Consiglio Nazionale delle Ricerche

Accessing Open Data on Tourism: the Tourpedia Software

A. Lo Duca, A. Marchetti

IIT TR-18/2020

Technical Report

Ottobre 2020

Istituto di Informatica e Telematica

Accessing Open Data on Tourism: the Tourpedia Software

Angelica Lo Duca, Andrea Marchetti
Institute of Informatics and Telematics,National Research Council
via G. Moruzzi 1, 56124 Pisa (Italy)
email: [name].[surname]@iit.cnr.it

Abstract

Tourpedia is a knowledge base containing about 70.000 records extracted from official open data released by governments and institutions. Data are referred to the tourism domain and include accommodations and attractions related to Italy, France and Spain. Through a specific procedure, data in Tourpedia are continuously updated. This document describes the Tourpedia software.

Keywords

Tourism, Open Data, Web API, Accomodation, Attraction

1. Introduction

Open data about tourism are often distributed through different websites and in different formats or data structures. Tourpedia aims to unify all these open data in order to provide a single website to access open data regarding tourism (Lo Duca 2019). Currently Tourpedia contains more than 70,000 accommodation facilities, collected by 21 of the official open data websites provided by Italian, French and Spanish Regions (12 sources are from Italy, 6 from France and 3 from Spain). All the available data is aggregated, updated continuously, stored in a local database, released under a public license and can be accessed through a Web API, called Tourpedia Web API (T-API). This document describes the Tourpedia software.

2. Tourpedia Architecture

Tourpedia is based on a modular architecture, which permits a developer to add new open datasets easily and keep them updated. Fig.1 illustrates the Tourpedia architecture. Starting from the bottom of the figure, the system is composed of the following modules:

- Sources - for each source of data, a new module is built. This module downloads the specific dataset from a given source. In the specific case of accommodations, sources are represented by Italian, French and Spanish Regions. However, new sources could be added dynamically to the system, without compromising the existing ones. For each source, the URL of the dataset must be specified, as well as its mapping file, as described later;
- Mapping - each dataset is mapped to a common mapping schema, as defined by the Tourpedia Data Model;
- Storage - each mapped dataset is saved into a local no-relational database;
- Web API - this module allows a user to access data directly.;
- Web application - at the top of the stack, a Web application can be built. An example of Web application could be a geographical map, with all the accommodations related to a Region.

As additional services, Tourpedia provides also a mechanism to update datasets periodically (i.e. once a day) and to log extracted information, such as the last modified date associated to each dataset and the number of downloaded data. Simply scheduling the crawler for each source every day, through a cron job, does datasets update.

The Tourpedia software is available as an open source Github repository[1]. The software is organized in three main modules: Download and Update (contained in the update folder), Web API (contained in the api folder) and App (contained in the app folder).

[1] https://github.com/alod83/tourpedia-it

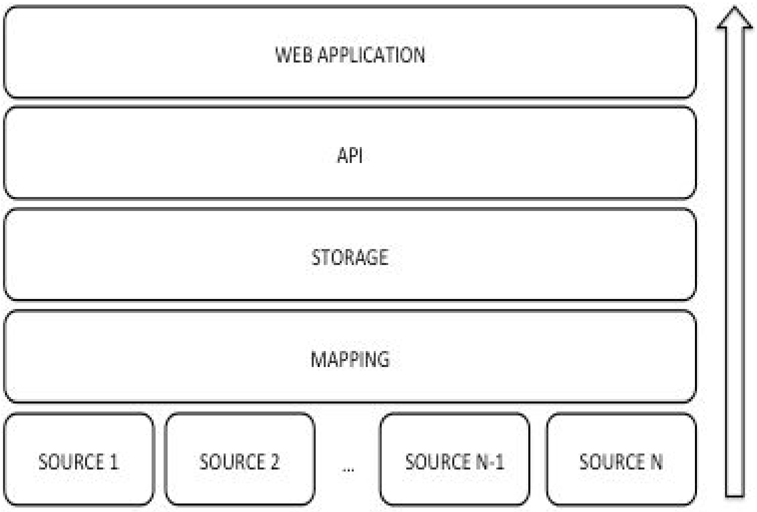

Fig. 1 The Tourpedia Architecture

3. Tourpedia Data Model (TDM)

Each downloaded dataset provides a specific data model. Tuscany has the simplest data model, with only 14 fields, while Lombardia defines the most complicated data model, with 35 fields. In order to standardize all the datasets, each of them is converted to a specific data model, which is defined by Tourpedia, i.e. the Tourpedia Data Model (TDM). This allows the system to standardize all data and to have a single data model for all the sources. Currently the TDM does not exploit anyone of the existing ontologies, because it simply takes all the possible fields from the open datasets and tries to unify them. The next step will be to represent data in one or more existing data models.

The TDM contains all the possible fields of all the open datasets and renames them to a single tag in English. The following Table shows the structure of the TDM for accommodations.

Field	Type	Description
_id	string	ID
name	string	Accommodation name
description	string	Description of the accommodation
category	string	Accommodation category (e.g. Bed and Breakfast, hotel, ..)
address	string	Address where the accommodation is located
postal-code	string	Postal code associated to the address
city	string	City where the accommodation is located

province	string	Acronym of the province
hamlet	string	Hamlet associated to the address
locality	string	Locality associated to the address
region	string	Region associated to the address
latitude	double	Latitude where the accommodation is located
longitude	double	Longitude where the accommodation is located
number of stars	integer	Number of stars
telephone	string	Telephone number
telephone2	string	Alternative telephone number
cellular phone	string	Cellular number
fax	string	Fax number
web site	string	Link to the Web site
email	string	Email address
beds	integer	Number of beds
rooms	integer	Number of rooms
suites	integer	Number of suites
toilets	integer	Number of toilets
facilities	list	List of facilities
sports equipment	list	List of sports equipment
languages	list	List of spoken languages
breakfast	int	Specify whether or not breakfast if available. Possible ranges are 0 and 1
congress halls	string	Available congress halls
opening period	string	Opening period
credit/debit card	list	List of support credit/debit cards
location	list	List of locations (e.g near the airport, near the train

		station,…)
manager	string	The name of the accommodation’s manager
elevation	integer	The accommodation elevation
high season price	list	A list containing the high season price for each category of room
low season price	list	A list containing the low season price for each category of room
photo	list	A list of photos URLs about the accommodation
chain	string	The chain which the accommodation belongs to

The following table shows the TDM used to represent tourist attractions.

Field	**Type**	**Description**
_id	string	unique identifier
name	string	name of attraction
description	string	description of attraction
category	string	category of attraction
address	string	address relative to attraction
region	string	region where is the attraction
province	string	province in which the attraction is located
city	string	name of the city in which the attraction is located
codistat	integer	code assigned by the Istat to the municipality where the attraction is located
postal-code	integer	code of the city in which is the attraction
latitude	double	latitude of the point in which is the attraction
longitude	double	longitude of the point where the attraction is located

telephone	string	telephone number
fax	string	fax number
email	string	email address
url	string	URL associated to the attraction
accessibility	string	conditions of accessibility to the public of the attraction
opening hours	string	opening times related to the attraction
entrance	string	type of entry (free or no) and price of input
image	string	URL associated with an image related to the attraction

4. Data Sources

The following table shows information regarding Italian data sources. Empty cells means that data are not available. It is interesting to note that for some sources the current status is not found.

Region	Format	Number of records	Last update	License	Status
Abruzzo					not found
Basilicata					not found
Calabria					not found
Campania					not found
Emilia Romagna					password required
Friuli Venezia Giulia	CSV	752	22/05/2019	IODL 2.0	confirmed
Lazio	XLS		2018	CC-BY 4.0	Present the data of Frosinone and Latina
Lazio Roma Capitale	XLS	20371	01/10/2020	CC-BY 4.0	new file every month
Liguria	CSV	circa 5000	23/09/2020		several files
Lombardia	CSV	27551	05/10/2020	CC-BY	confirmed
Marche	XLS	4925	13/10/2020		confirmed

Molise					n/a
Piemonte	CSV	9630	28/03/2018	CC BY 2.5	
Puglia	CSV	7425	2018	CCZero	new file
Sardegna					not found
Sicilia	ZIP-CSV	7285	03/08/2020	cc-by-sa	
Toscana	CSV	16655	23/06/2019	CC BY	confirmed
Trentino-Alto Adige (Trento)	XML	1502	25/10/2017	CC BY 4.0	confirmed
Trentino-Alto Adige (Bolzano)					not found
Umbria	CSV	4278	14/10/2020	CC BY 3.0 IT	confirmed
Valle D’Aosta					not found
Veneto	CSV	8656	07/02/2020	IODL 2.0	confirmed

5. Download and Update

This module implements the mechanism used to update the Tourpedia knowledge base. It is based on a module structure, where it is possible to add new modules easily. Each module corresponds to a different category of data (accommodations, attractions, events, restaurants and so on). At the moment only accommodations and attractions are contained in Tourpedia.

The code contains the following scripts, folders and files (only the most important ones are described):

5.1 config.ini

The config.ini file contains all the urls of the sources from which information must be extracted. In addition, it contains the parameters used to connect to the Mongo database.
The file has the following structure:

```
[Mongo]
url                   = "mongodb://localhost:27017"
collection            = "NUOVO"
db_category1          = "Name1"
db_category2          = "Name2"
```

The first section is called `Mongo` and contains information regarding the Mongo database, such as the url of the database (parameter `url`) and the name the collection used for each category (parameter `collection`). For each category the name of the output database must be specified. For example, if the category is accommodation, you must specify the keyword `db_accommodation`.

The second section of the config.ini file is called Sources and contains the names of all the sources providing data. For each category, the keyword category_list[] must be filled with the names of the sources. For example, for the category accommodation, you should specify the following parameters:

```
[Sources]
accommodation_list[] = "Basilicata"
accommodation_list[] = "EmiliaRomagna"
accommodation_list[] = "Friuli"
accommodation_list[] = "Liguria"
accommodation_list[] = "Lombardia"
accommodation_list[] = "Marche"
accommodation_list[] = "Piemonte"
accommodation_list[] = "Puglia"
accommodation_list[] = "Toscana"
accommodation_list[] = "Trentino"
accommodation_list[] = "Umbria"
accommodation_list[] = "Veneto"
accommodation_list[] = "Roma"
```

The values assigned to the variable `category_list[]` must contain the name of the sections contained in the remainder of the config.ini file. For example, if for “Basilicata” you have two datasets, you could use the names “Basilicata1” and “Basilicata2”.

From the third section, the config.ini file contains the sections specified in the `category_list[]`. Thus, in the example for accommodations, there are the sections `[Basilicata]`, `[EmiliaRomagna]` and so on. For each section referring to a source, the following parameters must be specified:

```
[Source]
url_category                     = ""
dataset_category[format]         = format
dataset_category[separator]      = ","
dataset_category[coord]          = 'True'

category[field1]                      = 1
category[field2]                      = 2
accommodation[field3]            = 3
```

The parameter `url_category` (where category is for example accommodation) specifies the direct link where download the link. The parameter `dataset_category[format]` (where category is for example accommodation) specifies the format of the target document. Supported formats are CSV, XSL, JSON and ZIP. In the case of a CSV file you must also specify the parameter `dataset_category[separator]`, which can be for example a comma “,”. You can also specify if geographical coordinates are available, by setting to True or False the parameter `dataset_category[coord]`.

In the final part of the section, you must specify the mapping between the original source file and the fields used in the database. This is done by using the Tourpedia Mapping Language (TML). For example, you can say that the `field1` corresponds to the first column of the original file.
The following piece of code shows a practical example:

```
[Basilicata]
url_accommodation                    =
"http://www.aptbasilicata.it/fileadmin/uploads/Documenti_vari/Servizi_t
urisici_Professionisti_ecc/Strutture_ricettive_sito_APT.xls"
dataset_accommodation[format]        = XLS
dataset_accommodation[separator]     = ","
dataset_accommodation[coord]         = 'True'

accommodation[name]                  = 1
accommodation[description]           = 2
accommodation[number of stars]       = 3
accommodation[address]               = 4,5
accommodation[telephone]             = 6
accommodation[cellular phone]        = 7
accommodation[fax]                   = 8
accommodation[web site]              = 9
accommodation[email]                 = 10
accommodation[opening period]        = 11
accommodation[beds]                  = 12
accommodation[region]                = "Basilicata"
accommodation[country]               = "Italy"
```

In the example, the field `region` is set to a constant value for all the records of the dataset (`Basilicata`). Details regarding the TML are given in the next section.
If you want to manage a specific source, you can add it to the folder `sources/category` and define your custom management code.

5.2 update_category.php

For each category, a different script must be created (e.g. `update_accommodations.php`, `update_attractions.php`). If you want to create a new category, you can simply copy one existing script and adapt it to your purposes. For example, you can create a copy of the script `update_accommodations.php` and change the word *accommodation* to your category every time it appears in the script.

5.3 utilities

This folder contains all the scripts used to parse source files. It contains the following scripts:

1. `csv.php` - it executes the parsing of a CSV file
2. `convert_to_csv.php` - it converts a file available as http url to CSV
3. `zip.php` - it unzips a file
4. `mongo.php` - it contains some functions to interact with the mongo database
5. `functions.php` - it implements the Tourpedia Mapping Language

5.4 sources

This folder contains the script used to parse specific source files, which are not included in the default configuration (i.e. CSV, XSL, JSON). If you want to create a new specific parser, you should add a new folder within sources folder, having the name of the analysed category. Within the category folder, you should create your parser script, which should be defined as a function.

5.5 Dataset Enrichment

This folder contains all the functions which can be used to enrich the dataset. At the moment, the only supported enrichment is geocoding (geoconding.php), which is implemented as a function. If you want to add a new enrichment service (e.g. Named Entities), you can create the script `named_entities.php`, which defines the function `named_entities()`.

6. Tourpedia Web API

6.1 Overview

The Tourpedia Web API (T-API) is the principal way to get data out of the Tourpedia Knowledge Base. T-API is an HTTP-based API that applications and developers can use to query data. Since the T-API is HTTP-based, it works with any language that supports HTTP, such as cURL and urllib. For example, you can use the T-API directly in your browser, by pasting the following URL in the search bar:

http://tour-pedia.org/it/api/query.php?category=accommodation&city=Roma

Similarly, you can perform the following cURL request for the same HTTP request:

```
curl -i -X GET \
 "http://tour-pedia.org/it/api/query.php?category=accommodation&city=Roma”
```

All HTTP requests are passed to the http://tour-pedia.org/it/api/query.php? host URL. Data transfers conform to HTTP/1.1. The T-API is completely free and does not require any access token. In addition it is not subjected to a limitation rate.

The HTTP response of the T-API is in JSON. The following piece of code shows an example of JSON response:

```
{
	LOM1546:
	{
		_id: "LOM1546",
		name: "1 STILE",
		address: "VIA CALVI 51",
		latitude: 45.15707,
		longitude: 10.794338
	},
	LOM9053:
	{
		_id: "LOM9053",
		name: "1000 TORRI ALEX E ANGIE",
            address: "VIA G. E G. PAGLIA 2/E",
            latitude: 45.692795,
            longitude: 9.66768
      },
}
```

6.2 T-API Usage

Queries always begin with a category, which is mandatory. A category is a tourism category, such as accommodation, attractions, restaurants and events. At the moment two categories of data can be accessed through the T-API:

- accommodation - this category includes hotels, bed & breakfast, campings and so on.
- attraction - this category contains points of interests, such as churches, libraries, monuments and so on.

An example of query for the category *attraction* is the following:

```
curl -i -X GET \
 "http://tour-pedia.org/it/api/query.php?category=attraction"
```

Two types of requests can be done: 1) collections of objects, 2) single objects, which is performed by specifying the specific id of the object. In the first case, some basic information is returned regarding each object. In the second case, all the available details regarding an object are returned.

Collections of objects

Requests for both categories support the following parameters:

Parameter	Mandatory (yes/no)	Description
`region`	no	The region where the category is located, e.g. Toscana, Lazio The string must be specified in capital letters and in Italian
`province`	no	The province where the category is located. The province must be specified through two characters, e.e. Bg indicates Bergamo The string must be specified in capital letters
`city`	no	The city where the category is located, e.g. Roma The string must be specified in capital letters and in Italian
`place`	no	The city or the region where the category is located. The string must be specified in capital letters and in Italian
`min_latitude`	no	the minimum latitude at which the category is

		located. This field can be used with max_latitude, min_longitude and max_longitude to define a rectangle within the search must be done
max_latitude	no	the maximum latitude at which the category is located. See min_latitude for other details.
min_longitude	no	the minimum longitude at which the category is located. See min_latitude for other details.
max_longitude	no	the maximum longitude at which the category is located. See min_latitude for other details.
not_null_latitude	no	if set, this parameter returns only results with not null latitudes
not_null_longitude	no	if set, this parameter returns only results with not null longitudes

Requests for accommodation support the following additional parameters:

Parameter	**Mandatory (yes/no)**	**Description**
min_beds	no	a number indicating the minimum number of beds provided by the accommodation, e.g. 5
max_beds	no	a number indicating the minimum number of beds provided by the accommodation, e.g. 5

Parameters can be combined, thus, for example you can search for accommodations located in a city and having a minimum number of beds. Every query returns the following JSON response:

```
{
      LOM8975 : {
            _id: "LOM8975",
            name: "15 QUINDICI BY SERENDIPITY ROOMS",
            address: "CORSO ITALIA 15A",
            latitude: 45.458141,
            longitude: 9.188362
       }
      LOM8976 : {...}
      ...
}
```

Each object of the collection is identified through its identifier (*LOM8975* and *LOM8976* in the example above). Then, for each object, the following fields are returned:

Field	Description
`_id`	the unique identifier representing the object within the Tourpedia Knowledge Base. This id can be used as the value of a single object query. This field cannot be null
`name`	The name of the category, e.g. “Hotel Bologna”. This field could be null, if the information is not available.
`address`	The address of the category, e.g. “via Bologna 5”. Usually, this field does not contain the name of the city. This field could be null, if the information is not available.
`latitude`	The latitude of the category. This field could be null, if the information is not available.
`longitude`	The longitude of the category. This field could be null, if the information is not available.

Single Objects
Requests for both categories support the following parameter:

Parameter	Mandatory (yes/no)	Description
`_id`	no	the specific id associated with a single category object. This parameter is used to specify a request for a single object

Every query asking for an accommodation returns a JSON response, like the following one:

```
{
LOM8975: {
   _id: "LOM8975",
   name: "15 QUINDICI BY SERENDIPITY ROOMS",
   category: "Complementari",
   description: "Foresterie lombarde",
   address: "CORSO ITALIA 15A",
   city: "Milano",
   province: "Mi",
   postal-code: 20122,
   email: "15quindici.sr@gmail.com",
   web site: "",
   telephone: "02 36707046",
   fax: "0376 88335",
   rooms: 5,
   suites: 0,
   beds: 10,
   latitude: 45.458141,
   longitude: 9.188362,
   region: "Lombardia",
   country: "Italy",
   toilets: 0,
   credit/debit cards: [
      "American express",
      "Diners",
      "Master Card",
      "Visa"
  ],
  languages: [
      "English",
      "Polacco",
      "German"
  ],
  breakfast: "1",
  facilities: [
      "Access. diversamente abili parziale",
      "Accesso a mezzi pubblici",
      "Aria condizionata centralizzata",
      "Frigorifero",
      "Somministrazione alcolici",
      "Somministrazione bevande"
  ],
   congress halls: "Congressi capacita' Max 100,Congressi capacita'
Min  80,Numero sale congressi",
   sports equipment: [
       "Piscina scoperta"
   ],
   location: [
       "ZONA_STAZIONE_FS"
   ],
   locality: "G forse diventa P"
 }
}
```

The following table shows the returned fields for accommodations. Note that the only mandatory field is the _id, while the others are returned only if they are available in the Tourpedia Knowledge Base.

Field	Description
_id	ID
name	Accommodation name
description	Description of the accommodation
category	Accommodation category (e.g. Bed and Breakfast, hotel, ..)
address	Address where the accommodation is located
postal-code	Postal code associated to the address
city	City where the accommodation is located
province	Acronym of the province
hamlet	Hamlet associated to the address

locality	Locality associated to the address
region	Region associated to the address
latitude	Latitude where the accommodation is located
longitude	Longitude where the accommodation is located
number of stars	Number of stars
telephone	Telephone number
telephone2	Alternative telephone number
cellular phone	Cellular number
fax	Fax number
web site	Link to the Web site
email	Email address
beds	Number of beds
rooms	Number of rooms
suites	Number of suites
toilets	Number of toilets
facilities	List of facilities
sports equipment	List of sports equipment
languages	List of spoken languages
breakfast	Specify whether or not breakfast if available. Possible ranges are 0 and 1
congress halls	Available congress halls
opening period	Opening period
credit/debit card	List of support credit/debit cards
location	List of locations (e.g near the airport, near the train station,…)
manager	The name of the accommodation's manager
elevation	The accommodation elevation
high season price	A list containing the high season price for each category of room
low season price	A list containing the low season price for each category of

	room
photo	A list of photos URLs about the accommodation
chain	The chain which the accommodation belongs to

Every query asking for an attraction returns the following JSON object:

```
{
      LAZ6_854:
      {
            _id: "LAZ6_854",
            name: " Biblioteca del Dipartimento di Storia, Culture,
            Religioni dell'Università degli studi di Roma La Sapienza -
            Punto di servizio di Studi storico religiosi",
            latitude: 41.903359,
            longitude: 12.515426,
            description: "Biblioteche",
            region: "Lazio",
            province: "Rm",
            city: "Roma",
            address: "Piazzale A. Moro 5",
            category: "Istituto di insegnamento superiore",
            telephone: "+39 0649913903",
            postal-code: 185,
            email: "walter.mazzotta@uniroma1.it",
            country: "Italy",
            fax: "+39 064453753",
            url:
            "http://w3.uniroma1.it/archeologia/Biblioteca/Sezioni/filgre
            .htm"
      }
}
```

Returned fields for attractions are described in the following table. Similar to accommodation, also for attractions, the only mandatory field is the _id.

Field	**Description**
_id	unique identifier
name	name of attraction
description	description of attraction
category	category of attraction
address	address relative to attraction
region	region where is the attraction

province	province in which the attraction is located
city	name of the city in which the attraction is located
codistat	code assigned by the Istat to the municipality where the attraction is located
postal-code	code of the city in which is the attraction
latitude	latitude of the point in which is the attraction
longitude	longitude of the point where the attraction is located
telephone	telephone number
fax	fax number
email	email address
url	URL associated to the attraction
accessibility	conditions of accessibility to the public of the attraction
opening hours	opening times related to the attraction
entrance	type of entry (free or no) and price of input
image	URL associated with an image related to the attraction

6.3 Some Examples

This section provides a list of common scenarios regarding the usage of the T-API. For all the following example, but the last one, the response has the following structure:

Response

```
{
      LOM8975:
      {
            _id: "LOM8975",
            name: "15 QUINDICI BY SERENDIPITY ROOMS",
            address: "CORSO ITALIA 15A",
            latitude: 45.458141,
            longitude: 9.188362
      },
      LOM7895:
      {
            _id: "LOM7895",
            name: "18TH CENTURY HOME",
            address: "PIAZZA SANT'ALESSANDRO 3",
            latitude: 45.460721,
            longitude: 9.186486
      },
      LOM7854:
      {
           _id: "LOM7854",
           name: "1988 DI HE ZHIYI",
           address: "VIA LUDOVICO D'ARAGONA 10",
           latitude: 45.490185,
           longitude: 9.234756
       }
       ....
}
```

Get the list of all the accommodations located in a city (Milano)

Request

```
curl -i -X GET \
"http://tour-pedia.org/it/api/query.php?category=accommodation&city=Mil
ano”
```

Get the list of all the accommodations located in a city or region with a given name (Milano)

Request

```
curl -i -X GET \
"http://tour-pedia.org/it/api/query.php?category=accommodation&place=Mi
lano”
```

Get the list of all the accommodations having a number of beds between 3 and 60

Request

```
curl -i -X GET \
"http://tour-pedia.org/it/api/query.php?category=accommodation&min_beds
=3&max_beds=60”
```

Get the list of all the accommodations contained into a rectangle bounding box

Request

```
curl -i -X GET \
"http://tour-pedia.org/it/api/query.php?category=accommodation&min_lati
tude=41.90&max_latitude=41.91&min_longitude=12.46&max_longitude=12.47”
```

Get the list of all the attractions with latitude and longitude not null

Request

```
curl -i -X GET \
"http://tour-pedia.org/it/api/query.php?category=attraction&not_null_la
titude=1&not_null_longitude=1″
```

Get all the accommodations located in a city (Milano) and having not null latitude and longitude

Request

```
curl -i -X GET \
"http://tour-pedia.org/it/api/query.php?category=accommodation&city=Mil
ano&not_null_latitude=1&not_null_longitude=1″
```

Get a single accommodation by _id (LOM8975)

Request

```
curl -i -X GET \
"http://tour-pedia.org/it/api/query.php?category=accommodation&_id=LOM8
975″
```

Response

```
{
LOM8975: {
   _id: "LOM8975",
   name: "15 QUINDICI BY SERENDIPITY ROOMS",
   category: "Complementari",
   description: "Foresterie lombarde",
   address: "CORSO ITALIA 15A",
   city: "Milano",
   province: "Mi",
   postal-code: 20122,
   email: "15quindici.sr@gmail.com",
   web site: "",
   telephone: "02 36707046",
   fax: "0376 88335",
   rooms: 5,
   suites: 0,
   beds: 10,
   latitude: 45.458141,
   longitude: 9.188362,
   region: "Lombardia",
   country: "Italy",
   toilets: 0,
   credit/debit cards: [
      "American express",
      "Diners",
      "Master Card",
      "Visa"
  ],
  languages: [
      "English",
      "Polacco",
      "German"
  ],
  breakfast: "1",
  facilities: [
      "Access. diversamente abili parziale",
      "Accesso a mezzi pubblici",
      "Aria condizionata centralizzata",
      "Frigorifero",
      "Somministrazione alcolici",
      "Somministrazione bevande"
  ],
   congress halls: "Congressi capacita' Max 100,Congressi capacita'
Min  80,Numero sale congressi",
   sports equipment: [
       "Piscina scoperta"
   ],
   location: [
       "ZONA_STAZIONE_FS"
   ],
   locality: "G forse diventa P"
 }
}
```

7. The Tourpedia Mapping Language (TML)

Tourpedia defines a mapping language, namely Tourpedia Mapping Language (TML) to specify a mapping strategy from the source data model to the TDM. The TML allows the definition of some basic operations, such as the aggregation of two fields or the normalization of a number.

simple column

The simple assignment specifies which column number in the source file must be assigned to a field. Column numbers start from 0.

```
field = number
```

For example, if you have the following CSV table:

name	address	city	n_beds

you can specify that the address field must be associated to column 1:

```
address = 1
```

concatenation

You can specify that a field is the concatenation of two or more columns by using the comma operator without a separating space.

```
field = number1,number2,...,numberN
```

Considering the previous CSV table, you can say that the field address is composed of the columns 1 and 2 (address and city)

```
address = 1,2
```

normalization

This operation is available for numbers. It permits the division of a column by a constant value. The syntax is the following:

```
field = number/constant
```

Considering the previous CSV table, you can divide the number of beds by 100:

```
n_beds = 3/100
```

UTF8 encoding

This operation is available for strings. It permits to convert a column value to the UTF8 encoding. The syntax is the following:

```
field = number<utf8
```

Considering the previous CSV table, you can encode the address column in UTF8:

```
address = 1<utf8
```

other operations

You can define your custom operations by using the < operator. This can be done by modifying the function `get_record()` contained in the `utilities/functions.php`.

References

Lo Duca, A. and Marchetti, A. (2019). Open data for tourism: the case of Tourpedia. Journal of Hospitality and Tourism Technology.