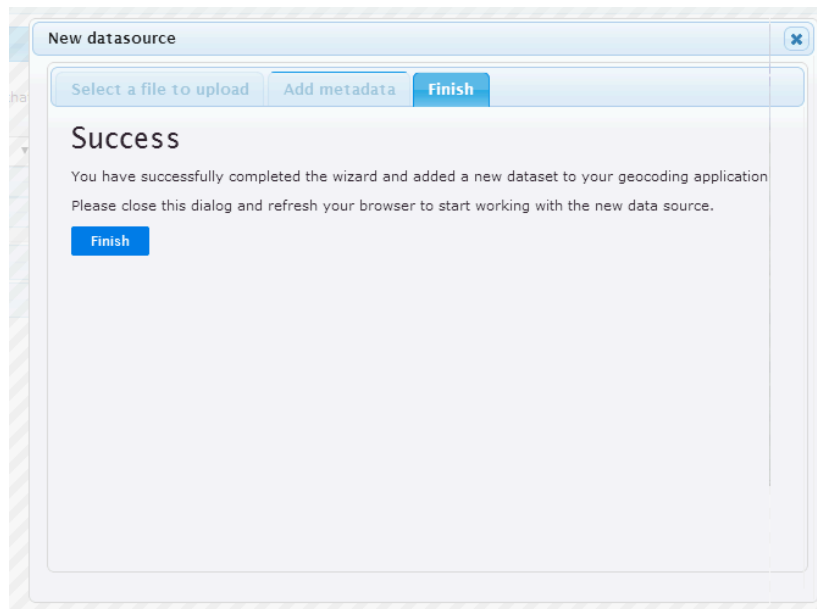


Figure 11: Step 3 of the new datasource wizard

1. Create a CSV (comma separated variables) file with the information that you would like to geocode. If your application cannot export such a file directly, you can easily create one using Microsoft Excel by choosing, File --> Save as... and choose "Comma Separated Variables" as file type.
2. Open the geocoding application and log in with your user account
3. Choose the menu option "New datasource..."
4. Select the file you'd like to upload using the "Select file..." dialog
5. Select which column in the table (if any) that contains the:
 1. unique ID of the dataset (mandatory)
 2. name of the *item* (mandatory)
 3. category *item* (optional)
 4. 1st level area division (optional)
 5. 2nd level area division (optional)
 6. existing X-coordinate column (optional)
 7. existing Y-coordinate column (optional)
 8. spatial reference system code
 1. The default coordinate system is WGS1984 geographical coordinates, specified by the keyword (EPSG:)4326, other coordinate systems can be specified. If your desired spatial reference system is not in the drop-down, you can request it to be added by contacting AVINET

- The value you select here must correspond to the spatial reference of existing coordinates, if you have any. If you have a data source without existing coordinates, you can specify any coordinate system here.

6. Click the *Upload file* button

7. If there are any error messages, please correct the issues highlighted and try again.

Manage data sources

Once a data source has been created it can be edited, shared and deleted by clicking the manage data sources button in the data source panel in the left-margin area of the user interface.

1. Select a data source

Data source title

- 2 gater i Oslo
- Admin's private datasources (1)
- Brænderiveien 3**
- Ft1910 (1554)
- Ft1910 (eksport-test frå desktop app)

2a) Edit data source (owners only)

Title: Brænderiveien 3

Primary key col.: id_1

Persist URI col.:

Name column: display_name

Image column: digark_uri

URL column: <not set>

X/Longitude: coordinate_y

Y/Latitude: coordinate_x

Update data source

Delete data source

2b) Edit access control (owners, editors)

Access level: Editor

User: Type a couple of letters of the user name...

Add to access control list

User name	Access
admin	Owner

Remove selected user from access control list

Done

Close

Figure 12: Dialog for management of datasources

A data source can be updated to specify which attributes have what meaning as per the pre-defined field types recognized by the geocoding application.

Furthermore, it is possible to determine which users shall have access to which data source and what level of rights they shall enjoy whether users, editors or administrators.

Select and filter your data source

At this stage you have uploaded your data source and need to come to terms how the application user interface works. The first thing to understand is the data source panel.

The first element in the data source panel is a drop-down box that permits you to select which data source you want to work with.

- To proceed, please select a data source from in the drop down box with the label *Please select a source*

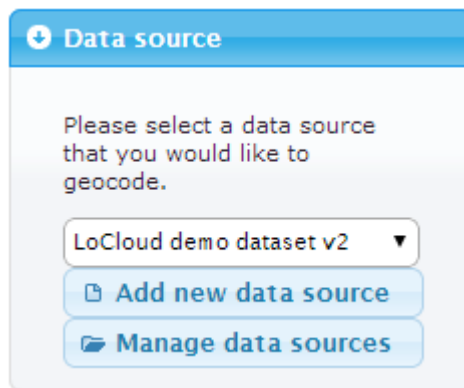


Figure 13: Data source panel

When you select that data source, two additional drop-down boxes appears in the *filter panel*.

1. Filter by areas
2. Filter by category

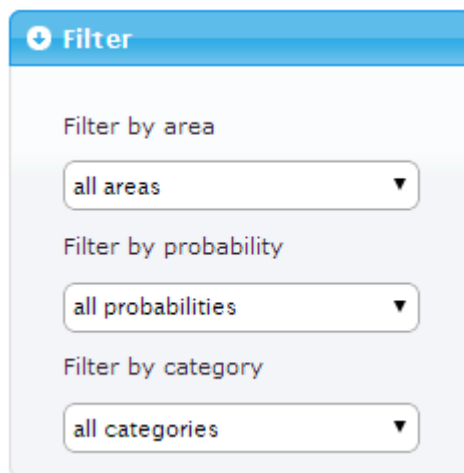


Figure 14: Filter panel

In addition there was already one drop-down box in place: Filter by probability. The latter is better understood after going through the functions of the *geocoding form*.

These drop-down boxes will allow you to limit the types and number of elements that will be displayed in the *paged* list of *items* that appears on the bottom of the *right margin area* when you select a data source.



Figure 15: Paged list of items to be geocoded

By *paged* we mean that not all *items* are shown in one tall list, rather each page contains ten *items* and *Previous* and *Next* buttons allow you to move between the pages.

Once you click on an *item* in the *item list*, the orange *geocoding form* appears at the lower part of the main content area.

Step 3: Using the geocode form and the view/edit attributes form

Once you have clicked on an *item* from your selected data source you are now ready to update the location and attributes of the *item*.

Figure 16: Geocoding form controls

The *geocoding form* contains four editable elements:

1. The editable field *Name of item*
2. The editable field *X-coordinate (or Longitude)*
3. The editable field *Y-coordinate (or Latitude)*
4. The field *Confidence (0-100%)*

In addition, the *geocoding form* contains five buttons

- A *View all attributes* button that opens the *View/edit attributes form* as a popup.
- A *View link* button that appears if the data source contains a URL field opens an external link as a separate browser window.



Figure 17: Example of pop-up page displaying information about item based on URL column specified at time of upload

- A *View image* button that appears if the data source contains an image URL field displays the image in a popup window.

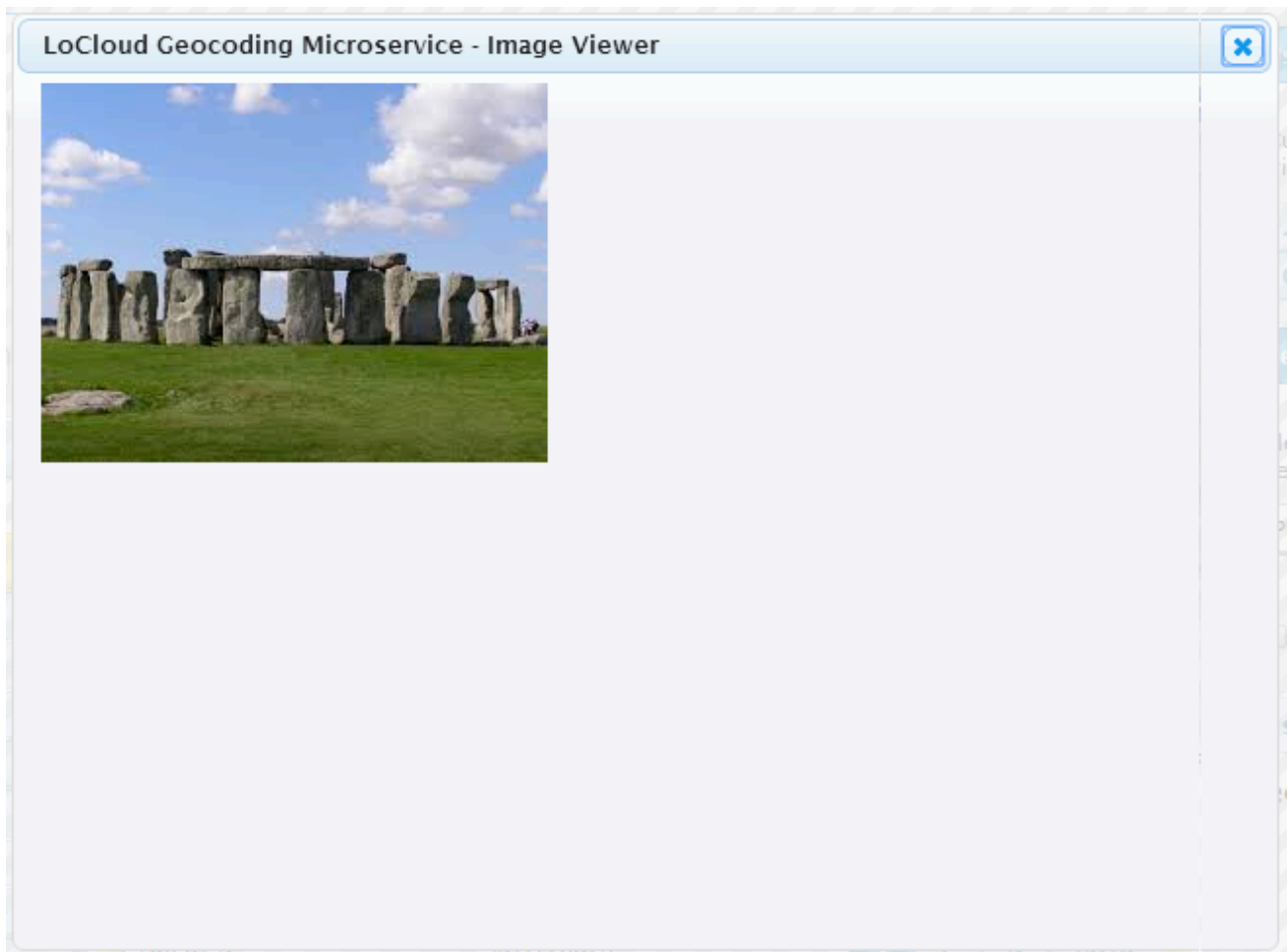


Figure 18: Example of image-popup based on `image_url` column specified at the time of upload

- A *Cancel* button that closes the *geocoding form* discarding any changes
- A *Save* button that saves any changes made in the form.
 - Please note that when you press the *Save* button, an indicator icon appears in front of the *item* name in the *item list*.
 - The icon uses the traffic light paradigm where the color corresponds to the *confidence value* set in the geocoding form where:
 1. = green light ($\geq 90\%$ confidence)
 2. = yellow light ($\geq 75\%$ confidence)
 3. = red light ($\geq 20\%$ confidence)

Records marked as *error* ($< 20\%$ confidence) are displayed with a prohibition sign icon in front of the item name.



Figure 19: Different confidence level indicators for items to be geocoded

Add coordinates to an item

There are two ways you can add coordinates to the currently selected item:

1. You can select one of the drawing tools from the tool bar in the upper right corner of the map and then click in the map. The coordinates will then be transferred to the respective fields in the form. All records will get a point representation but you can also draw lines and polygons.
2. You can manually edit the content of the fields if you for an example have collected GPS-coordinates - this is accurate but cumbersome

The first option is the one we are going to emphasize in this user manual.

View and edit source item attributes

The *geocoding form* only allows you to view and edit three attributes. If you want to see all the attributes that exist for an *item* you must use the *View all attributes* button. This will produce the *View/edit source attributes form*

This form in a table grid with three columns:

1. The first column contains the names of the attribute (or field)
2. The second column contains the original value of the attribute
3. The third column is empty but enable users to enter an alternative values into the form

Once a user has made changes to an items attributes he or she can choose to save them by clicking the *Save attribute edits button* or discard them by clicking the *Close without saving button*.

Either way, the data are not updated in the database until the user clicks *Save* in the Geocoding form.

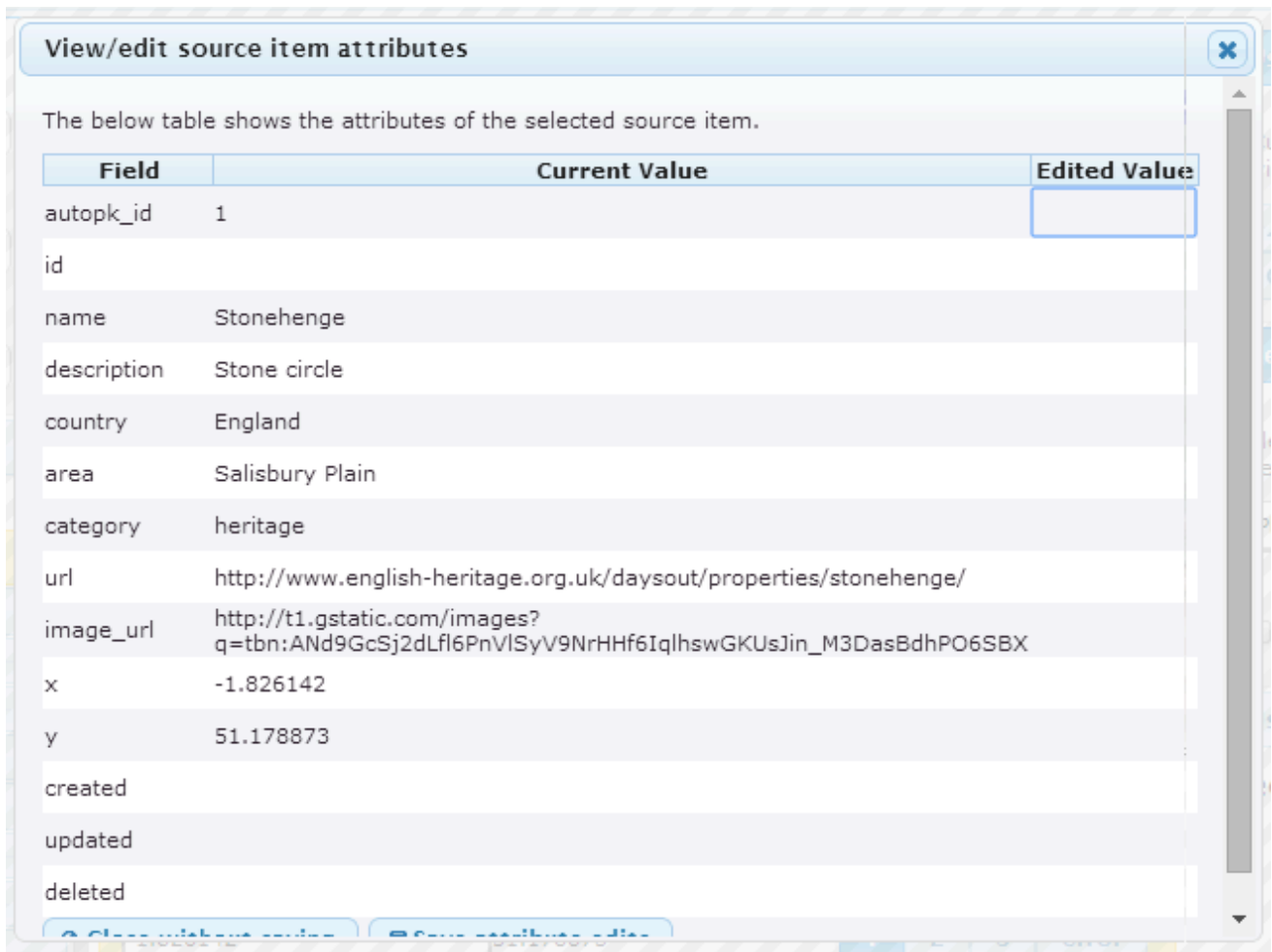


Figure 20: User interface for editing of attributes of source item

Step 4: Using the map interface

The map is really very simple. It is a tile-based map client similar to Google Maps built on the excellent, albeit similarly bulky, OpenLayers library.

The advantage with this library is that it supports literally any GIS requirement known to mankind. The drawback is that it is a bit large as a download: i.e. ≥ 700 Kb. This is not so nice in an end-user application (although it is the same as Google Maps) but it is quite ok for a professional application like the Geocoding Application.

The benefit of using this library is that it permits you to use other basemap data sources than Google. You can mix and match between Open Street Map, Cloud Made, Bing, Yahoo, Here, Google and more. You can even connect your own WMS servers.

The map is very simple and only support some very simple functions.

- Click and hold the mouse button while you drag the mouse to move around in the map - this is called *panning* in the GIS world. Learn it now and you won't have to later.
- Use the **+** button on the upper left to zoom in one step
- You can also zoom in one step by double-clicking in the map
- Use the **-** button second from the top on the left to zoom out one step

- If you wish to zoom to a specific area, you can hold the [shift](#) button and the left mouse button simultaneously and drag a rectangle around the area you'd like to zoom in on
- Single click to place the selected *item* and update the X- and Y-coordinate fields

You can switch between different background maps by selecting the [+](#) button on the upper right side of the map. Only one *base layer* can be visible at the same time.

Step 5: Using search databases

At this stage, the datasource has been uploaded, the map is visible - but you are still not able to find the location of the item you are geocoding.

The Geocoding microservice contains a *search panel* at the *right margin area* of the user interface. Here you can presently choose to search in the Geonames database to see if the place you are looking for exists in the source.

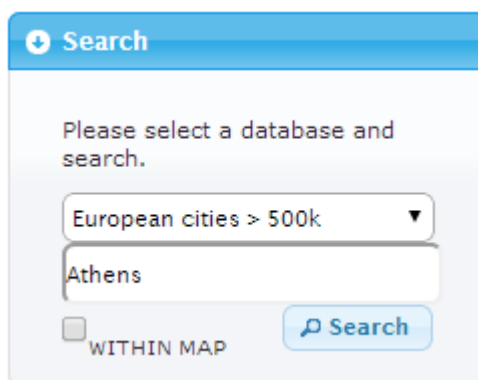


Figure 21: Search panel

1. Select a database using the Please select a database drop-down box in the Search panel.
2. At present, it only makes sense to choose Geonames as this is the only data source that covers all areas
3. Choose whether you want to limit the search to search results within the visible portion of the map
4. This can be useful if you are working with a common name that occurs many places but you know roughly where an item is located.
5. Enter a name or a partial name into the search field between the drop-down box and the within map check box and press search.
6. A search result list appears at the bottom of the screen
7. Select elements from the search result list to zoom and recenter the map on the respective search result
8. Now, use the map navigation functions to move around and single-click in the map to mark the correct location of your item

Invoking external web applications

Some times you will not be able to find the location of an object you would like to geocode simply by browsing the map. Don't worry. The world's biggest database of information is right in front of you: the Internet.

Be careful though, it is very easy to put information on the Internet and for this reason it is also very easy to put something wrong on Internet.

However, in order to help you with your geocoding work, we have integrated a number of mainstream map applications and made it possible to open them showing the same area as the one you are currently working in. These include:

- Google Maps (best in terms of completeness)
- Nokia Here (best in terms of accuracy)
- Wikimapia (best in terms of things you can't find anywhere else)
- Geonames (a global source of names)
- Google Search (when nothing else works)



Figure 22: Useful location sources panel

Whenever you click on the button with the name of one of these sources, a new browser window will open. The center of the map will be the same as the center of the map you have in the Geocoding application. The zoom level will usually be different.

By querying these external applications, you will be able to find the locations of many things that are otherwise impossible to locate.

Step 6: Downloading augmented data

At this stage, you have reached the end of the geocoding project and you would like to download the augmented data and put them back into the system where you manage your collections. This is easy and can be done like this:

1. Select the data source you would like to export in the drop-down box in the *data source panel* in the *left-margin* area of the user interface.
2. Click one of the *download buttons* in the top menu and save the file to your computer; you can choose between
 - CSV (the most basic format for working with any data source)
 - JSON (popular for many contemporary web applications)
 - KML (for viewing the data in Google Earth - or for loading into many GIS applications)
 - RDF (for loading into a graph database)

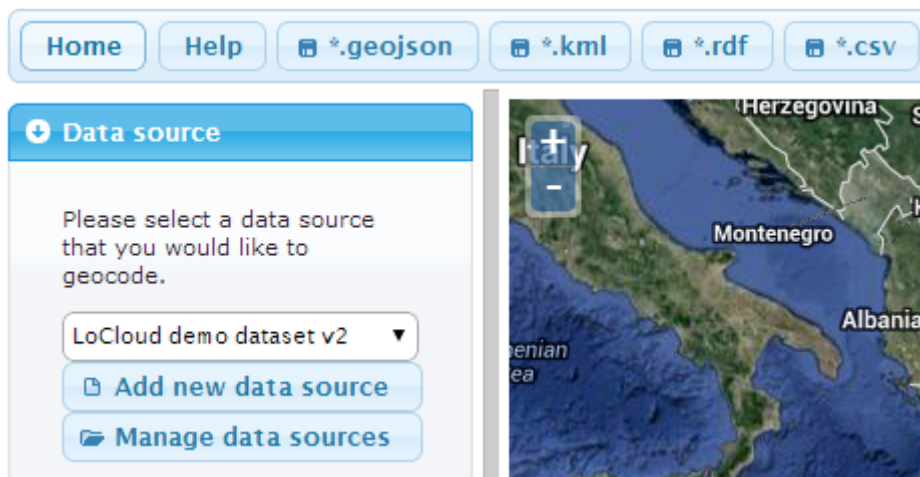


Figure 23: By selecting data source in the data source panel and clicking one of the “save” buttons data can be downloaded

6. How to install the geolocation enrichment tools

The geocoding enrichment tools consist of two independent components that may be configured to work together. The geocoding application and geolocation API may be installed “side-by-side” on the same server – or they may be installed on different instances/servers.

Geolocation API

The geolocation API is intended to be a single-instance centrally hosted API that can be invoked from any number of client applications across the Internet. It is therefore not envisaged that this will be installed with multiple instances.

Geocoding application

While the geolocation API is intended to be a single running instance as described above, the geocoding application has a different approach where easy deployability to any target environment is an objective in itself.

The application has been designed to be capable of running on low-end hardware and the typical configurations offered by cheap, mainstream shared web host providers like GoDaddy, BlueHost, WebHostingHub or similar. From this type of providers, for the price of ~10 Euro per month, a user gets access to a hosting environment that is capable of running the geocoding application with satisfactory performance.

This is an important consideration for the long-term sustainability of the LoCloud geocoding application. The sections below outline the system requirements and the installation instructions for the application.

System requirements

Operating system and software requirements

- Linux or Windows operating system
- Apache, Internet Information Server (IIS) or any other web server capable of executing PHP scripts
- PHP >= 5.3
- MySQL >= 5.5

Web browser requirements

The application has been tested on the following platforms but should in principle be capable of running in any mainstream Web Browser that has been updated as per 2014.

- Google Chrome v30.0.1599.69
- Firefox 24.0
- Internet Explorer >= 9

External Javascript libraries used

All of these libraries come packed with the installation and the end-user do not have to worry about the dependencies. They are included here for reference purposes only, in the event that 3rd party developers should wish to extend the code.

- OpenLayers 2.12 (for displaying map tiles on the client)

- Google API v3 (for displaying Google Maps web services)
- Proj4js (for client-side coordinate transformations)
- jQuery 1.10.2 (for enhanced Javascript functionality)
- jQuery-ui 1.10.2 (for enhanced user interfaces)

Step-by-step installation guide

The installation assumes you have a running Web Server that meets the requirements outlined above, i.e an operational LAMP, WAMP or WIMP platform. The installation then consists of three simple steps:

Step 1: Get the code

- Download the application as a zipped archive, i.e. locloud-geocoding.1.0.1.zip
- Unpack the application in a directory on your web server
 - e.g. "htdocs" on an Apache Server
 - e.g. "wwwroot" on an Internet Information Server

Step 2a: Manual configuration

- Create a MySQL database with the name of your choice
- Download the data definition SQL-file, i.e. locloud-geocoding.1.0.1.sql
- Run the SQL file in the newly created MySQL database
- Remember/note down the name of the host, database, username and password as you will need it for configuration (next step)
- Any variables that must be edited are located in the file *config.php*
- The config file is located in the folder "lib" folder in the root directory where you extracted the compressed archive
- Any configurable options are well explained with in-line comments
- At minimum, you must configure a valid MySQL database

Step 2b: Automatic configuration

- Point your web browser to the URL: <http://localhost/geocoding>.
- You will then be prompted to fill in information about the username and password for the MySQL instance as well as the title of your application.
- Once completed, you will be redirected to the login screen and can start using the application.

Step 3: Automatic updates subject to manual approval

- Once installed, the geocoding microservice client is capable of self-updating.
- When logged in as a super-user, you will see a button informing you of new versions and prompting you whether you would like to upgrade the local installation.

7. How the the tools are installed in LoCloud

Geolocation API

LoGeo API is developed to be available for end user as an API or as a console and is therefore fully operable.

LoGeo API: http://locloudgeo.eculturelab.eu/LoGeo_1_1/loGeo.aspx

LoGeo API console: http://locloudgeo.eculturelab.eu/Console_LoGeo_1_1_m/

Geocoding application

The geocoding application is a SaaS service that may be installed to any compatible web-hosting environment, whether virtualized or physical. It already has several running instances across the LoCloud partnership, but the latest development version may always be found at the URL below:

Geocoding application: <http://locloud.avinet.no/demo>

Integration with other LoCloud services

The end-user application is capable of “plugging into” the data-stream from local cultural heritage institutions to Europeana in order to geocode content in the LoCloud MoRe repository before it is ingested into Europeana. Similarly important, the end-user application is capable of ingesting record-based CSV files from any source.

This flexibility permits the application to be easily integrated with local content enrichment processes so that data are enriched with spatial coordinates prior to being ingested into MORE - and onwards into Europeana.

Conducting enrichment as close to the data source as possible is important in order to achieve persistent quality improvements. If enrichment only occurs up-stream by means of automatic methods, the origin and quality of the geocoded locations may not be verified, nor trusted, by end-users.

8. References

1. Bittner, T. et al. (2009). A spatio-temporal ontology for geographic information integration, *International Journal of Geographical Information Science*, vol. 23, no. 6
2. Clough P. (2010). *Extracting Metadata for Spatially-Aware Information Retrieval on the Internet*, University of Sheffield
3. Goldberg, D. W., Wilson, J. P., Knoblock, C. A. (2007). *From Text to Geographic Coordinates*
4. Guo, Q., Liu, Y. and Wiczorek, J.(2008). Georeferencing locality descriptions and computing associated uncertainty using a probabilistic approach, *International Journal of Geographical Information Science*, vol. 22, no. 10
5. Hastings, J. T. (2008). Automated conflation of digital gazetteer data, *International Journal of Geographical Information Science*, vol. 22, no. 10
6. <http://developer.yahoo.com/geo/placemaker/>
7. http://digmap3.ist.utl.pt:8080/geoparser_econnect/
8. <http://www.digmap.eu/doku.php>
9. <http://www.geonames.org/>
10. http://www.getty.edu/research/conducting_research/vocabularies/tgn/
11. Janowicz, K. and Keßler, C. (2008). The role of ontology in improving gazetteer interaction, *International Journal of Geographical Information Science*, vol. 22, no. 10
12. Vodeb V., Zakrajsek, F. (2013). *Geocoded Digital Cultural Content*, Roma: Linked Heritage project
13. Vodeb V., Zakrajsek, F. (2014). Geographical Mapping of Art Nouveau Collections. In: *Uncommon Culture*, vol. 4, no. 7/8, presents also 3D samples of ArtNouveau heritage
14. Vodeb, Vlasta (2012). Georazčlenjevanje metapodatkovnega opisa kulturne dediščine/Geoparsing the Cultural Heritage Metadata. *Knjižnica*, letnik 56, številka 3, pp 191-203
15. Zakrajsek, F., Vodeb, V. (2014). eCultureMap – Link to Europeana Knowledge. In: *Theory and Practice of Digital Libraries - TPDL 2013 Selected Workshops, Communications in Computer and Information Science Volume 416*, 2014, pp 184-189
16. Santos, W. (2012). 56 Geocoding APIs: Geocoder, Google and MapLarge, July 25th, 2012 (<http://www.programmableweb.com/news/56-geocoding-apis-geocoder-google-and-maplarge/2012/07/25>, accessed 1.8.2012)
17. Pouliquen, B. et al. (2006). Geocoding Multilingual Texts: Recognition, Disambiguation and Language Resource and Evaluation Conference (LREC) proceedings, ELRA/ELDA
18. Kebeck, J. (2010). Batch Geocoding and Batch Reverse-Geocoding with Bing Maps, Bing Maps Blog
19. Al-Gfou, R., Skiena, S. (2012). SpeedRead: A Fast Named Entity Recognition Pipeline, *Proceedings of COLING 2012: Technical Papers*, p. 51–66, COLING 2012, Mumbai, December 2012

20. Cardoso, N., Silva, M.J. (Experiments with Semantic-flavored
21. Query Reformulation of Geo-Temporal Queries, Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access, NII, June 15-18, 2010, Tokyo, Japan, p.173-180
22. The Stanford Natural Language Processing Group , <http://nlp.stanford.edu/index.shtml>
23. Grover, C. Et.al. (2010). Use of the Edinburgh geoparser for georeferencing digitized historical collections, *Phil. Trans. R. Soc. A* 28 August 2010 vol. 368 no. 1925, p. 3875-3889
24. GATE ANNIE, Natural Language Processing, <http://gate.ac.uk>
25. MinorThird, Natural Language Processing, <http://minorthird.sourceforge.net/>
26. Freire, N. (2010). M5.5.5 The Europeana Geoparser – Second Prototype, version 0.1, *EuropeanaConnect*, 24.6.2010
27. Bloomberg, R. et al. (2010). D3.2. Functional specification for the Europeana Danube Release, Europeana v1.0, 31 August 2010, final version
28. Zakrajšek, F. (2010). D7.2: Guidelines for Geographic Location Description. Athena Project, 30 April 2010, Final

9. Glossary

Term	Description
API	Application Programming Interface
API console	API Console is obligatory service for deployment of the API, it demonstrates the use of an API
EDM	Europeana Data Model
ETRS89	European Terrestrial Reference System 1989
Gazeteer	Geographical dictionary or index which contains information on places and place names and is meant to be used in conjunction with a map or atlas
Geocoding	The process of translating a textual geo-reference such as a geographical name, a property reference or a street address into map coordinates
Geoparsing	Process of assigning geographic coordinates to textual words and phrases or other media
GIS	Geographical Information System
LAMP	Linux, Apache, MySQL, PHP
NER	Name Entity Recognition
NLP	Natural Language Processing
Reverse geocoding	The process of retrieving a textual geo-reference such as an address based on a set of coordinates.
WAMP	Windows, Apache, MySQL, PHP
WGS – 84	World Geodetic System 1984
WIMP	Windows, Internet Information Server, MySQL, PHP