Location Name Finder - Analyser Documentation

1

Generated by Doxygen 1.8.9.1

Sat Apr 18 2015 15:02:19

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 Comparison Namespace Reference

Namespaces

- OSComparator
- YPComparator

5.2 Comparison.OSComparator Namespace Reference

Classes

- class ComparatorOrdnanceSurvey

Functions

- def run_test_query ()

5.2.1 Detailed Description

```
This module is added to form comparisons between the convex hull's that have been
established and official mapping data. This module will only be used for experimentation purposes. Note,
that this module will use the 'shapely' module to produce shape objects and perform
comparisons.
Created on 11 Mar 2015

@author: Billy Hickman
```

5.2.2 Function Documentation

5.2.2.1 def Comparison.OSComparator.run_test_query ( )

```
Runs a test query on the shape analyser based on the place names found in a
'shp' file. Will use the ComparatorOrdnanceSurvey class to perform this analysis.
```

Definition at line 302 of file OSComparator.py.

5.3 Comparison.YPComparator Namespace Reference

Classes

- class ComparatorYourPlaceNames

Functions

- def run_test_query ()

5.3.1 Detailed Description

```
This module is used to perform comparisons between the convex hull's that this
system creates and the data sets gathered from yourplacenames.com

Created on 17 Mar 2015

@author: Billy Hickman
```

5.3.2 Function Documentation

5.3.2.1 def Comparison.YPComparator.run_test_query ( )

```
Runs a test query on the comparator, based on a pre-configured list
of search terms. Will use the ComparatorOrdnanceSurvey class to perform this analysis.
```

Definition at line 272 of file YPComparator.py.

5.4 Main Namespace Reference

Namespaces

- AnalyserModule

Classes

- class ApplicationEntry

Variables

- tuple CURRDIR = os.path.dirname(inspect.getfile(inspect.currentframe()))
- tuple PARENTDIR = os.path.dirname(CURRDIR)
- tuple urls
- tuple app = web.application(urls, globals())

  API instance from 'web.py'.

5.4.1 Variable Documentation

5.4.1.1 tuple Main.app = web.application(urls, globals())

API instance from 'web.py'.

Definition at line 18 of file __init__.py.

5.4.1.2 tuple Main.CURRDIR = os.path.dirname(inspect.getfile(inspect.currentframe()))

Definition at line 11 of file __init__.py.

5.4.1.3 tuple Main.PARENTDIR = os.path.dirname(CURRDIR)

Definition at line 12 of file __init__.py.

5.4.1.4 tuple Main.urls

Initial value:

```
1 = (
2     '/(.*)', 'ApplicationEntry'
3 )
```

Definition at line 15 of file __init__.py.

5.5 Main.AnalyserModule Namespace Reference

Classes

- class Analyser

5.6 Mining Namespace Reference

Namespaces

- DBScan
- GeoNameFinder

5.7 Mining.DBScan Namespace Reference

Classes

- class DBSCAN

5.8 Mining.GeoNameFinder Namespace Reference

Classes

- class GeoNameFinder

5.9 Utils Namespace Reference

Namespaces

- Constants

- DBAdapter
- Marker
- Methods

5.10 Utils.Constants Namespace Reference

Classes

- class Constants

5.10.1 Detailed Description

```
Created on 9 Feb 2015

A series of useful constants are defined within this module.
This includes usernames, passwords and DB table names.

@author: Billy
```

5.11 Utils.DBAdapter Namespace Reference

Classes

- class DBAdapter

5.11.1 Detailed Description

```
Created on 9 Feb 2015

A module designed to perform all database communication.
The class 'DBAdapter' provides facilities for maintaining a connection
and providing all required query methods.

@author: Billy Hickman
```

5.12 Utils.Marker Namespace Reference

Classes

- class Marker

5.12.1 Detailed Description

```
Created on 11 Feb 2015

@author: Billy Hickman
```

5.13 Utils.Methods Namespace Reference

Classes

- class Methods

Chapter 6

Class Documentation

6.1 Main.AnalyserModule.Analyser Class Reference

Public Member Functions

- def __init__ (self)
- def retrieve_all_tweets (self, query, flag)
- def retrieve_cluster_tweets
- def retrieve_yourplacenames_cluster
- def re_run_and_grow_db_scan (self, query, flag, previous_eps)
- def remove_duplicates (self, points)
- def set_up_for_vdscan (self, points)

Public Attributes

- recent_eps_value_used

  Stores the most recent EPS value used.
- recent_minpts_value_used

  Stores the most recent minPts value used.
- db_adapter

  Database Communicator/Adapter instance.
- found_fcl_tag

  The FCL Tag found on the GeoNames API.

6.1.1 Detailed Description

```
This the main 'Analyser'.

It will perform all of the ordering, clustering and meaningful output.
It uses other classes to perform the computations but acts as a master
class to handle all types of queries.
```

Definition at line 17 of file AnalyserModule.py.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 def Main.AnalyserModule.Analyser.__init__ (*self*)

```
CONSTRUCTOR
```

Definition at line 26 of file AnalyserModule.py.

6.1.3 Member Function Documentation

6.1.3.1 def Main.AnalyserModule.Analyser.re_run_and_grow_db_scan (*self, query, flag, previous_eps*)

```
The method that performs the growing effect.

This is a method which has been added to increase the DBScan EPS
value, should more than one cluster be found. It increases the EPS
value in increments of 0.1 until it reaches a maximum. It will iterate
upwards from the previously used EPS value.

Once it finds only one cluster it will terminate.
@param query: The query string.
@param flag: A flag indicating whether it is for edinburgh or cardiff.
@param previous_eps: The previous EPS value. (used to increment from this)
```

Definition at line 206 of file AnalyserModule.py.

6.1.3.2 def Main.AnalyserModule.Analyser.remove_duplicates (*self, points*)

```
Added to remove duplicate latitude and longitude values.
This has since been deprecated and is solved by only accepting
tweets from one username.
@deprecated
```

Definition at line 253 of file AnalyserModule.py.

6.1.3.3 def Main.AnalyserModule.Analyser.retrieve_all_tweets (*self, query, flag*)

```
A method to retrieve all tweets which match the search query. This will be all
tweets where the 'query' resides in the tweet text, with wildcards either side.
The data returned from this method will be in a JSON format.

@attention: Note, this method also includes code that filters out duplicate
usernames.

@param query: The query string being looked for within the Tweet text.
@param flag: The flag indicating what table to retrieve Tweets from.
@return: JSON formatted results
```

Definition at line 39 of file AnalyserModule.py.

6.1.3.4 def Main.AnalyserModule.Analyser.retrieve_cluster_tweets (*self, query, flag, dbscan_eps_override* = None, *dbscan_minpoints_override* = None)

```
A method to locate clusters within the data set. It will use the DBScan algorithm in order
to establish clusters. This DBScan algorithm is run from a separate class.
It will also perform checks to see whether override values have been provided and apply
these to the algorithm. If no override parameters exist it will also try and start the
EPS and minPts values based on the geonames API.
Finally, it will also check to see whether more than one cluster has
been found and then perform growing of the EPS value until only one cluster exists.

@attention: Note, this method also includes code that filters out duplicate
usernames.

@param query: The query string being looked for within the Tweet text.
@param flag: The flag indicating what table to retrieve Tweets from.
@param dbscan_eps_override: The EPS override value. Defaults to None if not provided.
@param dbscan_minpoints_override: The min points (DBScan) override values.
Defaults to none if not provided.
@return: JSON formatted results
```

Definition at line 73 of file AnalyserModule.py.

6.1.3.5 def Main.AnalyserModule.Analyser.retrieve_yourplacenames_cluster (*self, query, flag, dbscan_eps_override =* None*, dbscan_minpoints_override =* None)

```
Serves an external 'YourPlaceNames.com' comaprison request.

A method added towards the end of the development.
It will use the class that was originally designed just
for comparison, 'YPComparator.com' to provide an API
method.
@param query: The query string being looked for within the Tweet text.
@param flag: The flag indicating what table to retrieve Tweets from.
@param dbscan_eps_override: The EPS override value. Defaults to None if not provided.
@param dbscan_minpoints_override: The min points (DBScan) override values.
Defaults to none if not provided.
@return: JSON formatted results
```

Definition at line 136 of file AnalyserModule.py.

6.1.3.6 def Main.AnalyserModule.Analyser.set_up_for_vdscan (*self, points*)

```
This will be called, once it has been established there are enough
results. It will set up the parameters for DBSCAN automatically.

@points: Points in tweets.

@deprecated: This was added as a test to see if VDBScan could
be implemented.
```

Definition at line 292 of file AnalyserModule.py.

6.1.4 Member Data Documentation

6.1.4.1 Main.AnalyserModule.Analyser.db_adapter

Database Communicator/Adapter instance.

Definition at line 37 of file AnalyserModule.py.

6.1.4.2 Main.AnalyserModule.Analyser.found_fcl_tag

The FCL Tag found on the GeoNames API.

Definition at line 110 of file AnalyserModule.py.

6.1.4.3 Main.AnalyserModule.Analyser.recent_eps_value_used

Stores the most recent EPS value used.

The most recent EPS value used.

Definition at line 32 of file AnalyserModule.py.

6.1.4.4 Main.AnalyserModule.Analyser.recent_minpts_value_used

Stores the most recent minPts value used.

The most recent minPts value used.

Definition at line 34 of file AnalyserModule.py.

The documentation for this class was generated from the following file:

- Main/AnalyserModule.py

6.2 Main.ApplicationEntry Class Reference

Public Member Functions

- def GET (self, name)

Public Attributes

- query_string

  The query entered by the user.
- type_flag

  The type flag indicating whether Edinburgh/Cardiff/Southampton.
- analyser

  Master class instance which handles all queries.
- dbscan_eps_value

  Latest EPS value used, ready to be returned via the API.
- dbscan_minpoints_value

  Latest minPts value used, ready to be returned via the API.

6.2.1 Detailed Description

```
This is the main class. This class will be called through the API.

It operates as a web service (API). The GET method is the one that is called
through the API. This class relies on the web.py package in order to run.
It handles the dispatching of the query to the operate classes and methods.

@author: Billy Hickman
```

Definition at line 20 of file __init__.py.

6.2.2 Member Function Documentation

6.2.2.1 def Main.ApplicationEntry.GET (*self, name*)

```
This is the main method that will be run by the API.

The 'web.py' will call this module whenever it receives a request.
This method will determine what query is requested and forward
this request to the appropriate method.

@param name: The name passed in as a URL.

@attention: Queries (name) are expected in a set format.
URL/getAll/*query*/*type_flag*
*query* - Query Text
*type_flag* - 1 (Cardiff) 2(Edinburgh) 3(Southampton/Portsmouth)
```

Definition at line 31 of file __init__.py.

6.2.3 Member Data Documentation

6.2.3.1 Main.ApplicationEntry.analyser

Master class instance which handles all queries.

Definition at line 60 of file __init__.py.

6.2.3.2 Main.ApplicationEntry.dbscan_eps_value

Latest EPS value used, ready to be returned via the API.

Definition at line 73 of file __init__.py.

6.2.3.3 Main.ApplicationEntry.dbscan_minpoints_value

Latest minPts value used, ready to be returned via the API.

Definition at line 75 of file __init__.py.

6.2.3.4 Main.ApplicationEntry.query_string

The query entered by the user.

Definition at line 56 of file __init__.py.

6.2.3.5 Main.ApplicationEntry.type_flag

The type flag indicating whether Edinburgh/Cardiff/Southampton.

Definition at line 58 of file __init__.py.

The documentation for this class was generated from the following file:

- Main/__init__.py

6.3 Comparison.OSComparator.ComparatorOrdnanceSurvey Class Reference

Public Member Functions

- def get_percentage_overlap (self, system_area, admin_area)
- def get_overlap_status (self, system_area, admin_area)
- def get_inside_percentage (self, un_clustered_points, admin_area)
- def __init__ (self)
- def calculate_amount_points_inside (self, un_clustered_points, admin_area)
- def calculate_overlap_percentage (self, system_area, admin_area)
- def check_for_overlap (self, system_area, admin_area)
- def start_admin_comparison (self, shapes)
- def get_convex_hull_points (self, points)
- def test_print_a_polygon (self, poly)
- def get_system_area (self, query)
- def get_system_unclustered_points (self, query)

Public Attributes

- yes_or_no_for_overlap
- overall_percentage_overlap

6.3.1 Detailed Description

```
The class that was implemented to perform experiments.
It will be used to compare the system results against
official mapping boundaries. Note, when the comments refer
to the 'admin' area this refers to the official shp file region.

@author: Billy Hickman
```

Definition at line 26 of file OSComparator.py.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 def Comparison.OSComparator.ComparatorOrdnanceSurvey.__init__ (*self*)

```
A constructor for the class that performs all comparisons.

Reads in all of the shapes from the .shp file and calculates
all the comparison factors for each shape.

@attention: The names from the shape file are used as place name input.
```

Definition at line 67 of file OSComparator.py.

6.3.3 Member Function Documentation

6.3.3.1 def Comparison.OSComparator.ComparatorOrdnanceSurvey.calculate_amount_points_inside (*self, un_clustered_points, admin_area*)

```
It will loop through the unclustered points and check how many
reside in the admin area
@param un_clustered_points: A list of unclustered points.
@param admin_area: An admin established area.

@return Percentage of points inside formatted as a String.
```

Definition at line 83 of file OSComparator.py.

6.3.3.2 def Comparison.OSComparator.ComparatorOrdnanceSurvey.calculate_overlap_percentage (*self, system_area, admin_area*)

```
Retrieves the percentage that one object intersects another.
It will use the admin area as the larger area and check for
an intersection percentage.
@param system_area: System established convex hull.
@param admin_area: Ordnance Survey convex hull.

@return Percentage of overlap formatted as a String.
```

Definition at line 115 of file OSComparator.py.

6.3.3.3 def Comparison.OSComparator.ComparatorOrdnanceSurvey.check_for_overlap (*self, system_area, admin_area*)

```
A method which will check for an overlap between the system area
and OS data.

@return: True - There is an overlap. False - No overlap.
```

Definition at line 135 of file OSComparator.py.

6.3.3.4 def Comparison.OSComparator.ComparatorOrdnanceSurvey.get_convex_hull_points (*self, points*)

```
A method to retrieve the convex hull points or the vertices that make
up this convex hull.
This method uses the hullPy module to calculate these points.
@param point objects.
@return A convex hull object - 'hullPy'.
```

Definition at line 200 of file OSComparator.py.

6.3.3.5 def Comparison.OSComparator.ComparatorOrdnanceSurvey.get_inside_percentage (*self, un_clustered_points, admin_area*)

```
Get percentage of points that reside inside the admin area.
@param un_clustered_points: Un-clustered points.
@param admin_area: Ordnance survey convex hull.
@return percentage formatted string.
```

Definition at line 57 of file OSComparator.py.

6.3.3.6 def Comparison.OSComparator.ComparatorOrdnanceSurvey.get_overlap_status (*self, system_area, admin_area*)

```
Return whether there is an overlap as a boolean between
the two geo objects.
@param system_area: System established convex hull.
@param admin_area: Ordnance Survey convex hull.
@return boolean indicating whether there is an overlap
```

Definition at line 46 of file OSComparator.py.

6.3.3.7 def Comparison.OSComparator.ComparatorOrdnanceSurvey.get_percentage_overlap (*self, system_area, admin_area*)

```
Retrieve the percentage intersection as a string.
@param system_area: System established convex hull.
@param admin_area: Ordnance Survey convex hull.
@return A string representing the percentage overlap.
```

Definition at line 36 of file OSComparator.py.

6.3.3.8 def Comparison.OSComparator.ComparatorOrdnanceSurvey.get_system_area (*self, query*)

```
Retrieve system established convex hull.

It will retrieve these Tweets from the database and then
apply the DBScan algorithm.
@param query: The query term (Tweets contain this word).
@return An array of point objects.
```

Definition at line 246 of file OSComparator.py.

6.3.3.9 def Comparison.OSComparator.ComparatorOrdnanceSurvey.get_system_unclustered_points (*self, query*)

```
Retrieves all system points prior to any clustering being undertaken.
The DBScan algorithm will not have been used on these points.

@param query: The query term (tweets contain this word).
@return: An array of point objects.
```

Definition at line 278 of file OSComparator.py.

6.3.3.10 def Comparison.OSComparator.ComparatorOrdnanceSurvey.start_admin_comparison (*self, shapes*)

```
Retrieves all of the boundaries that are included inside
the shape file. Loops through each shape and performs
a comparison for each one.

@param shapes: Shapes found in the shape file.
@return: None
```

Definition at line 144 of file OSComparator.py.

6.3.3.11 def Comparison.OSComparator.ComparatorOrdnanceSurvey.test_print_a_polygon (*self, poly*)

```
Print all of the coordinates in a polygon.

@return None
```

Definition at line 235 of file OSComparator.py.

6.3.4 Member Data Documentation

6.3.4.1 Comparison.OSComparator.ComparatorOrdnanceSurvey.overall_percentage_overlap

Definition at line 132 of file OSComparator.py.

6.3.4.2 Comparison.OSComparator.ComparatorOrdnanceSurvey.yes_or_no_for_overlap

Definition at line 54 of file OSComparator.py.

The documentation for this class was generated from the following file:

- Comparison/OSComparator.py

6.4 Comparison.YPComparator.ComparatorYourPlaceNames Class Reference

Public Member Functions

- def get_overlap_status (self)
- def get_inside_percentage (self)
- def get_intersection_percentage_as_string (self)
- def __init__ (self, system_generated_points, unclustered_points, query, flag)
- def get_points_as_list (self, list_of_point_objects)
- def start_the_checking_process (self)
- def calculate_amount_points_inside (self)
- def calculate_overlap_percentage (self)

- def check_for_overlap (self)
- def test_print_a_polygon (self, poly)
- def get_yourplacenames_area (self)
- def check_lat_lng_in_scope (self, lat, lng)
- def get_your_placenames_points (self, query)

Public Attributes

- area_flag

  The type flag indicating whether Edinburgh/Cardiff/Southampton.
- unclustered_points

  The points before clustering has taken place.
- yourplacenames_points

  YourPlaceNames clustered points.
- system_generated_points

  Points retrieved from the system after clustering has taken place.
- yourplacenames_area

  Convex hull produced based on the 'YourPlaceNames.com' data.
- system_generated_area

  System produced convex hull.
- inside_count_as_percentage
- overall_percentage_overlap

  The percentage of overlap between the two produced clusters.
- yes_or_no_for_overlap

  Boolean value indicating whether there is an overlap between the two areas.

6.4.1 Detailed Description

```
The class that was implemented to perform experiments.
It will be provided with yourplacenames.com mapping data.
It will be run by a test method which will supply certain
test queries.
```

Definition at line 20 of file YPComparator.py.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 def Comparison.YPComparator.ComparatorYourPlaceNames.__init__ (*self, system_generated_points, unclustered_points, query, flag*)

```
A constructor for the class that performs all comparisons.
@param system_generated_points: The points retrieved from the system
that form the result cluster.
@param unclustered_points: The system points prior to being clustered.
@param query: The Query String.
@param flag: Flag indicating search region.
```

Definition at line 63 of file YPComparator.py.

6.4.3 Member Function Documentation

6.4.3.1 def Comparison.YPComparator.ComparatorYourPlaceNames.calculate_amount_points_inside (*self*)

```
It will loop through the unclustered points and check how many
reside within the cluster that has been provided by 'yourplacenames.com'.
It will save the results as an object property.
```

Definition at line 127 of file YPComparator.py.

6.4.3.2 def Comparison.YPComparator.ComparatorYourPlaceNames.calculate_overlap_percentage (*self*)

```
Retrieves the percentage that one object intersects another.
It will use the yourplacenames.com area as the larger area and check for
an intersection percentage. It will set an object parameter to indicate the
overall percentage.
```

Definition at line 154 of file YPComparator.py.

6.4.3.3 def Comparison.YPComparator.ComparatorYourPlaceNames.check_for_overlap (*self*)

```
A method which will check for an overlap.
@return: True - There is an overlap. False - No overlap.
```

Definition at line 168 of file YPComparator.py.

6.4.3.4 def Comparison.YPComparator.ComparatorYourPlaceNames.check_lat_lng_in_scope (*self, lat, lng*)

```
Check the 'YourPlaceNames.com' points being retrieved are
within the correct search region e.g. Cardiff or Ediburgh.
This is done using a bounding box.

@return True - In scope. False - Not in Scope.
```

Definition at line 203 of file YPComparator.py.

6.4.3.5 def Comparison.YPComparator.ComparatorYourPlaceNames.get_inside_percentage (*self*)

```
Just a .get method.
Get percentage of points that reside inside the yourplacenames.com
convex hull.
@return percentage formatted string.
```

Definition at line 40 of file YPComparator.py.

6.4.3.6 def Comparison.YPComparator.ComparatorYourPlaceNames.get_intersection_percentage_as_string (*self*)

```
A utilities method.
Retrieves the percentage intersection as a string.
This percentage will be in a form such as 0.13 = 13%.
@return A string representing the percentage overlap.
```

Definition at line 50 of file YPComparator.py.

6.4.3.7 def Comparison.YPComparator.ComparatorYourPlaceNames.get_overlap_status (*self*)

```
Just a .get method.
Return whether there is an overlap as a boolean between
the two geo objects.
@return boolean indicating whether there is an overlap
```

Definition at line 28 of file YPComparator.py.

6.4.3.8 def Comparison.YPComparator.ComparatorYourPlaceNames.get_points_as_list (*self, list_of_point_objects*)

```
When the list of points are provided they will be
given in the form of a list of Point objects. This method converts
the points to a list in the form [x,y] that makes it easier to use
with the shapely library.
@param list_of_point_objects: The list of point objects ready to be converted.
@return A list of points in the form [x,y]
```

Definition at line 97 of file YPComparator.py.

6.4.3.9 def Comparison.YPComparator.ComparatorYourPlaceNames.get_your_placenames_points (*self, query*)

```
This was divided from the main run() method as it performed
slightly different functionality. This method will retrieve the points,
convert them to Point objects and add them to a list of objects.
This method will also run the DBScan algorithm on the data.
@param query: The query term.
@return a list in [x,y] format.
```

Definition at line 227 of file YPComparator.py.

6.4.3.10 def Comparison.YPComparator.ComparatorYourPlaceNames.get_yourplacenames_area (*self*)

```
Retrieve the yourplacenames.com area.
@return 'YourPlaceNames.com' area as a convex hull array in the format [x,y].
```

Definition at line 188 of file YPComparator.py.

6.4.3.11 def Comparison.YPComparator.ComparatorYourPlaceNames.start_the_checking_process (*self*)

```
This method will start the checking process.
It will first check for an overlap between the two
area polygons and if there is, continue the checking process.
```

Definition at line 115 of file YPComparator.py.

6.4.3.12 def Comparison.YPComparator.ComparatorYourPlaceNames.test_print_a_polygon (*self, poly*)

```
Print all of the coordinates in a polygon.
```

Definition at line 179 of file YPComparator.py.

6.4.4 Member Data Documentation

6.4.4.1 Comparison.YPComparator.ComparatorYourPlaceNames.area_flag

The type flag indicating whether Edinburgh/Cardiff/Southampton.

Definition at line 73 of file YPComparator.py.

6.4.4.2 Comparison.YPComparator.ComparatorYourPlaceNames.inside_count_as_percentage

Definition at line 88 of file YPComparator.py.

6.4.4.3 Comparison.YPComparator.ComparatorYourPlaceNames.overall_percentage_overlap

The percentage of overlap between the two produced clusters.

Definition at line 89 of file YPComparator.py.

6.4.4.4 Comparison.YPComparator.ComparatorYourPlaceNames.system_generated_area

System produced convex hull.

Definition at line 84 of file YPComparator.py.

6.4.4.5 Comparison.YPComparator.ComparatorYourPlaceNames.system_generated_points

Points retrieved from the system after clustering has taken place.

Definition at line 79 of file YPComparator.py.

6.4.4.6 Comparison.YPComparator.ComparatorYourPlaceNames.unclustered_points

The points before clustering has taken place.

Definition at line 75 of file YPComparator.py.

6.4.4.7 Comparison.YPComparator.ComparatorYourPlaceNames.yes_or_no_for_overlap

Boolean value indicating whether there is an overlap between the two areas.

Definition at line 122 of file YPComparator.py.

6.4.4.8 Comparison.YPComparator.ComparatorYourPlaceNames.yourplacenames_area

Convex hull produced based on the 'YourPlaceNames.com' data.

Definition at line 82 of file YPComparator.py.

6.4.4.9 Comparison.YPComparator.ComparatorYourPlaceNames.yourplacenames_points

YourPlaceNames clustered points.

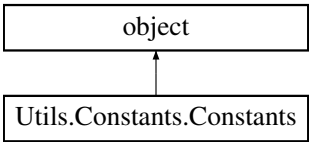
Definition at line 77 of file YPComparator.py.

The documentation for this class was generated from the following file:

- Comparison/YPComparator.py

6.5 Utils.Constants.Constants Class Reference

Inheritance diagram for Utils.Constants.Constants:



Public Member Functions

- def __init__ (self)

Public Attributes

- DB_HOST

Static Public Attributes

- int CARDIFF_TYPE_FLAG = 1
- int EDIN_TYPE_FLAG = 2
- int SOUTHAMPTON_TYPE_FLAG = 3
- string DB_USERNAME = "root"
- string DB_PASSWORD = "NA"
- string DB_NAME = "miner"
- string DB_TWEETS_CARDIFF = "tweets_cardiff"
- string DB_TWEETS_EDIN = "tweets_edin"
- string DB_TWEETS_SOUTH = "tweets_southampton"
- string DB_YOUR_PLACE_NAME = "yourplacename_data"

  Yourplacenames.com data table name.
- string DB_YOURPLACENAME_COMPARISON = "comparison_yourplacenames_table"

  Comparison results tables.
- string DB_OS_COMPARISON = "comparison_os__table"
- int FLAG_COMPARISON_YOURPLACENAME = 1
- int FLAG_COMPARISON_OS = 2
- int VDBSCAN_K_VALUE = 5

  VDBSCAN K VALUE.
- float DEFAULT_DB_SCAN_EPS = 0.5
- int DEFAULT_DB_SCAN_MINPTS = 5
- float MAXIMUM_EPS_VALUE = 1.1
- string GEONAMECONSTANT_COUNTRY_OR_REGION_CODE = 'A'
- string GEONAMECONSTANT_STREAM_OR_LAKE_CODE = 'H'
- string GEONAMECONSTANT_PARKS_OR_AREA_CODE = 'L'
- string GEONAMECONSTANT_CITY_OR_VILLAGE_CODE = 'P'
- string GEONAMECONSTANT_ROAD_OR_RAILWAY_CODE = 'R'
- string GEONAMECONSTANT_SPOT_BUILDING_CODE = 'S'
- float CARDIFF_BOTTOM_LEFT_LAT = 51.384029
- float CARDIFF_BOTTOM_LEFT_LNG = -3.277109
- float CARDIFF_TOP_RIGHT_LAT = 51.584029
- float CARDIFF_TOP_RIGHT_LNG = -3.077109
- float EDIN_BOTTOM_LEFT_LAT = 55.784203
- float EDIN_BOTTOM_LEFT_LNG = -3.346678
- float EDIN_TOP_RIGHT_LAT = 56.084202
- float EDIN_TOP_RIGHT_LNG = -3.046678

6.5.1 Detailed Description

```
Useful constants defined here.
```

Definition at line 13 of file Constants.py.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 def Utils.Constants.Constants.__init__ (*self*)

Definition at line 69 of file Constants.py.

6.5.3 Member Data Documentation

6.5.3.1 float Utils.Constants.Constants.CARDIFF_BOTTOM_LEFT_LAT = 51.384029 [static]

Definition at line 59 of file Constants.py.

6.5.3.2 float Utils.Constants.Constants.CARDIFF_BOTTOM_LEFT_LNG = -3.277109 [static]

Definition at line 60 of file Constants.py.

6.5.3.3 float Utils.Constants.Constants.CARDIFF_TOP_RIGHT_LAT = 51.584029 [static]

Definition at line 61 of file Constants.py.

6.5.3.4 float Utils.Constants.Constants.CARDIFF_TOP_RIGHT_LNG = -3.077109 [static]

Definition at line 62 of file Constants.py.

6.5.3.5 int Utils.Constants.Constants.CARDIFF_TYPE_FLAG = 1 [static]

Definition at line 18 of file Constants.py.

6.5.3.6 Utils.Constants.Constants.DB_HOST

Definition at line 72 of file Constants.py.

6.5.3.7 string Utils.Constants.Constants.DB_NAME = "miner" [static]

Definition at line 24 of file Constants.py.

6.5.3.8 string Utils.Constants.Constants.DB_OS_COMPARISON = "comparison_os__table" [static]

Definition at line 36 of file Constants.py.

6.5.3.9 string Utils.Constants.Constants.DB_PASSWORD = "NA" [static]

Definition at line 23 of file Constants.py.

6.5.3.10 string Utils.Constants.Constants.DB_TWEETS_CARDIFF = "tweets_cardiff" [static]

Definition at line 26 of file Constants.py.

6.5.3.11 string Utils.Constants.Constants.DB_TWEETS_EDIN = "tweets_edin" [static]

Definition at line 27 of file Constants.py.

6.5.3.12 string Utils.Constants.Constants.DB_TWEETS_SOUTH = "tweets_southampton" [static]

Definition at line 28 of file Constants.py.

6.5.3.13 string Utils.Constants.Constants.DB_USERNAME = "root" [static]

Definition at line 22 of file Constants.py.

6.5.3.14 string Utils.Constants.Constants.DB_YOUR_PLACE_NAME = "yourplacename_data" [static]

Yourplacenames.com data table name.

Definition at line 31 of file Constants.py.

6.5.3.15 string Utils.Constants.Constants.DB_YOURPLACENAME_COMPARISON = "comparison_yourplacenames_table" [static]

Comparison results tables.

Definition at line 35 of file Constants.py.

6.5.3.16 float Utils.Constants.Constants.DEFAULT_DB_SCAN_EPS = 0.5 [static]

Definition at line 46 of file Constants.py.

6.5.3.17 int Utils.Constants.Constants.DEFAULT_DB_SCAN_MINPTS = 5 [static]

Definition at line 47 of file Constants.py.

6.5.3.18 float Utils.Constants.Constants.EDIN_BOTTOM_LEFT_LAT = 55.784203 [static]

Definition at line 64 of file Constants.py.

6.5.3.19 float Utils.Constants.Constants.EDIN_BOTTOM_LEFT_LNG = -3.346678 [static]

Definition at line 65 of file Constants.py.

6.5.3.20 float Utils.Constants.Constants.EDIN_TOP_RIGHT_LAT = 56.084202 [static]

Definition at line 66 of file Constants.py.

6.5.3.21 float Utils.Constants.Constants.EDIN_TOP_RIGHT_LNG = -3.046678 [static]

Definition at line 67 of file Constants.py.

6.5.3.22 int Utils.Constants.Constants.EDIN_TYPE_FLAG = 2 [static]

Definition at line 19 of file Constants.py.

6.5.3.23 int Utils.Constants.Constants.FLAG_COMPARISON_OS = 2 [static]

Definition at line 39 of file Constants.py.

6.5.3.24 int Utils.Constants.Constants.FLAG_COMPARISON_YOURPLACENAME = 1 [static]

Definition at line 38 of file Constants.py.

6.5.3.25 string Utils.Constants.Constants.GEONAMECONSTANT_CITY_OR_VILLAGE_CODE = 'P' [static]

Definition at line 55 of file Constants.py.

6.5.3.26 string Utils.Constants.Constants.GEONAMECONSTANT_COUNTRY_OR_REGION_CODE = 'A' [static]

Definition at line 52 of file Constants.py.

6.5.3.27 string Utils.Constants.Constants.GEONAMECONSTANT_PARKS_OR_AREA_CODE = 'L' [static]

Definition at line 54 of file Constants.py.

6.5.3.28 string Utils.Constants.Constants.GEONAMECONSTANT_ROAD_OR_RAILWAY_CODE = 'R' [static]

Definition at line 56 of file Constants.py.

6.5.3.29 string Utils.Constants.Constants.GEONAMECONSTANT_SPOT_BUILDING_CODE = 'S' [static]

Definition at line 57 of file Constants.py.

6.5.3.30 string Utils.Constants.Constants.GEONAMECONSTANT_STREAM_OR_LAKE_CODE = 'H' [static]

Definition at line 53 of file Constants.py.

6.5.3.31 float Utils.Constants.Constants.MAXIMUM_EPS_VALUE = 1.1 [static]

Definition at line 48 of file Constants.py.

6.5.3.32 int Utils.Constants.Constants.SOUTHAMPTON_TYPE_FLAG = 3 [static]

Definition at line 20 of file Constants.py.

6.5.3.33 int Utils.Constants.Constants.VDBSCAN_K_VALUE = 5 [static]

VDBSCAN K VALUE.

Definition at line 42 of file Constants.py.

The documentation for this class was generated from the following file:

- Utils/Constants.py

6.6 Utils.DBAdapter.DBAdapter Class Reference

Public Member Functions

- def __init__ (self)
- def get_all_tweets (self, search_query, type_flag)
- def insert_comparison_tweet (self, query_name, percentage_points_inside, percentage_of_intersection, overlap, table_name)
- def get_yourplacenames_points (self, search_query)

Public Attributes

- constants
- connection

  MYSQLDB CONNECTION.
- result

6.6.1 Detailed Description

```
This is a utilities class. It provides database access methods.

These are all in a separate method for utility purposes. By instatiating
this class a DB Connection is also formed.
@author: Billy Hickman
```

Definition at line 14 of file DBAdapter.py.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 def Utils.DBAdapter.DBAdapter.__init__ (*self*)

Definition at line 23 of file DBAdapter.py.

6.6.3 Member Function Documentation

6.6.3.1 def Utils.DBAdapter.DBAdapter.get_all_tweets (*self, search_query, type_flag*)

```
This a method to retrieve all tweets which match the query.

@param search_query: This is the search query. Tweets that contain this text will be retrieved.
@param type_flag: This indicates the type of table being accessed. E.g. Edinburgh or Cardiff.
@return A cursor containing the results (MySQL DB).
```

Definition at line 35 of file DBAdapter.py.

6.6.3.2 def Utils.DBAdapter.DBAdapter.get_yourplacenames_points (*self, search_query*)

```
Retrieves the yourplacenames.com queries for the search query.
These results will have already been added to the database.

@param search_query: This is the search query. For a point to be
retrieved it will be in the name column.
@return A cursor containing the results (MySQL DB).
```

Definition at line 97 of file DBAdapter.py.

6.6.3.3 def Utils.DBAdapter.DBAdapter.insert_comparison_tweet (*self, query_name, percentage_points_inside, percentage_of_intersection, overlap, table_name*)

```
Once the comparator class has found similarities with the admin
area this method will save it to the database.

@param table_name: The table name to save the results in.
@param query_name: The place name query.
@param percentage_of_intersection: Intersection percentage.
@param overlap: A boolean value indicating whether there is an overlap.
```

Definition at line 64 of file DBAdapter.py.

6.6.4 Member Data Documentation

6.6.4.1 Utils.DBAdapter.DBAdapter.connection

MYSQLDB CONNECTION.

Definition at line 27 of file DBAdapter.py.

6.6.4.2 Utils.DBAdapter.DBAdapter.constants

Definition at line 25 of file DBAdapter.py.

6.6.4.3 Utils.DBAdapter.DBAdapter.result

Definition at line 33 of file DBAdapter.py.

The documentation for this class was generated from the following file:

- Utils/DBAdapter.py

6.7 Mining.DBScan.DBSCAN Class Reference

Public Member Functions

- def __init__ (self, points)
- def set_override_values (self, eps_override, minpoints_override)
- def run_algorithm (self)
- def expand_cluster (self, given_point, neighbour_points)
- def get_points_in_vicinity (self, point)
- def is_a_member_of_an_existing_cluster (self, point)
- def get_used_eps (self)
- def get_used_min_pts (self)

Public Attributes

- constants
  An instance of the Utilities class 'Constants'.
- distance_to_form_a_cluster
  Minimum neighbourhood distance - similar to the EPS value.
- minimum_points_to_form_a_cluster
  Minimum number of points required to form a cluster.
- db_points
  Array of the 'Point' class.
- amount_of_clusters_index
  A rolling count of clusters that have so far been found.
- found_clusters
  An array of clusters found so far.
- methods
  An instance of the 'Methods' utilities class.

6.7.1 Detailed Description

```
This is the DBScan algorithm class.

It utilizes the class 'Marker' (utils) in order to define a point.
It is based loosely on the implementation found at http://iamtawit.blogspot.in/2012/12/dbscan.html.
Reference to the DBScan algorithm can be found at http://en.wikipedia.org/wiki/DBSCAN.
```

Definition at line 14 of file DBScan.py.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 def Mining.DBScan.DBSCAN.__init__ (*self, points*)

```
Constructor.

@param points: An array of instances of the 'Marker' class.
```

Definition at line 23 of file DBScan.py.

6.7.3 Member Function Documentation

6.7.3.1 def Mining.DBScan.DBSCAN.expand_cluster (*self, given_point, neighbour_points*)

```
Expand the cluster that has been found.

It will find any new neighbours of this point.
```

Definition at line 95 of file DBScan.py.

6.7.3.2 def Mining.DBScan.DBSCAN.get_points_in_vicinity (*self, point*)

```
Same as the regionQuery

Gets points that are within the correct distance.
Does not return itself.
@param point: Searching for points around one.
@param index_of_this_point: Index of this point, ensuring it is
not added to the neighbour hood points of itself.
@return An array of points that are within the vicinity.
```

Definition at line 123 of file DBScan.py.

6.7.3.3 def Mining.DBScan.DBSCAN.get_used_eps (*self*)

```
Get previously used EPS.

@return The floating point EPS value used in this instance of the application.
```

Definition at line 166 of file DBScan.py.

6.7.3.4 def Mining.DBScan.DBSCAN.get_used_min_pts (*self*)

```
Get the recently used min pts value.

@return The floating point minPts value used in this instance of the application.
```

Definition at line 174 of file DBScan.py.

6.7.3.5 def Mining.DBScan.DBSCAN.is_a_member_of_an_existing_cluster (*self, point*)

```
Checks whether this point belongs to an existing cluster.

@param point: Marker to check
@return True - It does, False - It does not.
```

Definition at line 149 of file DBScan.py.

6.7.3.6 def Mining.DBScan.DBSCAN.run_algorithm (*self*)

```
Run the algorithm.

Starts by looping through the provided points in the data set.
It also sets the point to visited if it has not been visited yet.
If a cluster is found it will attempt to expand it.

@return found clusters.
```

Definition at line 61 of file DBScan.py.

6.7.3.7 def Mining.DBScan.DBSCAN.set_override_values (*self, eps_override, minpoints_override*)

```
This is a method which is used to override the default algorithm parameters.

This in the case that the user wants to override the default values. It can
also be found if no or more than one clusters have been found in order to override
the default EPS and use an incremented version of this value.

@param eps: EPS override value.
@param minpoints_override: Minimum number of points override.
```

Definition at line 46 of file DBScan.py.

6.7.4 Member Data Documentation

6.7.4.1 Mining.DBScan.DBSCAN.amount_of_clusters_index

A rolling count of clusters that have so far been found.

Definition at line 39 of file DBScan.py.

6.7.4.2 Mining.DBScan.DBSCAN.constants

An instance of the Utilities class 'Constants'.

Definition at line 30 of file DBScan.py.

6.7.4.3 Mining.DBScan.DBSCAN.db_points

Array of the 'Point' class.

Definition at line 37 of file DBScan.py.

6.7.4.4 Mining.DBScan.DBSCAN.distance_to_form_a_cluster

Minimum neighbourhood distance - similar to the EPS value.

Definition at line 32 of file DBScan.py.

6.7.4.5 Mining.DBScan.DBSCAN.found_clusters

An array of clusters found so far.

Definition at line 41 of file DBScan.py.

6.7.4.6 Mining.DBScan.DBSCAN.methods

An instance of the 'Methods' utilities class.

Definition at line 44 of file DBScan.py.

6.7.4.7 Mining.DBScan.DBSCAN.minimum_points_to_form_a_cluster

Minimum number of points required to form a cluster.

Definition at line 34 of file DBScan.py.

The documentation for this class was generated from the following file:

- Mining/DBScan.py

6.8 Mining.GeoNameFinder.GeoNameFinder Class Reference

Public Member Functions

- def __init__ (self, query, db_scan_inst)
- def start_checking_against_geo_names_api (self)
- def get_parameters_for_this_geo_code (self)
- def get_found_fcl_name (self)

Public Attributes

- query_to_find
- db_scan_instance

  DBScan Instance, this is the instance used on this current iteration.
- constants

Constants utilities instance.

- geoname_all_codes
- found_geoname_title

  The GeoNames FCL Title if found - undefined if not.

- is_on_geonames

  Boolean value indicating whether the query has been found on the GeoNames API.

- found_geo_code

  The FCL code found from the GeoNames API - Undefined if not found.

6.8.1 Detailed Description

```
This is a class which is used to establish whether the query
is included on GeoNames and could therefore provide clues about
the scope of the place and therefore clues about parameter selection.

@author Billy Hickman
```

Definition at line 17 of file GeoNameFinder.py.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 def Mining.GeoNameFinder.GeoNameFinder.__init__ (*self, query, db_scan_inst*)

```
Constructor.

Adds the query and associated codes.

@param Query: The place name being used with the system.
@param db_scan_inst: The DBScan algorithm instance.
```

Definition at line 26 of file GeoNameFinder.py.

6.8.3 Member Function Documentation

6.8.3.1 def Mining.GeoNameFinder.GeoNameFinder.get_found_fcl_name (*self*)

```
Get the FCL name that is associated with the code found on the
GeoNames API. This value is then returned as part of the API.

@return The FCL code that this query matches. FCL Code is a GeoName Feature Class.
```

Definition at line 106 of file GeoNameFinder.py.

6.8.3.2 def Mining.GeoNameFinder.GeoNameFinder.get_parameters_for_this_geo_code (*self*)

```
Should a feature class have been established for this GeoNames API then it overrides
the starting parameters of the DBScan algorithm.
@return geoname_code - The GeoCode identifier letter (used for testing)
```

Definition at line 94 of file GeoNameFinder.py.

6.8.3.3 def Mining.GeoNameFinder.GeoNameFinder.start_checking_against_geo_names_api (*self*)

```
This will call the API and check for an exact match. It will use
the first exact match and retrieve the FCL (feature class)
code which matches an aggregation of the feature types.

@return: true - exact match on geonames. false - not an exact match on geonames.
```

Definition at line 57 of file GeoNameFinder.py.

6.8.4 Member Data Documentation

6.8.4.1 Mining.GeoNameFinder.GeoNameFinder.constants

Constants utilities instance.

Definition at line 42 of file GeoNameFinder.py.

6.8.4.2 Mining.GeoNameFinder.GeoNameFinder.db_scan_instance

DBScan Instance, this is the instance used on this current iteration.

Definition at line 39 of file GeoNameFinder.py.

6.8.4.3 Mining.GeoNameFinder.GeoNameFinder.found_geo_code

The FCL code found from the GeoNames API - Undefined if not found.

Definition at line 83 of file GeoNameFinder.py.

6.8.4.4 Mining.GeoNameFinder.GeoNameFinder.found_geoname_title

The GeoNames FCL Title if found - undefined if not.

Definition at line 66 of file GeoNameFinder.py.

6.8.4.5 Mining.GeoNameFinder.GeoNameFinder.geoname_all_codes

Definition at line 44 of file GeoNameFinder.py.

6.8.4.6 Mining.GeoNameFinder.GeoNameFinder.is_on_geonames

Boolean value indicating whether the query has been found on the GeoNames API.

Definition at line 71 of file GeoNameFinder.py.

6.8.4.7 Mining.GeoNameFinder.GeoNameFinder.query_to_find

Definition at line 35 of file GeoNameFinder.py.

The documentation for this class was generated from the following file:

- Mining/GeoNameFinder.py

6.9 Utils.Marker.Marker Class Reference

Public Member Functions

- def __init__ (self, given_lat, given_lng)
- def set_as_visited (self)
- def is_visited (self)
- def set_as_noise (self)

- def is_noise (self)
- def get_longitude (self)
- def get_latitude (self)
- def get_distance_to_k_neighbour (self)
- def calculate_distance_to_kth_neighbour (self, points, k_value, this_index)

Public Attributes

- visited
- latitude
- longitude
- is_noise
- k_distance_neighbour

6.9.1 Detailed Description

```
This is a utilities class used to represent a point.

It is used as the clustering algorithm is being run.

@attention: It is called 'Marker' due to compatibility purposes.
```

Definition at line 10 of file Marker.py.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 def Utils.Marker.Marker.__init__ (*self, given_lat, given_lng*)

Definition at line 18 of file Marker.py.

6.9.3 Member Function Documentation

6.9.3.1 def Utils.Marker.Marker.calculate_distance_to_kth_neighbour (*self, points, k_value, this_index*)

```
In order to implement the 'VDBScan' approach to the algorithm.

This method calculates the distance to the Kth nearest neighbour.
It uses the point array (handed as a parameter) to do this.

@param points - Points array which includes all points (Points object) on the map.
@param k_value - The value of the Kth nearest neighbour.
@param this_index: Index of this point in the point array.
@return void


@deprecated: As VDBScan was not adopted as a method this method has since been deprecated.
```

Definition at line 67 of file Marker.py.

6.9.3.2 def Utils.Marker.Marker.get_distance_to_k_neighbour (*self*)

```
@return KM distance to K'th neighbour.
```

Definition at line 61 of file Marker.py.

6.9.3.3 def Utils.Marker.Marker.get_latitude (*self*)

```
@return Latitude decimal.
```

Definition at line 55 of file Marker.py.

6.9.3.4 def Utils.Marker.Marker.get_longitude (*self*)

```
@return Longitude decimal.
```

Definition at line 49 of file Marker.py.

6.9.3.5 def Utils.Marker.Marker.is_noise (*self*)

```
@return Boolean indicating whether this point is noise.
```

Definition at line 43 of file Marker.py.

6.9.3.6 def Utils.Marker.Marker.is_visited (*self*)

```
@return Boolean indicating whether this point has been visited.
```

Definition at line 31 of file Marker.py.

6.9.3.7 def Utils.Marker.Marker.set_as_noise (*self*)

```
Set the point as noise.
```

Definition at line 37 of file Marker.py.

6.9.3.8 def Utils.Marker.Marker.set_as_visited (*self*)

```
Set the point as visited.
```

Definition at line 25 of file Marker.py.

6.9.4 Member Data Documentation

6.9.4.1 Utils.Marker.Marker.is_noise

Definition at line 22 of file Marker.py.

6.9.4.2 Utils.Marker.Marker.k_distance_neighbour

Definition at line 23 of file Marker.py.

6.9.4.3 Utils.Marker.Marker.latitude

Definition at line 20 of file Marker.py.

6.9.4.4 Utils.Marker.Marker.longitude

Definition at line 21 of file Marker.py.

6.9.4.5 Utils.Marker.Marker.visited

Definition at line 19 of file Marker.py.

The documentation for this class was generated from the following file:

- Utils/Marker.py

6.10 Utils.Methods.Methods Class Reference

Public Member Functions

- def get_distance_between_two_points (self, point, temporary_point)

6.10.1 Detailed Description

```
A utilities class of useful methods.
```

Definition at line 4 of file Methods.py.

6.10.2 Member Function Documentation

6.10.2.1 def Utils.Methods.Methods.get_distance_between_two_points (*self, point, temporary_point*)

```
Harvesine distance between two points.

This is included as a Utils method due to its use in multiple classes.

@attention: This returns the distance between the two points in KM.

@param point - One point
@param temporary_point - Second point
@return km distance between two points.
```

Definition at line 9 of file Methods.py.

The documentation for this class was generated from the following file:

- Utils/Methods.py

Chapter 7

File Documentation

7.1 Comparison/__init__.py File Reference

Namespaces

- Comparison

7.2 Main/__init__.py File Reference

Classes

- class Main.ApplicationEntry

Namespaces

- Main

Variables

- tuple Main.CURRDIR = os.path.dirname(inspect.getfile(inspect.currentframe()))
- tuple Main.PARENTDIR = os.path.dirname(CURRDIR)
- tuple Main.urls
- tuple Main.app = web.application(urls, globals())

  API instance from 'web.py'.

7.3 Mining/__init__.py File Reference

Namespaces

- Mining

7.4 Utils/__init__.py File Reference

Namespaces

- Utils

7.5 Comparison/OSComparator.py File Reference

Classes

- class Comparison.OSComparator.ComparatorOrdnanceSurvey

Namespaces

- Comparison.OSComparator

Functions

- def Comparison.OSComparator.run_test_query ()

7.6 Comparison/YPComparator.py File Reference

Classes

- class Comparison.YPComparator.ComparatorYourPlaceNames

Namespaces

- Comparison.YPComparator

Functions

- def Comparison.YPComparator.run_test_query ()

7.7 Main/AnalyserModule.py File Reference

Classes

- class Main.AnalyserModule.Analyser

Namespaces

- Main.AnalyserModule

7.8 Mining/DBScan.py File Reference

Classes

- class Mining.DBScan.DBSCAN

Namespaces

- Mining.DBScan

7.9 Mining/GeoNameFinder.py File Reference

Classes

- class Mining.GeoNameFinder.GeoNameFinder

Namespaces

- Mining.GeoNameFinder

7.10 Utils/Constants.py File Reference

Classes

- class Utils.Constants.Constants

Namespaces

- Utils.Constants

7.11 Utils/DBAdapter.py File Reference

Classes

- class Utils.DBAdapter.DBAdapter

Namespaces

- Utils.DBAdapter

7.12 Utils/Marker.py File Reference

Classes

- class Utils.Marker.Marker

Namespaces

- Utils.Marker

7.13 Utils/Methods.py File Reference

Classes

- class Utils.Methods.Methods

Namespaces

- Utils.Methods

Index