



Sii-Mobility

Supporto di Interoperabilità Integrato per i Servizi al Cittadino e alla Pubblica Amministrazione

Trasporti e Mobilità Terrestre, SCN_00112

Deliverable ID: DE3.16

**Titolo: Specifica di dettaglio dei moduli di integrazione
con applicazioni fisse e mobili**

Data corrente	
Versione (solo il responsabile puo' cambiare versione)	
Stato (draft, final)	Draft , V1
Livello di accesso (solo consorzio, pubblico)	Pubblico
WP	
Natura (report, report e software, report e HW..)	
Data di consegna attesa	
Data di consegna effettiva	
Referente primario, coordinatore del documento	<Nome> <Cognome> <email>
Contributor	<Nome> <Cognome> <email>, <Nome> <Cognome> <email>, <Nome> <Cognome> <email>
Coordinatore responsabile del progetto	Paolo Nesi, UNIFI, paolo.nesi@unifi.it

Sommario

	Background Colour Meanings in documents	3
	Rules for File naming and management	4
1	Executive Summary	6
2	Service MAP vs Smart City API.....	6
3	Smart City: API v1	7
3.1	Basics.....	7
3.2	Service search near GPS position.....	9
3.3	Service search within a GPS area.....	12
3.4	Service search within a WKT described area.....	14
3.5	Service search within a stored WKT described area.....	15
3.6	Service search by municipality.....	16
3.7	Service search by query id.....	17
3.8	Full text search	18
3.9	Event search	19
3.10	Address and geometry search by GPS.....	20
3.11	Service info.....	21
3.12	Generic service	23
3.12.1	Event	24
3.12.2	Parking service	24
3.12.3	Traffic sensor	26
3.12.4	Weather Forecast.....	27
3.12.5	Bus stop.....	28
3.12.6	Fuel Station	30
3.12.7	First aid (added with RESOLUTE project).....	32
3.12.8	Smart waste container (added with REPLICATE project)	34
3.12.9	Smart bench (added with REPLICATE project).....	35
3.12.10	Smart irrigator (added with REPLICATE project)	36
3.12.11	Energy meter (added with REPLICATE project)	37
3.12.12	Recharge station (added with REPLICATE project)	37
3.12.13	Smart street light (added with REPLICATE project)	37
3.12.14	Air quality monitoring station	37
3.13	Public transport API	37
3.13.1	Agency list	37
3.13.2	(Bus) Lines list.....	38
3.13.3	(Bus) Routes list.....	39
3.13.4	(Bus) Stop list.....	40
3.13.5	Search (Bus) Routes in a geographic area	41
3.13.6	Estimated Bus position.....	43
3.14	Feedback API.....	44
3.14.1	Rating and comment API.....	44
3.14.2	Service Photo API.....	45
3.14.3	Last contributions API	45
3.15	Recommender API.....	46
3.16	Shortest path finder API	48
3.17	Image caching API	48
4	Linked data and SPARQL access	49
5	Bibliografia	49

6	Acronimi	49
---	----------------	----

NOTE DI COMPILAZIONE E LETTURA

The deliverable should to be changed only by using Windows Microsoft word DOCX format, office 2008, **not MAC**.

Document responsible has to:

- Create the index and assign the work by writing the responsible name for each section as follow: (first, second, etc.. Responsible)
- Prepare a schedule for completing the work
- Change section background colors
- Change the version number of the Document, he is the only one entitled to do it
- Post on the WEB a new version integrated with contributions and notify important issues to the mailing list, , he is the only one entitled to do it
- Make comments asking for more contributions, changing the doc structure, etc.

Il **Responsabile della sezione** viene identificato dal PRIMO acronimo riportato nel titolo della sezione stessa, seguito dagli altri acronimi dei partner coinvolti nella sezione. Il partner responsabile della sezione:

- Deve fare la prima stesura della sezione. Deve essere fatto in tempo secondo il programma definito dal documento. L' elenco dei partner che devono contribuire viene riportato dopo il suo acronimo.
- Deve coordinare la stesura e fornire un nuovo termine per la successiva iterazione.
- Passare ad azzurro la sessione quando il è considerata finalizzata

Tenete conto che:

- Colori per la gestione delle modifiche e dei contributi sono riportati qui di seguito
- **Non dimenticate di tenere traccia delle modifiche utilizzando lo strumento delle versioni di MS Word**

Background Colour Meanings in documents

- **Comments** and other aspects to help the editors and to write suggestion and discussion streams. Hot questions to be posed to the DOC responsible. The Yellow will be removed by the Project Coordinator only when the document is finalised. If the your is a comment, write you NAME and Affiliazione and then the comment.
- ARE YELLOW also Sections that have to be moved in other places they have to read by all responsible to see if they have to be relocated.
- Call for **CONTRIBUTION**: Section or segment to be revised by the section responsible and by all partners involved. The section responsible is the first partner mentioned in the section title. When the responsible is missing see the responsible of the upper level section.
- Call for **SECTION FINALISATION**: A version accepted by the section responsible, it passed a green phase. The decision to advance to cyan the section or segment has been performed by the section responsible or by a contributor. The partners involved in the section can still contribute to make it better and finalise the section.

- **CLOSED SECTION by its RESPONSIBLE:** To be still revised by the Document responsible, it has passed the cyan phase by the section responsible. This color is set by the Section responsible when the section is completely finalised. When this color is set, the Document Coordinator can decide to pass it in RED closing the section or to change the status if more work is needed.
- passed at clear by the Project Coordinator, so that close to be closed. This color (no color) is used typically in the closing phase of the document or when the document is restarted/reviewed at a major verification meeting of the project. The document can be cleaned from the colours only by the Project Coordinator.

Rules for File naming and management

The file name structure should not to be changed. If all the partners will try to enforce their standard for defining the file names we will have more than 20 different formats without any possibility of understanding if two files are different versions of the same one or is simply the same file. Moreover, please note that your changes in the file name structure will constraint all the other partners to change the file name once received. **Please don't do it.**

The responsible of the doc produces the file with the name with the following structure

Sii-Mobility-<DeliverableID>-<short title>-v<version>.DOCX

Such as:

Sii-Mobility -DE3-1-2-Framework-and-Tools-Specifications-v0-4.DOCX

- 1) use MSword on Windows only
- 2) do not change the file name structure
- 3) do not put in the file name spaces “ ”, or multiple dots “.” Or other symbols of separation different from “-“
- 4) the file name has to contain only one dot, multiple dots are not allowed
- 5) the version (both major and minor) **can be changed only by the document responsible.** The responsible is reported in the first page cover.
- 6) Major and minor version increment is decided and performed only by the doc responsible.
- 7) to put in Appendix Files media information and large segments of information that are already consolidated and that are not part of the major flow of the doc, for example with a name like:

Sii-Mobility -<Deliverable ID>-<TITLE>-APP-A-UML-v0-4.ppt

<Deliverable ID>, is the number of the deliverable according to the Annex I

<Title>, is the official title of the deliverable according to Annex I

0-4, is the version of the document, where 0 is the major number and 4 is the minor number, Major number can be from 0 to 99, while minor only from 0 to 9.

- 8) When you contribute as a contributor not as a document responsible you have to
 - 1) in any version of the doc, don't wait for the last minute or day it will be too late
 - 2) put your acronym at the end such as in the example:
Sii-Mobility -DE3-1-2-Framework-and-Tools-Specifications-v0-4-DISIT.DOCX

The integration of the contributions is performed by the document responsible

- PLEASE send the changes/contributions in time

- to contribute to a version you have to send it back to the document responsible and in copy to the Project Manager. Do not post the document on the Portal. It can be posted on the portal only by the Document Responsible or by the Project Manager.

This rule is also valid for Cost Statements and any other document in Sii-Mobility.

Please don't add changes and send back the file with the same file name.

Any change on the deliverable has to be performed by using the versioning mechanism of WORD or by marking with a different colour the changes performed. In that case, the file has to be returned back to the reflector and thus to the deliverable responsible as:

Sii-Mobility -DEX-X-X-title-vY-Z-*partner*.doc

Where *partner* is the acronym of the partner or the name of the LAB (in the case of multiple lab in the same partner) or the name of the person that has performed that changes, if the person is well known. The partner or lab name are preferred.

Questa parte sarà rimossa vicino alla chiusura del documento.

1 Executive Summary

2 Service MAP vs Smart City API

ServiceMap is a service and tool to pose geographic queries and see the knowledge base produced by the harvesting process based on [Km4City](#) model that includes: Street Graph from Tuscany region, Open Data from Florence Municipality, traffic monitoring, geo and weather forecast information from LAMMA, traffic sensors, services, events, parkings, real time of busses, from Florence Municipality, etc. Some of these data provide real time information as the status of the busses on the bus lines in Florence, parking sensors in Florence and Empoli, traffic sensors in Florence and Empoli, events, and the weather forecast in Tuscany. The ServiceMap accesses to an RDF Store based on [Km4City ontology and model](#).

Service map API can be very useful for shortening the time for developers that want to realize Web or Mobile Apps, exploiting the km4city data. As a service, different kinds of queries can be saved from the ServiceMap when you find the icon disk. They are located on Services, Weather panel, selection panel and on the general web page. Then, click them and fill the form, thus the servicemap will perform two issues (1) save the query performed in a database for your further reuse, (2) send to you an email with a set of links, which are the specific REST calls that you can use to invoke the ServiceMap API from Web and Mobile applications to obtain the service you requested.

In addition, it is possible to take the service map and put in a third party web page by using the embedded functionality that can be activated by the icon on the lower right corner. See an example of Embedding for the [services close to DISIT Lab from this link](#).

The call produced by ServiceMap and received by email will be different according to the icon you have selected, requesting: Services, Weather panel, selection panel and on the general web page. You can copy paste these links into your application shortening the programming time, since the REST call or query are visually programmed. You can avoid learning SPARQL query language. You can develop applications that can contain queries that can be directly updated by you on ServiceMap without redeploying the application on the mobile market. The provided mobile app source code can be used on iOS, Android, Windows Phone, BlackBerry, etc. You can access to the Sii-Mobility Km4City based RDF Store model via the browser <http://log.disit.org>.

APIs have been realized in order to cope with new possibilities and emerging needs of contextualize content, re-organizing information about services, keep real time data, and last but not least Re-use Open Data from Service Map to re-contextualize them in Mobile or Web Apps. These API are for those developers who want to exploit Open Data to create their own application.

The usage of the Smart City API is regulated by the so-called Affiliation Agreement. The Agreement has to be signed and give you access to more technical information and allows you to use the [Smart City API](#), test and trial them without need of understanding fully the technical details and reinstalling the km4City platform in your premise.

The **Affiliation Agreement** is available in [ITALIAN](#) and [ENGLISH](#) languages. On the contrary, if you would like to use the tools starting from the Open Source Version, installing them and using

them, you do not need to contact us. Just do it! We are happy to help you also in that case up to a reasonable amount of effort. Please note that you can access for free at our tutorial and training days.

In Section 2 there is a description of realization, semantic and uses of the REST APIs.

In alternative you can access directly to the [km4city RDF store](#) by using this link and interface for developers (see Section 3 for some details).

3 Smart City: API v1

This section provides a description of the API at version 1, an older version is available at <http://servicemap.disit.org/WebAppGrafo/api/> that is used by old applications and it no more maintained please don't use them.

WARNING: These APIs are still under development and may change in the future (for bugs solving and improvements) however we will try to keep them backward compatible and introduce new parameters and new properties in the JSON objects but not change parameters names or property names. When it will not possible to be backward compatible we will switch to version 2.

3.1 Basics

The APIs are accessible mainly via HTTP GET requests at specific URLs with specific parameters provided in the query string. Query parameters are case sensitive (e.g. use maxDists and not maxdists). The “format” parameter in many cases can be equal to html or json (and json is assumed if it is not provided) to provide the result as machine readable JSON or as a human readable web page. Most APIs accept an optional user identifier (uid) that should be provided to identify the device (and indirectly the user) making the requests. The uid should be a unique identifier, currently the uid is generated as a SHA256 hash of the device uuid generated by cordova device plugin (see <https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-device/>). The history of user requests is used to produce suggestions and user engagements.

The *multimedia* property provided by some APIs contain a URL to a multimedia file that is in many cases no more available, a caching service for images was setup because the images were too large for mobiles and now using this cache is currently the only way to retrieve these images. See the multimedia caching API to see how to use these images. Unfortunately the cache was realized only for images and thus pdf files and audio files are no more available.

The requests to the API are CORS enabled thus APIs can be used cross domain from other sites. Currently no API key or authentication is needed but this may change in the future.

The following table reports a list of the APIs currently available and reports:

- if the API call can be visually generated from the ServiceMap user interface using a Save button,
- if the API can be used to embed in a HTML iframe the results and
- where the API is currently available if in the *Production* site (<http://servicemap.disit.org/WebAppGrafo>) or on the *Test* site (<http://www.disit.org/ServiceMap>).

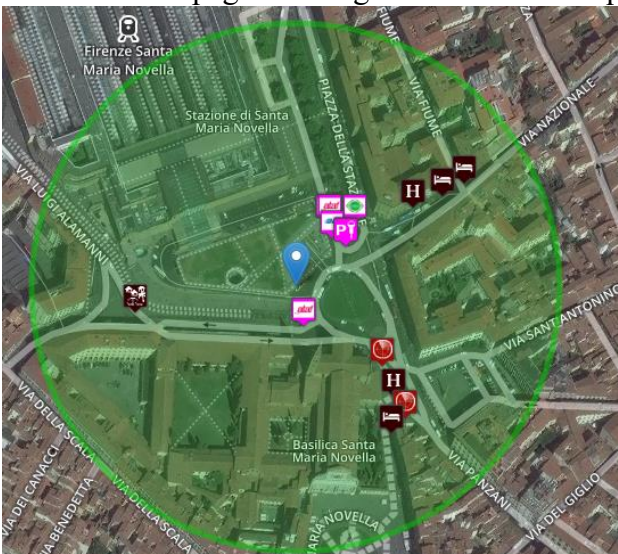
Description	API	Save on ServiceMap	embed	where
Service search near GPS position	Y	Y	Y	Prod & Test
Service search within a GPS area	Y	Y	Y	Prod & Test
Service search within a WKT described area	Y	N	Y	Prod & Test
Service search within a stored WKT described area	Y	Y	Y	Prod & Test
Service search by municipality	Y	Y	Y	Prod & Test
Service search by query id	Y		Y	Prod & Test
Full text search	Y	Y	Y	Prod & Test
Event search	Y	Y?	Not yet	Prod & Test
Address and geometry search by GPS	Y	Not yet	Not yet	Prod & Test
Service info	Y	Y	Y	Prod & Test
Generic Service	Y	Y	Y	Prod & Test
Event	Y	Y	Y	Prod & Test
Parking service	Y	Y	Y	Prod & Test
Traffic sensor	Y	Y	Y	Prod & Test
Weather Forecast	Y	Y	N	Prod & Test
Bus station	Y	Y	Y	Prod & Test
Fuel Station	Y	Y	Y	Test
First aid	Y	Y	Y	Test
Smart waste container	Y	Y	Y	Test
Smart bench	Y	Y	Y	Test
Smart irrigator	Y	Y	Y	Test
Energy meter	Not yet	Not yet	Not yet	-
Recharge station	Not yet	Not yet	Not yet	-
Smart street light	Not yet	Not yet	Not yet	-
Air quality monitoring station	Not yet	Not yet	Not yet	-
(Bus) Agency list	Y	N	N	Prod & Test
(Bus) Lines list	Y	N	N	Prod & Test
(Bus) Routes list	Y	Not yet	Not yet	Prod & Test
(Bus) Stop list	Y	Not yet	Not yet	Prod & Test
Search (Bus) Routes in a geographic area	Y	Not yet	Not yet	Prod & Test
Estimated Bus position	Y	Not yet	Y	Prod & Test
Rating and comment API	Y	N	N	Prod & Test
Service Photo API	Y	N	N	Prod & Test
Last contributions API	Y	N	N	Prod & Test
Recommender API	Y	N	N	Prod & Test
Shortest path finder API	Not yet	Not yet	Not yet	-
Image caching API	Y	N	Y	Prod & Test

Note: For APIs supporting format “html” the following additional optional parameters may be used:

- *map*: to set the type of map to be used (“satellite”, “streets” or “grayscale”);
- *controls*: to control the appearance of the controls on the left and right of the page, it can be “hidden” or “false” to be not visible or “collapsed” to be collapsed;

- *info*: to control the appearance of the info tab on the lower left of the page, it can be “hidden” or “false” to be not visible or “collapsed” to be collapsed;

3.2 Service search near GPS position

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/
it allows to retrieve the set of services that are near a given GPS position. The services can be filtered as belonging to specific categories (e.g. Accomodation, Hotel, Restaurant etc), or having specific words in any textual field. It can also be used to find services that have a WKT spatial description that contains a specific GPS position.	
Parameters:	
<i>selection</i>	<latitude>;<longitude> of the GPS position
<i>categories</i>	the list of categories of the services to be retrieved, if omitted all kinds of services are returned. It can contain macro categories or categories, if a macro category is specified all categories in the macro category are used. The complete list of categories and macro categories can be retrieved on servicemap.disit.org .
<i>text</i>	words in this parameter are used to retrieve services that contain all these words in any textual description associated with the service.
<i>maxDists</i>	maximum distance from the GPS position of the services to be retrieved, expressed in Km (0.1 is used if parameter is missing) if it is equal to “inside” it searches for services with a WKT geometry that contains the specified GPS position (e.g a park)
<i>maxResults</i>	maximum number of results to be returned (if parameter is missing 100 is assumed), if it is 0 all results are returned.
<i>lang</i>	ISO 2 chars language code (e.g. “it”, “en”, “fr”, “de”) to be used for returned descriptions if available in multiple languages. Currently for languages other than “it” and “en” it returns “en” descriptions. (if parameter is missing “en” is assumed)
<i>geometry</i>	true/false, if true it returns a “hasGeometry” property for each service stating if the service has a complex WKT geometries (linestring, polygon) associated with it (if parameter is missing “false” is assumed)
<i>uid</i>	optional user identifier
<i>format</i>	html or json
Results:	
when format = “html” it produces a web page showing the results of the query, like the following:	
	

when format = “json” it returns the services split in three sections (BusStops , SensorSites, Services). Each section is provided as GeoJSON “FeatureCollection”, additionally in each section the “fullCount” property reports the full number of results available matching the query, for example:

```
{
  "BusStops": {
    "fullCount": 26,
    "type": "FeatureCollection",
    "features": [{
      "geometry": {
        "type": "Point",
        "coordinates": [11.249078, 43.775326]
      },
      "type": "Feature",
      "properties": {
        "name": "Stazione Abside S.M.N.",
        "typeLabel": "Fermata",
        "tipo": "fermata",
        "serviceType": "TransferServiceAndRenting_BusStop",
        "busLines": "13 - 36 - 37",
        "serviceUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Stop_FM0328_5",
        "agency": "Ataf&Linea",
        "agencyUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172",
        "photoThumbs": []
      }
    },
    { "id": 1 },
    ... ]
  },
  "SensorSites": {
    "fullCount": 3,
    "type": "FeatureCollection",
    "features": [{
      "geometry": {
        "type": "Point",
        "coordinates": [11.24982, 43.77505]
      },
      "type": "Feature",
      "properties": {
        "name": "FI055ZTL00101",
        "tipo": "sensore",
        "typeLabel": "Sensore",
        "serviceType": "TransferServiceAndRenting_SensorSite",
        "serviceUri": "http://www.disit.org/km4city/resource/FI055ZTL00101",
        "photoThumbs": []
      }
    },
    { "id": 1 },
    ... ]
  },
  "Services": {
    "fullCount": 84,
    "type": "FeatureCollection",
    "features": [{
      "geometry": {
        "type": "Point",
        "coordinates": [11.249473, 43.775867]
      },
      "type": "Feature",
      "properties": {
        "name": "Parcheggio Stazione Firenze S.M.N.",
        "tipo": "Parcheggio_auto",
        "typeLabel": "Parcheggio auto",
        "serviceType": "TransferServiceAndRenting_Car_park",
        "serviceUri": "http://www.disit.org/km4city/resource/CarParkStazioneFirenzeS.M.N.",
        "multimedia": ""
      }
    },
    { "id": 1 },
    ... ]
  }
}
```

Examples:

- Search for Accommodation, bus stop, sensor site or car park within 200m

http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=43.7756;11.2490&categories=Accommodation;BusStop;SensorSite;Car_park&maxResults=10&maxDists=0.2&lang=it&format=json

- **Any entertainment service within 200m**

<http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=43.7756;11.2490&categories=Entertainment&maxResults=20&maxDists=0.2&lang=it&format=json&geometry=true>

```
{
  "Services": {
    "fullCount": 8,
    "type": "FeatureCollection",
    "features": [{
      "geometry": {
        "type": "Point",
        "coordinates": [11.24851, 43.77566]
      },
      "type": "Feature",
      "properties": {
        "name": "Giardino di piazza della Stazione",
        "tipo": "Aree_verdi",
        "typeLabel": "Aree verdi",
        "serviceType": "Entertainment_Green_areas",
        "hasGeometry": true,
        "serviceUri": "http://www.disit.org/km4city/resource/e62bc5f14bd412db00fcdcd6f9506857",
        "multimedia": ""
      },
      "id": 1
    }, {
      "geometry": {
        "type": "Point",
        "coordinates": [11.249722, 43.77561]
      },
      "type": "Feature",
      "properties": {
        "name": "Spartitraffico di piazza della Stazione",
        "tipo": "Aree_verdi",
        "typeLabel": "Aree verdi",
        "serviceType": "Entertainment_Green_areas",
        "hasGeometry": true,
        "serviceUri": "http://www.disit.org/km4city/resource/37a2cdb39f7c8e86c55990b4f3125256",
        "multimedia": ""
      },
      "id": 2
    }, {
      "geometry": {
        "type": "Point",
        "coordinates": [11.249624, 43.77658]
      },
      "type": "Feature",
      "properties": {
        "name": "SCUDERIA DEL BEJ DI SIVORI GIOVAN BATTISTA E C. - S.A.S.",
        "tipo": "Societa_sportive",
        "typeLabel": "Societa' sportive",
        "serviceType": "Entertainment_Sports_clubs",
        "hasGeometry": false,
        "serviceUri": "http://www.disit.org/km4city/resource/0a98b2ea221ba49356c20bed3c7b8f38",
        "multimedia": ""
      },
      "id": 3
    }
  ]
}
```

- **Any service whose geometry contains GPS position**

<http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=43.7754;11.2494&categories=Service&maxResults=20&maxDists=inside&lang=it&format=json&geometry=true>

```
{
  "Services": {
    "fullCount": 6,
```

```

"type": "FeatureCollection",
"features": [{
  "geometry": {
    "type": "Point",
    "coordinates": [11.249722, 43.77561]
  },
  "type": "Feature",
  "properties": {
    "name": "Spartitraffico di piazza della Stazione",
    "tipo": "Aree_verdi",
    "typeLabel": "Aree verdi",
    "serviceType": "Entertainment_Green_areas",
    "hasGeometry": true,
    "serviceUri": "http://www.disit.org/km4city/resource/37a2cdb39f7c8e86c55990b4f3125256",
    "multimedia": ""
  },
  "id": 1
}]
}

```

- **Accommodation within 1Km with “casa di dante” in a textual description**

<http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=43.7754;11.2494&categories=Accommodation&maxResults=2&maxDists=1&lang=it&format=json&text=casa%20di%20dante>

```

{
  "Services": {
    "fullCount": 2,
    "type": "FeatureCollection",
    "features": [{
      "geometry": {
        "type": "Point",
        "coordinates": [11.256365, 43.771023]
      },
      "type": "Feature",
      "properties": {
        "name": "CASA_DI_DANTE",
        "tipo": "Affittacamere",
        "typeLabel": "Affittacamere",
        "serviceType": "Accommodation_Boarding_house",
        "serviceUri": "http://www.disit.org/km4city/resource/c1cd4b12fabce2d9b3a1527fd5a7be79",
        "multimedia": ""
      },
      "id": 1
    }, {
      "geometry": {
        "type": "Point",
        "coordinates": [11.256365, 43.771023]
      },
      "type": "Feature",
      "properties": {
        "name": "CASA_DI_DANTE",
        "tipo": "Affittacamere",
        "typeLabel": "Affittacamere",
        "serviceType": "Accommodation_Boarding_house",
        "serviceUri": "http://www.disit.org/km4city/resource/8cb399e95b39475a9838eefa8ff5e683",
        "multimedia": ""
      },
      "id": 2
    }]
  }
}

```

Notes:

3.3 Service search within a GPS area

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/
-----	---

it allows to retrieve the set of services that are inside a rectangular area. The services can be filtered as belonging to specific categories (e.g. Accommodation, Hotel, Restaurant etc), or having specific words in any textual field.

Parameters:

<i>selection</i>	<lat1>;<lng1>;<lat2>;<lng2> are two GPS coordinates describing a rectangle where (lat1,lng1) is a south west point and (lat2, lng2) is a north east point.
<i>categories</i>	the list of categories of the services to be retrieved, if omitted all kinds of services are returned. It can contain macro categories or categories, if a macro category is specified all categories in the macro category are used. The complete list of categories and macro categories can be retrieved on servicemap.disit.org .
<i>text</i>	words in this parameter are used to retrieve services that contain all these words in any textual description associated with the service.
<i>maxResults</i>	maximum number of results to be returned (if parameter is missing 100 is assumed), if it is 0 all results are returned.
<i>lang</i>	ISO 2 chars language code (e.g. “it”, “en”, “fr”, “de”) to be used for returned descriptions if available in multiple languages. Currently for languages other than “it” and “en” it returns “en” descriptions. (if parameter is missing “en” is assumed)
<i>geometry</i>	true/false, if true it returns a “hasGeometry” property for each service stating if the service has a complex WKT geometries (linestring, polygon) associated with it (if parameter is missing “false” is assumed)
<i>uid</i>	optional user identifier
<i>format</i>	html or json

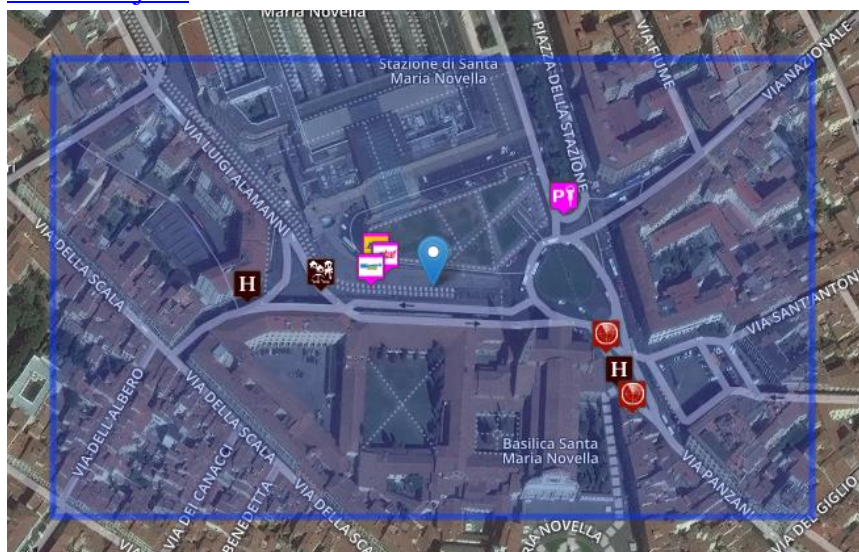
Results:

the results format is the same as the previous API

Examples:


- **Search for an accommodation, bus stop, sensor site or car park in a GPS area**

http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=43.7741;11.2453;43.7768;11.2515&categories=Accommodation;BusStop;SensorSite;Car_park&maxResults=10&lang=it&format=json



Notes:

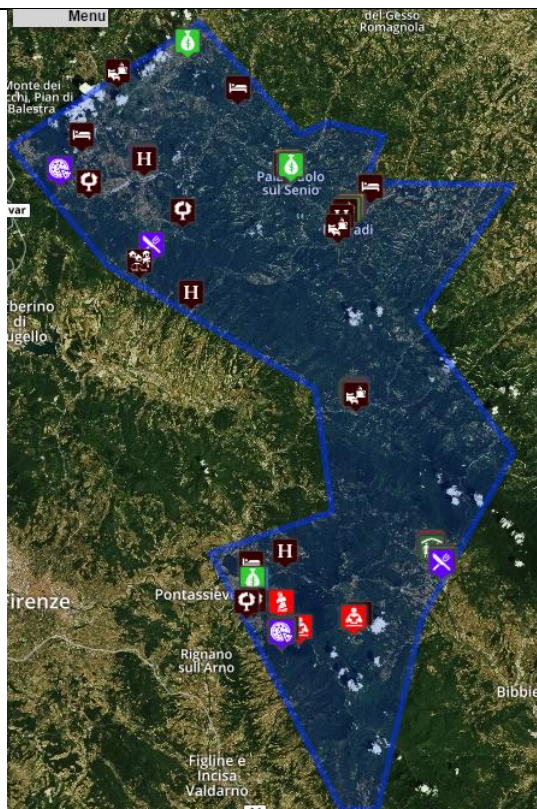
3.4 Service search within a WKT described area

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/
it allows to retrieve the set of services that are inside a geographic region described using the Well Known Text (WKT) format. The services can be filtered as belonging to specific categories (e.g. Accomodation, Hotel, Restaurant etc), or having specific words in any textual field.	
Parameters:	
<i>selection</i>	wkt:<WKT string> describes the geographic region as WKT string.
<i>categories</i>	the list of categories of the services to be retrieved, if omitted all kinds of services are returned. It can contain macro categories or categories, if a macro category is specified all categories in the macro category are used. The complete list of categories and macro categories can be retrieved on servicemap.disit.org .
<i>text</i>	words in this parameter are used to retrieve services that contain all these words in any textual description associated with the service.
<i>maxResults</i>	maximum number of results to be returned (if parameter is missing 100 is assumed), if it is 0 all results are returned.
<i>lang</i>	ISO 2 chars language code (e.g. “it”, “en”, “fr”, “de”) to be used for returned descriptions if available in multiple languages. Currently for languages other than “it” and “en” it returns “en” descriptions. (if parameter is missing “en” is assumed)
<i>geometry</i>	true/false, if true it returns a “hasGeometry” property for each service stating if the service has a complex WKT geometries (linestring, polygon) associated with it (if parameter is missing “false” is assumed)
<i>uid</i>	optional user identifier
<i>format</i>	html or json
Results:	
the results format is the same as the previous API	
Examples:	
<ul style="list-style-type: none"> to write a WKT string the following service can be used https://arthur-e.github.io/Wicket/sandbox-gmaps3.html Search for any service in a WKT area POLYGON((11.25539 43.77339,11.25608 43.77348,11.25706 43.77362,11.25759 43.77328,11.25755 43.77291,11.25675 43.77260,11.25536 43.77270,11.25539 43.77339)) http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=wkt:POLYGON((11.25539%2043.77339,11.25608%2043.77348,11.25706%2043.77362,11.25759%2043.77328,11.25755%2043.77291,11.25675%2043.77260,11.25536%2043.77270,11.25539%2043.77339))&categories=Service&maxResults=0&lang=it&format=html 	
	

Bugs: the html version may not consider all the parameters
--

3.5 Service search within a stored WKT described area

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/
it allows to retrieve the set of services that are inside a geographic region described using the Well Known Text (WKT) format, by referring to the WKT with an identifier provided when the WKT is stored. The services can be filtered as belonging to specific categories (e.g. Accommodation, Hotel, Restaurant etc), or having specific words in any textual field.	
Parameters:	
<i>selection</i>	<i>geo:<geo_id></i> where <geo_id> identifies a WKT string stored on the server.
<i>categories</i>	the list of categories of the services to be retrieved, if omitted all kinds of services are returned. It can contain macro categories or categories, if a macro category is specified all categories in the macro category are used. The complete list of categories and macro categories can be retrieved on servicemap.disit.org .
<i>text</i>	words in this parameter are used to retrieve services that contain all these words in any textual description associated with the service.
<i>maxResults</i>	maximum number of results to be returned (if parameter is missing 100 is assumed), if it is 0 all results are returned.
<i>lang</i>	ISO 2 chars language code (e.g. “it”, “en”, “fr”, “de”) to be used for returned descriptions if available in multiple languages. Currently for languages other than “it” and “en” it returns “en” descriptions. (if parameter is missing “en” is assumed)
<i>geometry</i>	true/false, if true it returns a “hasGeometry” property for each service stating if the service has a complex WKT geometries (linestring, polygon) associated with it (if parameter is missing “false” is assumed)
<i>uid</i>	optional user identifier
<i>format</i>	html or json
Results:	
the results format is the same as the previous API	
Examples:	
<ul style="list-style-type: none"> Search for any service in a WKT area http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=geo:ritmi_01&categories=Service&maxResults=100&lang=it&format=html 	




Bugs:

the html version may not consider all the parameters

3.6 Service search by municipality

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/
it allows to retrieve the set of services that are in a specific municipality. The services can be filtered as belonging to specific categories (e.g. Accomodation, Hotel, Restaurant etc), or having specific words in any textual field.	
Parameters:	
<i>selection</i>	name of the municipality like FIRENZE, EMPOLI, PISA possibly with prefix “COMUNE di “
<i>categories</i>	the list of categories of the services to be retrieved, if omitted all kinds of services are returned. It can contain macro categories or categories, if a macro category is specified all categories in the macro category are used. The complete list of categories and macro categories can be retrieved on servicemap.disit.org .
<i>text</i>	words in this parameter are used to retrieve services that contain all these words in any textual description associated with the service.
<i>maxResults</i>	maximum number of results to be returned (if parameter is missing 100 is assumed), if it is 0 all results are returned.
<i>lang</i>	ISO 2 chars language code (e.g. “it”, “en”, “fr”, “de”) to be used for returned descriptions if available in multiple languages. Currently for languages other than “it” and “en” it returns “en” descriptions. (if parameter is missing “en” is assumed)
<i>geometry</i>	true/false, if true it returns a “hasGeometry” property for each service stating if the service has a complex WKT geometries (linestring, polygon) associated with it (if parameter is missing “false” is assumed)

<i>uid</i>	optional user identifier
<i>format</i>	html or json
Results:	
the results format is the same as the previous API	
Examples:	
<ul style="list-style-type: none"> Search for any Entertainment service in the municipality of FIRENZE http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=COMUNE%20di%20FIRENZE&categories=Entertainment&maxResults=100&lang=it&format=html 	
	
Bugs:	
the html version accepts only a selection with prefix “COMUNE di ”	

3.7 Service search by query id

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/
it allows to retrieve the set of services associated with a query stored using the servicemap user interface.	
Parameters:	
<i>queryId</i>	identifier of the query stored on servicemap
<i>lang</i>	ISO 2 chars language code (e.g. “it”, “en”, “fr”, “de”) to be used for returned descriptions if available in multiple languages. Currently for languages other than “it” and “en” it returns “en” descriptions. (if parameter is missing “en” is assumed)
<i>geometry</i>	true/false, if true it returns a “hasGeometry” property for each service stating if the service has a complex WKT geometries (linestring, polygon) associated with it (if parameter is missing “false” is assumed)
<i>uid</i>	optional user identifier
<i>format</i>	html or json
Results:	
the results format is the same as the previous API	
Examples:	
Search for any BusStop or CulturalActivity service in 100m near Santa Maria del Fiore	

http://servicemap.disit.org/WebAppGrafo/api/v1/?queryId=e02db54355fea40808300473c3537ff&format=json&lang=it
Bugs:

3.8 Full text search

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/
it allows to retrieve the geolocated entities (not only services) that match with a list of keywords. The results can be possibly filtered to be within a specified distance from a GPS position, or within a rectangular area or inside a WKT geolocated area.	
Parameters:	
<i>search</i>	the keywords separated with spaces that have to match with any textual description associated with an entity.
<i>selection</i>	optional "<lat>;<lng>" with a GPS position or "<lat1>;<lng1>;<lat2>;<lng2>" for a rectangular area or "wkt:<WKT_string>" or "geo:<geoid>" for a geographic area described as Well Known Text (see other APIs for more details)
<i>maxDists</i>	optional maximum distance from the GPS position of the entities to be retrieved, expressed in Km
<i>maxResults</i>	maximum number of results to be returned (if parameter is missing 100 is assumed), if it is 0 all results are returned.
<i>lang</i>	ISO 2 chars language code (e.g. "it", "en", "fr", "de") to be used for returned descriptions if available in multiple languages. Currently for languages other than "it" and "en" it returns "en" descriptions. (if parameter is missing "en" is assumed)
<i>geometry</i>	true/false, if true it returns a "hasGeometry" property for each service stating if the service has a complex WKT geometries (linestring, polygon) associated with it (if parameter is missing "false" is assumed)
<i>uid</i>	optional user identifier
<i>format</i>	html or json
Results:	
the results format is a GeoJSON "FeatureCollection" with the matching entities, additionally the "fullCount" property provides the full count of results available matching the query. For each "Feature" a minimal set of properties are provided.	
Examples:	
Search for any geolocated entity matching "via nave" http://servicemap.disit.org/WebAppGrafo/api/v1/?search=via%20nave&maxResults=10&lang=en&format=json <pre>{ "fullCount": 558, "type": "FeatureCollection", "features": [{ "geometry": { "type": "Point", "coordinates": [11.315443, 43.756367] }, "type": "Feature", "properties": { "serviceUri": "http://www.disit.org/km4city/resource/e96076db6e4e2b8b43fb660579eb4de8", "name": "PICCIOLI DANIELE", "tipo": "servizio", "photoThumbs": [], "multimedia": "", "civic": "" } }] }</pre>	

```

        "serviceType": "CulturalActivity_Theatre",
        "typeLabel": "Theatre"
    },
    "id": 1
}, ... {
    "geometry": {
        "type": "Point",
        "coordinates": [10.898357, 43.729973]
    },
    "type": "Feature",
    "properties": {
        "serviceUri": "http://www.disit.org/km4city/resource/RT04801406596TO",
        "name": "VIA NAVE DI VITIANA",
        "tipo": "servizio",
        "photoThumbs": [],
        "multimedia": "",
        "civic": "1",
        "serviceType": "",
        "typeLabel": "Road"
    },
    "id": 8
}, ... ]
}

```

Bugs:

the html version may not consider all the parameters

3.9 Event search

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/events/
it allows to retrieve the geolocated events in a given temporal range (day, week or month). The results can be possibly filtered to be within a specified distance from a GPS position, or within a rectangular area or inside a WKT described geographic area.	
Parameters:	
<i>range</i>	time range for the events to be retrieved, it can be 'day' for the events of the day of the request, 'week' for the events in the next 7 days or 'month' for the events in the next 30 days (if omitted 'day' is assumed).
<i>selection</i>	optional "<lat>;<lng>" with a GPS position or "<lat1>;<lng1>;<lat2>;<lng2>" for a rectangular area or "wkt:<WKT_string>" or "geo:<geoid>" for a geographic area described as Well Known Text (see other APIs for more details).
<i>maxDists</i>	optional maximum distance from the GPS position of the events to be retrieved, expressed in Km.
<i>maxResults</i>	maximum number of results to be returned (if parameter is missing 100 is assumed), if it is 0 all results are returned.
<i>uid</i>	optional user identifier
<i>format</i>	only json
Results:	
the results format is a GeoJSON "FeatureCollection" with the matching events. For each "Feature" a set of properties is provided.	
Examples:	
Search for events of today http://servicemap.disit.org/WebAppGrafo/api/v1/events/?range=day&format=json <pre> { "Event": { "type": "FeatureCollection", "features": [{ "geometry": { "type": "Point", "coordinates": [11.251058, 43.769848] }, "type": "Feature", </pre>	

```

    "properties": {
      "serviceUri": "http://www.disit.org/km4city/resource/Event_18794_973b96efaf3f99f1b70af19cda4e3bf4",
      "name": "Tra arte e moda",
      "tipo": "event",
      "place": "MUSEO SALVATORE FERRAGAMO ",
      "startDate": "2016-05-19",
      "startTime": "10.00 -19.30; chiuso 1/1, 01/05, 15/08 e 25/12",
      "endDate": "2017-04-07",
      "freeEvent": "NO",
      "address": "PIAZZA DI SANTA TRINITA",
      "civic": "2",
      "categoryIT": "Mostre",
      "price": "6 (incluso museo/including museum)",
      "phone": "055 3562466",
      "descriptionIT": "La mostra riflette il complesso rapporto fra arte e moda prendendo spunto dalla storia di
Salvatore Ferragamo che si ispirò alle avanguardie artistiche del '900 per realizzare le sue creazioni. ",
      "website": "www.ferragamomuseo.com/museo",
      "serviceType": "Event"
    },
    "id": 1
  }, ... ]
}

```

Bugs:

problems with duplicated events and with accented chars (solved for new events, still present for old events).

3.10 Address and geometry search by GPS

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/location
it allows to retrieve the complete address (municipality, street and civic number) given the GPS position. It may also provide a list of services or public transport lines intersecting with the provided GPS position.	
Parameters:	
<i>position</i>	"<lat>;<lng>" with a GPS position.
<i>intersectGeom</i>	true or false (assumed false if missing), if true it reports all the services and public transportation lines that have a geometry intersecting with the provided GPS position.
<i>uid</i>	optional user identifier
<i>format</i>	only json
Results:	
A JSON object with properties: <ul style="list-style-type: none"> • <i>address</i>: the street name. • <i>number</i>: the civic number. • <i>addressUri</i>: the URI identifying the civic number in the road graph. • <i>municipality</i>: the estimated municipality (it may not work properly on the municipalities borders) • <i>municipalityUri</i>: the URI identifying the municipality in the road graph. • <i>intersect</i>: array of objects with properties: <ul style="list-style-type: none"> ○ <i>name</i>: name of the intersecting service or public transport line. ○ <i>uri</i>: URI of the intersecting service or public transport line. ○ <i>class</i>: URI representing the class ○ <i>type</i>: type of geometry intersecting the GPS position, can be <i>lineString</i> or <i>Polygon</i> ○ <i>routeType</i>: type of route can be Bus, LightRail, Ferry, Train ○ <i>agency</i>: name of the agency providing the service 	

- *direction*: direction of the line
- *distance*: distance of the GPS position with the intersecting geometry

note: address, number and addressUri may be not present if the GPS position is outside a populated place.

Examples:

<http://servicemap.disit.org/WebAppGrafo/api/v1/location/?position=43.7741;11.2505&format=json>

```
{
  "address": "VIA PANZANI",
  "municipality": "FIRENZE",
  "number": "17/A",
  "addressUri": "http://www.disit.org/km4city/resource/RT048017023351CV",
  "municipalityUri": "http://www.disit.org/km4city/resource/048017"
}
```

<http://servicemap.disit.org/WebAppGrafo/api/v1/location/?position=43.7741;11.2505&intersectGeom=true&format=json>

```
{
  "address": "VIA PANZANI",
  "municipality": "FIRENZE",
  "number": "17/A",
  "addressUri": "http://www.disit.org/km4city/resource/RT048017023351CV",
  "municipalityUri": "http://www.disit.org/km4city/resource/048017",
  "intersect": [
    {
      "distance": 1.2392468323025842E-4,
      "name": "Firenze Card",
      "class": "http://www.disit.org/km4city/schema#Tourist_trail",
      "type": "LineString",
      "uri": "http://www.disit.org/km4city/resource/2a93692aa1eb7d680d9b4e0da668b408"
    },
    {
      "distance": 3.1448272583131523E-4,
      "routeType": "Bus",
      "direction": "Salviatino",
      "name": "11",
      "agency": "Ataf&Linea",
      "class": "http://vocab.gtfs.org/terms#Route",
      "type": "LineString",
      "uri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Trip_1923_3337883"
    },
    ... ]
}
```

Bugs:

3.11 Service info

The Service info API allows getting information about a specific service or entity identified by a serviceURI property returned from the search APIs. Information can be get using the following REST API but also using the Linked Data paradigm using the serviceURI itself.

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/
it allows to retrieve information about a service using its serviceUri. It can return an html representation (format="html") or a machine readable representation (format="json")	
Parameters:	
<i>serviceUri</i>	the serviceUri of the service
<i>lang</i>	ISO 2 chars language code (e.g. "it", "en", "fr", "de") to be used for returned descriptions if available in multiple languages. Currently for languages other than "it" and "en" it returns "en" descriptions. (if parameter is missing "en" is assumed)
<i>realtime</i>	true or false (if omitted true is implied) indicates if the last value of the time varying properties should be provided in the result or not.
<i>uid</i>	optional user identifier
<i>format</i>	html or json

Results:

if format is “html” provides a visual representation of the service on the map. If format is json the API provides a GeoJSON description of the service with the main properties (name, address, city, type, etc.) and possibly some time varying properties for some kinds of services (traffic sensors, car park sensors, etc.).

The following is an example for a SensorSite (traffic sensor)

```
{
  "Sensor": {
    "type": "FeatureCollection",
    "features": [{
      "geometry": {
        "type": "Point",
        "coordinates": [11.2702, 43.77467]
      },
      "type": "Feature",
      "properties": {
        "name": "FI055ZTL02001",
        "typeLabel": "Sensor",
        "serviceType": "TransferServiceAndRenting_SensorSite",
        "serviceUri": "http://www.disit.org/km4city/resource/FI055ZTL02001",
        "municipality": "FIRENZE",
        "address": "VIA DELLA MATTONAIA",
        "photos": [],
        "photoThumbs": [],
        "photoOrigs": [],
        "avgStars": 0.0,
        "starsCount": 0,
        "comments": []
      },
      "id": 1
    }]
  },
  "realtime": {
    "head": {
      "sensor": ["FI055ZTL02001"],
      "vars": ["avgDistance", "avgTime", "occupancy", "concentration", "vehicleFlow", "averageSpeed", "thresholdPerc", "speedPercentile", "instantTime"]
    },
    "results": {
      "bindings": [{
        "avgDistance": {
          "type": "literal",
          "value": "Not Available"
        },
        "avgTime": {
          "type": "literal",
          "value": "Not Available"
        },
        "occupancy": {
          "type": "literal",
          "value": "Not Available"
        },
        "concentration": {
          "type": "literal",
          "value": "0.0"
        },
        "vehicleFlow": {
          "type": "literal",
          "value": "42.0"
        },
        "averageSpeed": {
          "type": "literal",
          "value": "0.0"
        },
        "thresholdPerc": {
          "type": "literal",
          "value": "Not Available"
        },
        "speedPercentile": {
          "type": "literal",
          "value": "Not Available"
        }
      ]
    }
  }
}
```

<pre> "instantTime": { "type": "literal", "value": "2017-01-17T16:32:00+01:00" } } } } } </pre>
Examples:
<ul style="list-style-type: none"> • see the following sections for details on the various kinds of services
Bugs:

3.12 Generic service

For generic services (e.g. Accommodations, Restaurants, etc.) the following properties are provided in the GeoJSON properties:

- *serviceUri*: an URI identifying the service globally
- *name*: name of the service
- *typeLabel*: label associated with the type of service in the language provided with the lang parameter
- *serviceType*: a string containing “<MacroClass>_<ServiceType>”
- *city*, *address*, *civic*: municipality, address and civic number of the service
- *phone*, *fax*, *website*, *email*: phone, fax, website, email of the service
- *note*: notes associated with the service
- *description*, *description2*: two descriptions of the service, one in Italian and the other in English if available.
- *multimedia*: an url to a multimedia resource
- *linkDBpedia*: array of urls to dbpedia resources
- *photo*, *photoThumbs*, *photoOrigins*: array of urls to photos, thumbnails and original photos provided using the photo API.
- *wktGeometry*: a Well Known Text geometry associated with the service
- *avgStars*: average number of stars provided with the stars API
- *starsCount*: number of ratings provided by users.
- *comments*: array of comments on the service provided by users using the comments API

the following is an example:

```

{
  "Service": {
    "type": "FeatureCollection",
    "features": [
      {
        "geometry": {
          "type": "Point",
          "coordinates": [11.361144, 44.00213]
        },
        "type": "Feature",
        "properties": {
          "name": "IL_BRONCO",
          "typeLabel": "Boarding house",
          "serviceType": "Accommodation_Boarding_house",
          "phone": "0558430207",
          "fax": "",
          "website": "www.ristoranteilbronco.it",
          "province": "FI",

```

```

        "city": "SCARPERIA",
        "cap": "50038",
        "email": "info@ristoranteilbronco.it",
        "linkDBpedia": [],
        "note": "",
        "description": "",
        "description2": "",
        "multimedia": "",
        "serviceUri": "http://www.disit.org/km4city/resource/9fc542b468509b922aeb833273dd40d0",
        "address": "VIA DANTE",
        "civic": "95",
        "wktGeometry": "",
        "photos": [],
        "photoThumbs": [],
        "photoOrigins": [],
        "avgStars": 0.0,
        "starsCount": 0,
        "comments": []
    },
    "id": 1
}
]
}
}

```

3.12.1 Event

http://servicemap.disit.org/WebAppGrafo/api/v1/?serviceUri=http://www.disit.org/km4city/resource/Event_18794_973b96efaf3f99f1b70af19cda4e3bf4

```

{
  "Event": {
    "type": "FeatureCollection",
    "features": [{
      "geometry": {
        "type": "Point",
        "coordinates": [11.251058, 43.769848]
      },
      "type": "Feature",
      "properties": {
        "serviceUri": "http://www.disit.org/km4city/resource/Event_18794_973b96efaf3f99f1b70af19cda4e3bf4",
        "name": "Tra arte e moda",
        "name2": "Accross art and fashion",
        "website": "www.ferragamomuseo.com/museo",
        "address": "PIAZZA DI SANTA TRINITA",
        "number": "2",
        "province": "FI",
        "city": "Firenze",
        "note": "",
        "description": "La mostra riflette il complesso rapporto fra arte e moda prendendo spunto dalla storia di Salvatore Ferragamo che si ispirò alle avanguardie artistiche del '900 per realizzare le sue creazioni. ",
        "description2": "The exhibition reflects the complex relationship between art and fashion starting from the the story of Salvatore Ferragamo who realized his creations inspired by the avant-garde art of the '900. ",
        "startDate": "2016-05-19T00:00:00+02:00",
        "startTime": "10.00 -19.30; chiuso 1/1, 01/05, 15/08 e 25/12",
        "endDate": "2017-04-07T00:00:00+02:00",
        "eventCategory": "Mostre",
        "eventCategory2": "Exhibitions",
        "photos": [],
        "photoThumbs": [],
        "photoOrigins": [],
        "avgStars": 0.0,
        "starsCount": 0,
        "comments": []
      },
      "id": 1
    }]
  }
}

```

3.12.2 Parking service

<http://servicemap.disit.org/WebAppGrafo/api/v1/?serviceUri=http://www.disit.org/km4city/resource/RT04801702315PO>


```
{
  "Service": {
    "type": "FeatureCollection",
    "features": [
      {
        "geometry": {
          "type": "Point",
          "coordinates": [11.24947, 43.77587]
        },
        "type": "Feature",
        "properties": {
          "name": "Garage La Stazione Spa",
          "typeLabel": "Car park",
          "serviceType": "TransferServiceAndRenting_Car_park",
          "phone": "055284784",
          "fax": "",
          "website": "",
          "province": "FI",
          "city": "FIRENZE",
          "cap": "50123",
          "email": "",
          "linkDBpedia": [],
          "note": "",
          "description": "",
          "description2": "",
          "multimedia": "",
          "serviceUri": "http://www.disit.org/km4city/resource/RT04801702315PO",
          "address": "PIAZZA DELLA STAZIONE",
          "civic": "3A",
          "wktGeometry": "",
          "photos": [],
          "photoThumbs": [],
          "photoOrigins": [],
          "avgStars": 0.0,
          "starsCount": 0,
          "comments": []
        },
        "id": 1
      }
    ]
  },
  "realtime": {
    "head": {
      "parkingArea": ["Garage La Stazione Spa"],
      "vars": ["capacity", "freeParkingLots", "occupiedParkingLots", "occupancy", "updating"]
    },
    "results": {
      "bindings": [
        {
          "capacity": {
            "value": "617"
          },
          "freeParkingLots": {
            "value": "322"
          },
          "occupiedParkingLots": {
            "value": "579"
          },
          "occupancy": {
            "value": "0.0"
          },
          "status": {
            "value": "enoughSpacesAvailable"
          },
          "updating": {
            "value": "2017-01-18T14:25:00+01:00"
          }
        }
      ]
    }
  }
}
```

3.12.3 Traffic sensor

<http://servicemap.disit.org/WebAppGrafo/api/v1/?serviceUri=http://www.disit.org/km4city/resource/METRO487>

```
{
  "Sensor": {
    "type": "FeatureCollection",
    "features": [{
      "geometry": {
        "type": "Point",
        "coordinates": [11.25003, 43.7747]
      },
      "type": "Feature",
      "properties": {
        "name": "METRO487",
        "typeLabel": "Sensor",
        "serviceType": "TransferServiceAndRenting_SensorSite",
        "serviceUri": "http://www.disit.org/km4city/resource/METRO487",
        "municipality": "FIRENZE",
        "address": "ZTL02 - Preferenziale P.zza Unità-Panzani",
        "photos": [],
        "photoThumbs": [],
        "photoOrigins": [],
        "avgStars": 0.0,
        "starsCount": 0,
        "comments": []
      },
      "id": 1
    }]
  },
  "realtime": {
    "head": {
      "sensor": ["METRO487"],
      "vars": ["avgDistance", "avgTime", "occupancy", "concentration", "vehicleFlow", "averageSpeed", "thresholdPerc", "speedPercentile",
"instantTime"]
    },
    "results": {
      "bindings": [{
        "avgDistance": {
          "type": "literal",
          "value": "Not Available"
        },
        "avgTime": {
          "type": "literal",
          "value": "2.49806"
        },
        "occupancy": {
          "type": "literal",
          "value": "Not Available"
        },
        "concentration": {
          "type": "literal",
          "value": "3.522905"
        },
        "vehicleFlow": {
          "type": "literal",
          "value": "330.0"
        },
        "averageSpeed": {
          "type": "literal",
          "value": "93.6727"
        },
        "thresholdPerc": {
          "type": "literal",
          "value": "Not Available"
        },
        "speedPercentile": {
          "type": "literal",
          "value": "Not Available"
        },
        "instantTime": {
          "type": "literal",
          "value": "2017-01-18T09:41:00+01:00"
        }
      }]
    }
  }
}
```

```
    }
  }
}
```

3.12.4 Weather Forecast

<http://servicemap.disit.org/WebAppGrafo/api/v1/?serviceUri=http://www.disit.org/km4city/resource/048017>

```
{
  "head": {
    "location": "FIRENZE",
    "vars": ["day", "description", "minTemp", "maxTemp", "instantDateTime"]
  },
  "results": {
    "bindings": [{
      "day": {
        "type": "literal",
        "value": "Mercoledì"
      },
      "description": {
        "type": "literal",
        "value": "nuvoloso"
      },
      "minTemp": {
        "type": "literal",
        "value": "4"
      },
      "maxTemp": {
        "type": "literal",
        "value": "6"
      },
      "instantDateTime": {
        "type": "literal",
        "value": "2017-01-18T09:39:00+01:00"
      }
    }, {
      "day": {
        "type": "literal",
        "value": "Giovedì"
      },
      "description": {
        "type": "literal",
        "value": "coperto"
      },
      "minTemp": {
        "type": "literal",
        "value": "3"
      },
      "maxTemp": {
        "type": "literal",
        "value": "7"
      },
      "instantDateTime": {
        "type": "literal",
        "value": "2017-01-18T09:39:00+01:00"
      }
    }, {
      "day": {
        "type": "literal",
        "value": "Venerdì"
      },
      "description": {
        "type": "literal",
        "value": "poco nuvoloso"
      },
      "minTemp": {
        "type": "literal",
        "value": "1"
      },
      "maxTemp": {
        "type": "literal",
        "value": "7"
      },
      "instantDateTime": {

```

```

        "type": "literal",
        "value": "2017-01-18T09:39:00+01:00"
    }
}, {
    "day": {
        "type": "literal",
        "value": "Sabato"
    },
    "description": {
        "type": "literal",
        "value": "poco nuvoloso"
    },
    "minTemp": {
        "type": "literal",
        "value": ""
    },
    "maxTemp": {
        "type": "literal",
        "value": ""
    },
    "instantDateTime": {
        "type": "literal",
        "value": "2017-01-18T09:39:00+01:00"
    }
}, {
    "day": {
        "type": "literal",
        "value": "Domenica"
    },
    "description": {
        "type": "literal",
        "value": "nuvoloso"
    },
    "minTemp": {
        "type": "literal",
        "value": ""
    },
    "maxTemp": {
        "type": "literal",
        "value": ""
    },
    "instantDateTime": {
        "type": "literal",
        "value": "2017-01-18T09:39:00+01:00"
    }
}
}
}
}

```

3.12.5 Bus stop

http://servicemap.disit.org/WebAppGrafo/api/v1/?serviceUri=http://www.disit.org/km4city/resource/Bus_ataflinea_Stop_FM0022_5

```

{
  "BusStop": {
    "type": "FeatureCollection",
    "features": [{
      "geometry": {
        "type": "Point",
        "coordinates": [11.249069, 43.776485]
      },
      "type": "Feature",
      "properties": {
        "name": "Stazione Pensilina",
        "serviceUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Stop_FM0022_5",
        "typeLabel": "BusStop",
        "address": "",
        "agency": "Ataf&Linea",
        "agencyUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172",
        "serviceType": "TransferServiceAndRenting_BusStop",
        "photos": [],
        "photoThumbs": [],
        "photoOrigins": [],
        "avgStars": 0.0,

```

```

        "starsCount": 0,
        "comments": []
    },
    "id": 1
  ]
},
"busLines": {
  "head": {
    "busStop": "Stazione Pensilina",
    "vars": ["busLine", "lineUri", "lineDesc"]
  },
  "results": {
    "bindings": [{
      "busLine": {
        "type": "literal",
        "value": "1"
      },
      "lineUri": {
        "type": "literal",
        "value": "http://www.disit.org/km4city/resource/Bus_ataflinea_Route_122797549"
      },
      "lineDesc": {
        "type": "literal",
        "value": "Lapo\\Boccaccio - S.Maria Novella Fs"
      }
    }, {
      "busLine": {
        "type": "literal",
        "value": "11"
      },
      "lineUri": {
        "type": "literal",
        "value": "http://www.disit.org/km4city/resource/Bus_ataflinea_Route_1073492795"
      },
      "lineDesc": {
        "type": "literal",
        "value": "Salviatino-Le Gore"
      }
    }, {
      "busLine": {
        "type": "literal",
        "value": "17"
      },
      "lineUri": {
        "type": "literal",
        "value": "http://www.disit.org/km4city/resource/Bus_ataflinea_Route_1208385503"
      },
      "lineDesc": {
        "type": "literal",
        "value": "Viale Verga-Via Boito\\Cascine"
      }
    }
  ], ...
}
},
"timetable": {
  "head": {
    "vars": ["date", "arrivalTime", "lineName", "lineDesc", "routeName", "trip"]
  },
  "results": {
    "bindings": [{
      "date": {
        "type": "literal", "value": "2017-01-18"
      },
      "arrivalTime": {
        "type": "literal", "value": "14:52:00"
      },
      "departureTime": {
        "type": "literal", "value": "14:52:00"
      },
      "lineName": {
        "type": "literal", "value": "6"
      },
      "lineDesc": {
        "type": "literal", "value": "Novelli-Smn-Torregalli"
      }
    }
  ]
}

```

```

        "routeName": {
          "type": "literal", "value": "Ospedale Torre Galli"
        },
        "trip": {
          "type": "uri", "value": "http://www.disit.org/km4city/resource/Bus_ataflinea_Trip_1923_3364525"
        }
      }, {
        "date": {
          "type": "literal", "value": "2017-01-18"
        },
        "arrivalTime": {
          "type": "literal", "value": "14:56:00"
        },
        "departureTime": {
          "type": "literal", "value": "14:56:00"
        },
        "lineName": {
          "type": "literal", "value": "11"
        },
        "lineDesc": {
          "type": "literal", "value": "Salviatino-Le Gore"
        },
        "routeName": {
          "type": "literal", "value": "La Gora"
        },
        "trip": {
          "type": "uri", "value": "http://www.disit.org/km4city/resource/Bus_ataflinea_Trip_1923_3344062"
        }
      }, ...
    ], ...
  },
  "realtime": {
  }
}

```

3.12.6 Fuel Station

http://www.disit.org/ServiceMap/api/v1/?serviceUri=http://www.disit.org/km4city/resource/Fuel_station_01a234db6235dd55448a5044d9d26a52

```

{
  "Service": {
    "type": "FeatureCollection",
    "features": [
      {
        "geometry": {
          "type": "Point",
          "coordinates": [11.279211, 43.78041]
        },
        "type": "Feature",
        "properties": {
          "serviceUri": "http://www.disit.org/km4city/resource/Fuel_station_01a234db6235dd55448a5044d9d26a52",
          "serviceType": "TransferServiceAndRenting_Fuel_station",
          "name": "PINI E SETTESOLDI SNC",
          "typeLabel": "Fuel station",
          "phone": "",
          "fax": "",
          "website": "",
          "province": "FI",
          "city": "FIRENZE",
          "cap": "50131",
          "email": "",
          "note": "",
          "description": "",
          "description2": "",
          "multimedia": "",
          "address": "VIALE DEI MILLE",
          "civic": "",
          "brand": "AgipEni",
          "linkDBpedia": [],
          "wktGeometry": "",
          "photos": [],
          "photoThumbs": [],

```

```

        "photoOrigins": [],
        "avgStars": 0.0,
        "starsCount": 0,
        "comments": []
    }
}
],
},
"realtime": {
    "head": {
        "vars": ["measuredTime", "fuel", "price", "currency", "self"]
    },
    "results": {
        "bindings": [{
            "measuredTime": {
                "value": "2017-01-13 16:01:52"
            },
            "fuel": {
                "value": "Benzina"
            },
            "price": {
                "value": "1.579"
            },
            "currency": {
                "value": "EUR"
            },
            "self": {
                "value": "true"
            }
        }, {
            "measuredTime": {
                "value": "2017-01-13 16:01:52"
            },
            "fuel": {
                "value": "Blue Diesel"
            },
            "price": {
                "value": "1.539"
            },
            "currency": {
                "value": "EUR"
            },
            "self": {
                "value": "true"
            }
        }, {
            "measuredTime": {
                "value": "2017-01-13 16:01:52"
            },
            "fuel": {
                "value": "Blue Super"
            },
            "price": {
                "value": "1.729"
            },
            "currency": {
                "value": "EUR"
            },
            "self": {
                "value": "true"
            }
        }, {
            "measuredTime": {
                "value": "2017-01-13 16:01:52"
            },
            "fuel": {
                "value": "Gasolio"
            },
            "price": {
                "value": "1.439"
            },
            "currency": {
                "value": "EUR"
            },
            "self": {
                "value": "true"
            }
        }
    ]
}

```

```

    }
  }
}

```

3.12.7 First aid (added with RESOLUTE project)

<http://www.disit.org/ServiceMap/api/v1/?serviceUri=http://www.disit.org/km4city/resource/dde440c760ef578da41599feb2396631>

```

{
  "Service": {
    "type": "FeatureCollection",
    "features": [
      {
        "geometry": {
          "type": "Point",
          "coordinates": [11.260015, 43.773457]
        },
        "type": "Feature",
        "properties": {
          "name": "PRONTO SOCCORSO OSPEDALE SANTA MARIA NUOVA",
          "typeLabel": "First aid",
          "serviceType": "Emergency_First_aid",
          "phone": "0552758844",
          "fax": "0552758844",
          "website": "",
          "province": "FI",
          "city": "FIRENZE",
          "cap": "50100",
          "email": "",
          "linkDBpedia": [],
          "note": "",
          "description": "",
          "description2": "",
          "multimedia": "",
          "serviceUri": "http://www.disit.org/km4city/resource/dde440c760ef578da41599feb2396631",
          "address": "PIAZZA SANTA MARIA NUOVA",
          "civic": "1",
          "wktGeometry": "",
          "photos": [],
          "photoThumbs": [],
          "photoOrigins": [],
          "avgStars": 0.0,
          "starsCount": 0,
          "comments": []
        },
        "id": 1
      }
    ]
  },
  "realtime": {
    "head": {
      "vars": ["measuredTime", "state", "redCode", "yellowCode", "greenCode", "blueCode", "whiteCode"]
    },
    "results": {
      "bindings": [
        {
          "measuredTime": {
            "value": "2017/01/19T15:25:00.000"
          },
          "state": {
            "value": "Con Destinazione"
          },
          "redCode": {
            "value": "0"
          },
          "yellowCode": {
            "value": "5"
          },
          "greenCode": {
            "value": "5"
          },
          "blueCode": {

```



```

        "value": "1"
    },
    "whiteCode": {
        "value": "0"
    }
}, {
    "measuredTime": {
        "value": "2017/01/19T15:25:00.000"
    },
    "state": {
        "value": "In Attesa"
    },
    "redCode": {
        "value": "0"
    },
    "yellowCode": {
        "value": "2"
    },
    "greenCode": {
        "value": "5"
    },
    "blueCode": {
        "value": "1"
    },
    "whiteCode": {
        "value": "0"
    }
}, {
    "measuredTime": {
        "value": "2017/01/19T15:25:00.000"
    },
    "state": {
        "value": "In Visita"
    },
    "redCode": {
        "value": "0"
    },
    "yellowCode": {
        "value": "4"
    },
    "greenCode": {
        "value": "5"
    },
    "blueCode": {
        "value": "1"
    },
    "whiteCode": {
        "value": "0"
    }
}, {
    "measuredTime": {
        "value": "2017/01/19T15:25:00.000"
    },
    "state": {
        "value": "Oss. Temporanea"
    },
    "redCode": {
        "value": "0"
    },
    "yellowCode": {
        "value": "1"
    },
    "greenCode": {
        "value": "2"
    },
    "blueCode": {
        "value": "1"
    },
    "whiteCode": {
        "value": "0"
    }
}, {
    "measuredTime": {
        "value": "2017/01/19T15:25:00.000"
    },
    "state": {

```

```

        "value": "Totali"
      },
      "redCode": {
        "value": "0"
      },
      "yellowCode": {
        "value": "12"
      },
      "greenCode": {
        "value": "17"
      },
      "blueCode": {
        "value": "4"
      },
      "whiteCode": {
        "value": "0"
      }
    }
  }
}

```

3.12.8 Smart waste container (added with REPLICATE project)

<http://www.disit.org/ServiceMap/api/v1/?serviceUri=http://www.disit.org/km4city/resource/cassonetto01>

```

{
  "Service": {
    "type": "FeatureCollection",
    "features": [
      {
        "geometry": {
          "type": "Point",
          "coordinates": [11.2557, 43.7745]
        },
        "type": "Feature",
        "properties": {
          "serviceUri": "http://www.disit.org/km4city/resource/cassonetto01",
          "serviceType": "Environment_Smart_waste_container",
          "name": "Cassonetto via martelli",
          "typeLabel": "Smart waste container",
          "phone": "055232323",
          "province": "FI",
          "city": "Firenze",
          "cap": "",
          "address": "via martelli",
          "civic": "2",
          "wasteType": "http://www.disit.org/km4city/schema#anyWaste",
          "capacity": "200",
          "collectionTime": "alle 13:00 tutti I giorni",
          "physicalShape": "campana",
          "linkDBpedia": [],
          "wktGeometry": "",
          "photos": [],
          "photoThumbs": [],
          "photoOrigins": [],
          "avgStars": 0.0,
          "starsCount": 0,
          "comments": []
        }
      }
    ]
  },
  "realtime": {
    "head": {
      "vars": ["measuredTime", "wasteLevel", "batteryLevel"]
    },
    "results": {
      "bindings": [{
        "measuredTime": {
          "value": "2017-01-19T15:46:31+01:00"
        },
        "wasteLevel": {

```

```

        "value": "0.53592324"
      },
      "batteryLevel": {
        "value": "261.33566"
      }
    }
  }
}

```

3.12.9 Smart bench (added with REPLICATE project)

<http://www.disit.org/ServiceMap/api/v1/?serviceUri=http://www.disit.org/km4city/resource/bench001>

```

{
  "Service": {
    "type": "FeatureCollection",
    "features": [
      {
        "geometry": {
          "type": "Point",
          "coordinates": [11.2554, 43.7737]
        },
        "type": "Feature",
        "properties": {
          "serviceUri": "http://www.disit.org/km4city/resource/bench001",
          "serviceType": "Entertainment_Smart_bench",
          "name": "Panchina via martelli",
          "typeLabel": "Smart bench",
          "phone": "055232323",
          "fax": "",
          "website": "",
          "province": "FI",
          "city": "Firenze",
          "cap": "",
          "email": "",
          "note": "",
          "description": "",
          "description2": "",
          "multimedia": "",
          "address": "via martelli",
          "civic": "2",
          "seats": "4",
          "withWifi": "true",
          "withUsb": "true",
          "withAudio": "true",
          "linkDBpedia": [],
          "wktGeometry": "",
          "photos": [],
          "photoThumbs": [],
          "photoOrigins": [],
          "avgStars": 0.0,
          "starsCount": 0,
          "comments": []
        }
      }
    ]
  },
  "realtime": {
    "head": {
      "vars": ["measuredTime", "temperature", "humidity", "pressure", "airQualityCO2", "light", "sittingsInRefPeriod", "totalSittings", "passagesInRefPeriod", "totalPassages"]
    },
    "results": {
      "bindings": [{
        "measuredTime": {
          "value": "2017-01-19T15:42:52+01:00"
        },
        "temperature": {
          "value": "36.675144"
        },
        "humidity": {
          "value": "83.60987"
        }
      }
    ]
  }
}

```

```

    },
    "pressure": {
      "value": "24.017311"
    },
    "airQualityCO2": {
      "value": "85.21111"
    },
    "light": {
      "value": "0.51458716"
    },
    "sittingsInRefPeriod": {
      "value": "0"
    },
    "totalSittings": {
      "value": "656"
    },
    "passagesInRefPeriod": {
      "value": "3"
    },
    "totalPassages": {
      "value": "3313"
    }
  }
}
}
}
}

```

3.12.10 Smart irrigator (added with REPLICATE project)

<http://www.disit.org/ServiceMap/api/v1/?serviceUri=http://www.disit.org/km4city/resource/irrigatore01>

```

{
  "Service": {
    "type": "FeatureCollection",
    "features": [
      {
        "geometry": {
          "type": "Point",
          "coordinates": [11.2496, 43.7736]
        },
        "type": "Feature",
        "properties": {
          "serviceUri": "http://www.disit.org/km4city/resource/irrigatore01",
          "serviceType": "Environment_Smart_irrigator",
          "name": "Irrigatore p.zza S. Maria Novella",
          "typeLabel": "Smart irrigator",
          "phone": "0552556677",
          "province": "FI",
          "city": "Firenze",
          "note": "",
          "description": "",
          "description2": "",
          "address": "p.zza Santa Maria Novella",
          "civic": "23",
          "linkDBpedia": [],
          "wktGeometry": "",
          "photos": [],
          "photoThumbs": [],
          "photoOrigs": [],
          "avgStars": 0.0,
          "starsCount": 0,
          "comments": []
        }
      }
    ]
  },
  "realtime": {
    "head": {
      "vars": ["measuredTime", "currentlyActive", "temperature", "internalTemperature", "humidity", "soilWaterPotential", "leafWetness"]
    },
    "results": {
      "bindings": [{
        "measuredTime": {

```

```

        "value": "2017-01-19T15:46:31+01:00"
      },
      "currentlyActive": {
        "value": "true"
      },
      "temperature": {
        "value": "14.397217"
      },
      "internalTemperature": {
        "value": "26.770363"
      },
      "humidity": {
        "value": "26.808607"
      },
      "soilWaterPotential": {
        "value": "329.1715"
      },
      "leafWetness": {
        "value": "23.110199"
      }
    }
  }
}

```

3.12.11 Energy meter (added with REPLICATE project)

Under development

3.12.12 Recharge station (added with REPLICATE project)

Under development

3.12.13 Smart street light (added with REPLICATE project)

Under development

3.12.14 Air quality monitoring station

Under development

3.13 Public transport API

In the following the API that are related with public transports are reported.

Note: The information regarding timetable is acquired in GTFS format. Due to different names used in the previous version of the API that was only for buses, the names used in the API are not aligned with GTFS nomenclature in particular bus lines are mapped to GTFS routes and bus routes are mapped to GTFS trips. In the next version of the API names used may change to be aligned with GTFS.

3.13.1 Agency list

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/agencies
the API provide a list of the public transport agencies available	
Parameters:	
<i>uid</i>	optional user identifier
<i>format</i>	only json

Results:
the API provides an array of JSON objects of the agencies available, for each agency is provided the agency name and the agency URI used to identify the agency in other APIs
Examples:
<pre>{ "Agencies": [{ "agency": "http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172", "name": "Ataf&Linea" }, { "agency": "http://www.disit.org/km4city/resource/Bus_acvbus_Agency_173", "name": "Autolinee Chianti Valdarno" }, { "agency": "http://www.disit.org/km4city/resource/Bus_amvbus_Agency_171", "name": "Autolinee Mugello Valdisieve" }, { "agency": "http://www.disit.org/km4city/resource/Bus_blubus_Agency_175", "name": "BluBus" }, { "agency": "http://www.disit.org/km4city/resource/Bus_cap_Agency_169", "name": "C.A.P. Consorzio Autolinee Pratesi" }, { "agency": "http://www.disit.org/km4city/resource/Bus_ett_Agency_500", "name": "CTT NORD" }, { "agency": "http://www.disit.org/km4city/resource/Bus_cpt_Agency_176", "name": "Consorzio Pisano Trasporti" }, { "agency": "http://www.disit.org/km4city/resource/Bus_etruriamobilita_Agency_168", "name": "Etruria Mobilità" }, { "agency": "http://www.disit.org/km4city/resource/Tram_gest_Agency_303", "name": "GEST S.p.A." }, { "agency": "http://www.disit.org/km4city/resource/Bus_piubus_Agency_170", "name": "Piùbus" }, { "agency": "http://www.disit.org/km4city/resource/Bus_sienamobilita_Agency_167", "name": "Siena Mobilità" }, { "agency": "http://www.disit.org/km4city/resource/Train_tft_Agency_196", "name": "T.F.T. S.p.A." }, { "agency": "http://www.disit.org/km4city/resource/Bus_tiemme_Agency_400", "name": "TIEMME SPA" }, { "agency": "http://www.disit.org/km4city/resource/Train_trenitalia_Agency_163", "name": "TRENITALIA S.p.A." }, { "agency": "http://www.disit.org/km4city/resource/Ferry_toremarmar_Agency_205", "name": "Toremarmar Toscana Regionale Marittima Spa" }, { "agency": "http://www.disit.org/km4city/resource/Bus_vaibus_Agency_174", "name": "Vaibus" }]}</pre>
Bugs:

3.13.2 (Bus) Lines list

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-lines
the API provide a list of the public transport lines available for a given agency.	
Parameters:	
<i>agency</i>	URI of the agency whose lines are to be retrieved
<i>uid</i>	optional user identifier
<i>format</i>	only json
Results:	
the API provides an array of JSON objects of the lines available, for each line is provided the line	

long and short name, the uri identifying the line.
Examples:
http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-lines/?agency=http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172
<pre>{ "BusLines": [{ "agency": "http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172", "shortName": "C1", "longName": "Parterre-Ponte Alle Grazie", "uri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Route_1380827827" }, { "agency": "http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172", "shortName": "S3", "longName": "Scuola Marconi-L'Olmo", "uri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Route_1858266107" }, ...] }</pre>
Note:
The API can be used on any kind of public transport (Tram, Train, etc.) not only Bus.

3.13.3 (Bus) Routes list

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-routes
API provide a list of the public transport routes available for a given agency, line or passing by a specific stop.	
Parameters:	
<i>agency</i>	URI of the agency whose lines are to be retrieved
<i>line</i>	URI or shortName of a line (if URI is provided the agency is not needed)
<i>busStopName</i>	URI or name of a stop (if URI is provided the agency is not needed)
<i>geometry</i>	if true the WKT geometry of the route is returned (false is assumed if not provided)
<i>uid</i>	optional user identifier
<i>format</i>	only json
Results:	
the API provides an array of JSON objects of the routes available, for each route is provided:	
<ul style="list-style-type: none"> • <i>line</i>: line shot name • <i>route</i>: the route URI • <i>routeName</i>: optional route name • <i>wktGeometry</i>: the WKT geometry of the route • <i>firstBusStop</i>: name of the first bus stop • <i>lastBusStop</i>: name of the last bus stop 	
Examples:	
http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-routes/?agency=http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172&line=11&geometry=true	
<pre>{ "BusRoutes": [{ "line": "11", "route": "http://www.disit.org/km4city/resource/Bus_ataflinea_Trip_1923_3337883", "routeName": "", "wktGeometry": "LINESTRING(11.2172537345524 43.7326316393217, 11.2173853491045 43.7325390476232, ...)", }]</pre>	

```

    "firstBusStop": "La Gora",
    "lastBusStop": "Salviatino"
  }, {
    "line": "11",
    "route": "http://www.disit.org/km4city/resource/Bus_ataflinea_Trip_1923_3338595",
    "routeName": "",
    "wktGeometry": "LINESTRING(11.2939833018846 43.7848045962375, 11.2939931338599 43.7848236867993,...)",
    "firstBusStop": "Salviatino",
    "lastBusStop": "La Gora"
  }
]
}

```

http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-routes/?agency=http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172&busStopName=Stazione%20Pensilina

```

{
  "BusRoutes": [{
    "line": "1",
    "route": "http://www.disit.org/km4city/resource/Bus_ataflinea_Trip_1923_3337874",
    "routeName": "",
    "firstBusStop": "Boccaccio",
    "lastBusStop": "Stazione Palazzo Congressi"
  }, {
    "line": "2",
    "route": "http://www.disit.org/km4city/resource/Bus_ataflinea_Trip_1923_3322861",
    "routeName": "",
    "firstBusStop": "Calenzano",
    "lastBusStop": "Stazione Palazzo Congressi"
  }, {
    "line": "4",
    "route": "http://www.disit.org/km4city/resource/Bus_ataflinea_Trip_1923_3323029",
    "routeName": "",
    "firstBusStop": "Cappuccini",
    "lastBusStop": "Stazione Mercato Centrale"
  }, ... ]
}

```

Note:

The API can be used on any kind of public transport (Tram, Train, etc.) not only Bus.

3.13.4 (Bus) Stop list

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-stops
API provide a list of the public transport stops available for a given route.	
Parameters:	
<i>route</i>	URI of the route whose bus stops are to be retrieved
<i>geometry</i>	if true the WKT geometry of the route is returned
<i>uid</i>	optional user identifier
<i>format</i>	only json
Results:	
the API provides an JSON Object with line number (aka line short name) and line name (aka line long name) and a GeoJSON FeatureCollection with the stops. The stops are provided in stop order, from the first to the last.	
Examples:	
http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-stops/?route=http://www.disit.org/km4city/resource/Bus_ataflinea_Trip_1923_3337883&geometry=true	
<pre> { "Route": { "lineNumber": "11", "lineName": "Salviatino-Le Gore", "wktGeometry": "LINESTRING(11.2172537345524 43.7326316393217, 11.2173853491045 43.7325390476232, ...)" }, "BusStops": { </pre>	


```

"type": "FeatureCollection",
"features": [{
  "geometry": {
    "type": "Point",
    "coordinates": [11.217254, 43.73263]
  },
  "type": "Feature",
  "properties": {
    "popupContent": "La Gora",
    "name": "La Gora",
    "serviceUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Stop_FM1208_5",
    "tipo": "fermata",
    "agency": "Ataf&Linea",
    "agencyUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172",
    "serviceType": "TransferServiceAndRenting_BusStop"
  },
  "id": 1
}, {
  "geometry": {
    "type": "Point",
    "coordinates": [11.220704, 43.73418]
  },
  "type": "Feature",
  "properties": {
    "popupContent": "Volterrana 02",
    "name": "Volterrana 02",
    "serviceUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Stop_FM1209_5",
    "tipo": "fermata",
    "agency": "Ataf&Linea",
    "agencyUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172",
    "serviceType": "TransferServiceAndRenting_BusStop"
  },
  "id": 2
}, ... ]
}

```

Note:

The API can be used on any kind of public transport (Tram, Train, etc.) not only Bus.

3.13.5 Search (Bus) Routes in a geographic area

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/
API provides a list of the public transport routes that have a stop in a specified area.	
Parameters:	
<i>selection</i>	optional "<lat>;<lng>" with a GPS position or "<lat1>;<lng1>;<lat2>;<lng2>" for a rectangular area or "wkt:<WKT_string>" or "geo:<geoid>" for a geographic area described as Well Known Text (see other APIs for more details)
<i>maxDists</i>	optional maximum distance from the GPS position of the entities to be retrieved, expressed in Km (0.1 is assumed if not present)
<i>maxResults</i>	maximum number of results to be returned (if parameter is missing 100 is assumed), if it is 0 all results are returned.
<i>agency</i>	optional URI of an agency to restrict the search to a specified agency
<i>geometry</i>	if true the WKT geometry of each route is returned (considered false if not provided)
<i>uid</i>	optional user identifier
<i>format</i>	only json
Results:	
the API provides a JSON Object with all the routes that have stops on the specified area. For each route the following properties are provided:	
<ul style="list-style-type: none"> lineNumber: the line short name lineName: the line long name route: the route name 	

- routeUri: an URI identifying the route (it can be used to retrieve all the stops of the route)
- direction: with first and last stop
- agency: with agency name
- agencyUri: with agency URI
- polyline: with the WKT geometry of the route

Examples:

<http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/?selection=43.7755;11.2495&maxDists=0.1&maxResults=5&geometry=true>

```
{
  "PublicTransportLine": {
    "type": "FeatureCollection",
    "features": [{
      "type": "Feature",
      "properties": {
        "lineNumber": "12",
        "lineName": "",
        "route": "",
        "routeUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Trip_1923_3137547",
        "direction": "Campo Marte Fs → Stazione Parcheggio",
        "agency": "Ataf&Linea",
        "agencyUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172",
        "polyline": "LINESTRING(11.2762059770919 43.7774442270155, 11.2761623454295 43.777427353435, ...)",
        "serviceType": "PublicTransportLine"
      },
      "id": 1
    }, {
      "type": "Feature",
      "properties": {
        "lineNumber": "36",
        "lineName": "",
        "route": "",
        "routeUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Trip_1923_3317289",
        "direction": "Cascine Del Riccio → Stazione Abside S.M.N.",
        "agency": "Ataf&Linea",
        "agencyUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172",
        "polyline": "LINESTRING(11.2551477298522 43.7339067055819, 11.2550069037315 43.7335043206344, ...)",
        "serviceType": "PublicTransportLine"
      },
      "id": 2
    }, {
      "type": "Feature",
      "properties": {
        "lineNumber": "13",
        "lineName": "",
        "route": "",
        "routeUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Trip_1923_3300218",
        "direction": "Il David → Stazione Palazzo Congressi",
        "agency": "Ataf&Linea",
        "agencyUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172",
        "polyline": "LINESTRING(11.2648824363224 43.7625434190618, 11.2648878248007 43.7625306663665, ...)",
        "serviceType": "PublicTransportLine"
      },
      "id": 3
    }, {
      "type": "Feature",
      "properties": {
        "lineNumber": "11",
        "lineName": "",
        "route": "",
        "routeUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Trip_1923_3337883",
        "direction": "La Gora → Salviatino",
        "agency": "Ataf&Linea",
        "agencyUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172",
        "polyline": "LINESTRING(11.2172537345524 43.7326316393217, 11.2173853491045 43.7325390476232, ...)",
        "serviceType": "PublicTransportLine"
      },
      "id": 4
    }, {
      "type": "Feature",
      "properties": {
```

```

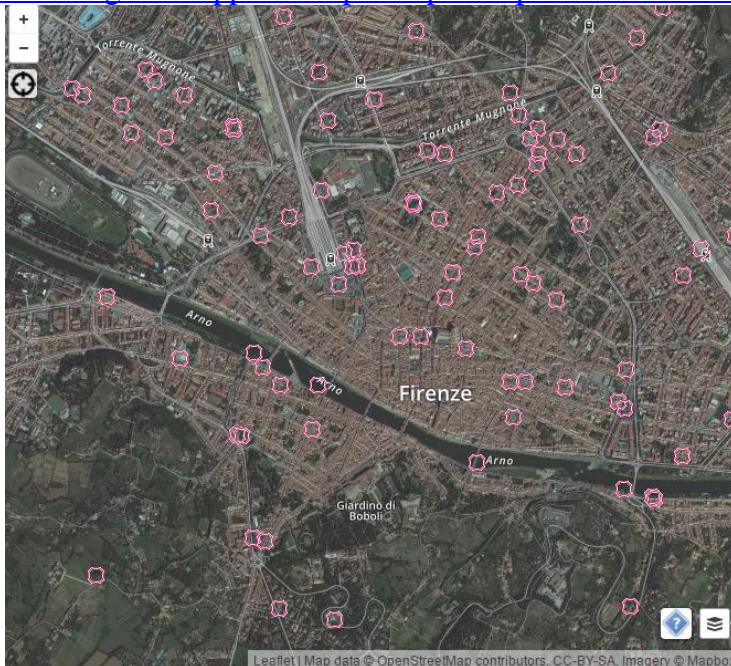
        "lineNumber": "C2",
        "lineName": "",
        "route": "",
        "routeUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Trip_1923_3365643",
        "direction": "Leopolda → Piazza Beccaria",
        "agency": "Ataf&Linea",
        "agencyUri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172",
        "polyline": "LINESTRING(11.2389601794313 43.7773069544217, 11.2389099511421 43.777364365556, ...)",
        "serviceType": "PublicTransportLine"
    },
    "id": 5
  }
}

```

Note:

The API can be used on any kind of public transport (Tram, Train, etc.) not only Bus.

3.13.6 Estimated Bus position

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-position
API provides the estimated current position of buses	
Parameters:	
<i>uid</i>	optional user identifier
<i>format</i>	json or html
Results:	
<p>when format is html the API provides web visualization of the current bus positions while if format is json it provides a GeoJSON “FeatureCollection” with the data of each bus that is currently active. For each bus the following properties are provided:</p> <ul style="list-style-type: none"> • vehicleNum: the number of vehicle • line: the line short name • direction: with first and last stop • detectionTime: the delay in minutes from the current time and the time the position was acquired. 	
Examples:	
http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-position/?format=html	
	
http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-position/?format=json	
{	

```

"type": "FeatureCollection",
"features": [{
  "geometry": {
    "type": "Point",
    "coordinates": [11.340633, 43.735943]
  },
  "type": "Feature",
  "properties": {
    "vehicleNum": "3133579",
    "line": "24",
    "direction": "Sorgane Piazza Rodolico &#10132; Grassina",
    "tipo": "RealTimeInfo",
    "serviceUri": "busCode3133579",
    "detectionTime": "0",
    "serviceType": "bus_real_time"
  },
  "id": 1
}, {
  "geometry": {
    "type": "Point",
    "coordinates": [11.272773, 43.774574]
  },
  "type": "Feature",
  "properties": {
    "vehicleNum": "3134531",
    "line": "12",
    "direction": "Piazzale Michelangelo &#10132; Stazione Parcheggio",
    "tipo": "RealTimeInfo",
    "serviceUri": "busCode3134531",
    "detectionTime": "0",
    "serviceType": "bus_real_time"
  },
  "id": 2
}, {
  "geometry": {
    "type": "Point",
    "coordinates": [11.253791, 43.78007]
  },
  "type": "Feature",
  "properties": {
    "vehicleNum": "3137538",
    "line": "12",
    "direction": "Piazzale Michelangelo &#10132; Stazione Parcheggio",
    "tipo": "RealTimeInfo",
    "serviceUri": "busCode3137538",
    "detectionTime": "2",
    "serviceType": "bus_real_time"
  },
  "id": 3
}, ... ]
}

```

Note:

Currently it provides the position of ATAF&Linea buses based on the timetable.

3.14 Feedback API

These APIs are used from applications to provide some kind of feedback on services from real users like photos of the services, comments on the services, ratings of the services.

3.14.1 Rating and comment API

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/feedback
API accepts a star rating (1-5) and/or a comment on a specific service. Comments are not automatically associated with the service, a moderator has to validate the comment provided.	
Parameters:	
<i>serviceUri</i>	URI identifying a service
<i>stars</i>	value 1 to 5 (if omitted no ratings is provided)
<i>comment</i>	comment provided by the user

<i>lang</i>	the language used in the comment
<i>uid</i>	a user identifier associated with the user providing the data
Results:	
the API fails using HTTP error code 404 if the serviceURI is not valid, stars or comment is not provided or user id is not provided.	
Examples:	
http://servicemap.disit.org/WebAppGrafo/api/v1/feedback?service=...&stars=2&comment=a%20comment&uid=	
Notes:	

3.14.2 Service Photo API

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/photo/
API accepts in POST as a multipart form the serviceUri, the user id and a photo in jpeg or png format. The photo provided is not automatically associated with the serviceUri a moderator will check it and decide.	
Parameters:	
<i>serviceUri</i>	URI identifying a service
<i>uid</i>	a user identifier associated with the user providing the data
<i>file</i>	a part named “file” with the photo to be uploaded, the part should contain the mimetype or the filename
Results:	
the API fails using HTTP error code 404 if the serviceURI is not valid, user id is not provided or a part named “file” is not present and the mimetype of this file cannot be found or if it’s not valid.	
Examples:	
NA	
Notes:	

3.14.3 Last contributions API

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/feedback/last
API reports a list of the last photos, comments and starred services from the users.	
Parameters:	
<i>uid</i>	a user identifier
<i>lang</i>	ISO 2 chars language code (e.g. “it”, “en”, “fr”, “de”) to be used for returned descriptions if available in multiple languages. Currently for languages other than “it” and “en” it returns “en” descriptions. (if parameter is missing “en” is assumed)
<i>format</i>	only json
Results:	
the API reports a JSON object with informations on the last contributions..	
Examples:	
http://servicemap.disit.org/WebAppGrafo/api/v1/feedback/last <pre>{ "LastPhotos": [{ "serviceUri": "http://www.disit.org/km4city/resource/af388d64a33b2624456a9a268ab01b54", "typeLabel": "Free WiFi point", "serviceType": "TourismService_Wifi", "long": "11.25355", "lat": "43.77682", "serviceName": "Firenze WIFI", }]</pre>	

<pre> "photo": "http://servicemap.disit.org/WebAppGrafo/api/v1/photo/file-5690474034488739316.jpg", "photoThumb": "http://servicemap.disit.org/WebAppGrafo/api/v1/photo/thumbs/file-5690474034488739316.jpg", "photoOrig": "http://servicemap.disit.org/WebAppGrafo/api/v1/photo/originals/file-5690474034488739316.jpg", "timestamp": "2017-01-22 16:38:20.0" }, ...], "LastComments": [{ "serviceUri": "http://www.disit.org/km4city/resource/cd9fa722072d84aa47d5bc6a74932c46", "typeLabel": "Museum", "serviceType": "CulturalActivity_Museum", "long": "11.263607", "lat": "43.769848", "serviceName": "MUSEO DI CASA BUONARROTI", "comment": "Palazzo del seicento comprato da Michelangelo nel quale si trovano diverse sculture e disegni di Michelangelo", "timestamp": "2016-12-17 09:04:14.0" }, ...], "LastStars": [{ "serviceUri": "http://www.disit.org/km4city/resource/20950a98d5fc0d1d69115d2b531b7793", "typeLabel": "Museum", "serviceType": "CulturalActivity_Museum", "long": "11.263603", "lat": "43.769836", "serviceName": "CASA_BUONARROTI", "stars": 5, "timestamp": "2016-12-17 10:14:37.0" }, ...] } </pre>
Notes:

3.15 Recommender API

URL	http://screcommender.km4city.org/SmartCityRecommender/
The API suggests a set of services near a user, the services are grouped depending on the user profile and it can suggest services on the basis of past user behavior (svd=true) or only by position.	
Parameters:	
<i>action</i>	have to be “recommend”
<i>user</i>	user identifier
<i>profile</i>	profile of the user (one of tourist, student, commuter, citizen, all)
<i>language</i>	user language (one of en, it, fr, es, de)
<i>latitude</i>	latitude in decimal format
<i>longitude</i>	longitude in decimal format
<i>distance</i>	the search range from GPS position in km
<i>mode</i>	optional, it can be “gps” or “manual”, states if the position provided is real user position acquired by a device or manually identified from the user on a map.
<i>version</i>	optional, version of the application, if provided the API suggests also tweets from some channels, depending on user profile, on Twitter Vigilance (e.g. PAAAlert, PAMeteoNews, PAMProtCivile)
<i>aroundme</i>	optional, it can be true or false (default false), if true imply svd false and it provides recommendation of the nearest services regardless if they have been already suggested.
<i>svd</i>	optional, it can be true or false (default true), if true and <i>aroundme</i> is false or missing the API chooses the category of services to suggest on the basis of past user of behavior (searches made and viewed services) and it can suggest services that are quite far away from user position.
<i>alreadyRecommended</i>	optional, it can be true or false (default false), if false and <i>aroundme</i> is false or missing the API does not suggest services already suggested in the last seven days.
Results:	

the API provides a JSON array of groups of suggested services (the groups names and types depend on the user profile), for each group are provided an array of suggestions with at most 3 services, the label of the group in the language provided (default English), a priority (used to order the groups) and a group identifier. For the tweets are provided some information as the message, the twitter user, the date, etc. The following is an example:

```
[{
  "suggestions": [ ... ],
  "label": "Things to do",
  "priority": 1,
  "group": "Things to do"
}, {
  "suggestions": [ ... ],
  "label": "Events",
  "priority": 2,
  "group": "Events"
}, {
  "suggestions": [ ... ],
  "label": "Wine and Food",
  "priority": 3,
  "group": "Wine and Food"
}, {
  "suggestions": [ ... ],
  "label": "Places Nearby",
  "priority": 4,
  "group": "Places Nearby"
}, {
  "suggestions": [ ... ],
  "label": "Services and Utilities",
  "priority": 5,
  "group": "Services and Utilities"
}, {
  "suggestions": [ ... ],
  "label": "Transfer Services",
  "priority": 6,
  "group": "Transfer Services"
}, {
  "suggestions": [...],
  "label": "Education",
  "priority": 8,
  "group": "Education"
}, {
  "suggestions": [ ... ],
  "label": "Bus",
  "priority": 9,
  "group": "Bus"
}, {
  "suggestions": [ ... ],
  "label": "Financial Services",
  "priority": 10,
  "group": "Financial Services"
}, {
  "suggestions": { ... },
  "label": "Weather",
  "priority": 11,
  "group": "Weather"
}, {
  "suggestions": [{
    "Tweet": {
      "hashtagsOnTwitter": "#Toscana",
      "geo_lat": "0.00000",
      "publicationTime": "2017-01-20 09:45:00",
      "twitterUser": "arpatoscana",
      "links": "https://t.co/bpej68mTli http://bit.ly/2ixn46s https://t.co/u6maxPUtjz
https://twitter.com/i/web/status/822364292355739650",
      "message": "Come ha lavorato ARPAT nel 2016: il parere dei cittadini della #Toscana https://t.co/bpej68mTli\u0026
https://t.co/u6maxPUtjz",
      "lang": "it",
      "twitterId": "822364292355739650",
      "retweetCount": "0",
      "favoriteCount": "0",
      "geo_long": "0.00000"
    }
  }
}, ...],
```

<pre> "label": "Twitter Environment", "priority": 12, "group": "Twitter3" }, { "suggestions": [...], "label": "Twitter News", "priority": 14, "group": "Twitter1" }, { "suggestions": [...], "label": "Twitter Alert", "priority": 15, "group": "Twitter2" }] </pre>
Examples:
http://screcommender.km4city.org/SmartCityRecommender/?action=recommend&user=3043b85d23d6f4879e1765c2c2e431cbc71d393065af06b03486ba4a04642b5b&profile=student&language=en&latitude=43.7727&longitude=11.2532&distance=1&version=1
Notes:

3.16 Shortest path finder API

Warning this API is under development.

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/shortestpath
Parameters:	
<i>source</i>	"<lat>;<lng>" or service URI of the starting point
<i>destination</i>	"<lat>;<lng>" or service URI of the destination
<i>routeType</i>	can be "public_transport", "feet", "car", "bike_security"
<i>maxFeetKM</i>	maximum distance by feet
<i>startDatetime</i>	datetime of start
<i>format</i>	json or html
Results:	
the API provides a JSON object with the path from the source to the destination.	
Examples:	
NA	
Notes:	

3.17 Image caching API

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/imgcache
The API provides a cache for the given image url, it downloads the image, scales it to the thumbnail or medium size depending on the size requested and save it for future requests.	
Parameters:	
<i>imageUrl</i>	url to the image
<i>size</i>	the size of the image to be produced, it can be equal to "thumb", "medium" or a number between 1 and 2000 pixels.
Results:	
It provides the scaled image produced in the same format as the original to be fit in a square of <i>size</i> x <i>size</i> , if the url is not an image it redirects to the original url.	

Examples:
http://servicemap.disit.org/WebAppGrafo/api/v1/imgcache?imageUrl=http://www.florenceheritage.it/mobileApp/immagini/zocchi/148.jpg&size=thumb
Notes:

4 Linked data and SPARQL access

The data is currently available also using the standard W3C linked data protocol. It allows getting a machine readable representation of a resource like <http://www.disit.org/km4city/resource/048017> as RDF/XML format. In the HTTP request protocol the header parameters Accept with “application/rdf+xml” should be specified while if the resource url is open in a web browser a (quite) human readable html version is generated.

Details on the RDF/XML format can be found at <https://www.w3.org/TR/rdf-syntax-grammar/>

Data can be also accessed using the standard W3C SPARQL 1.1 language and SPARQL query protocol at <http://servicemap.disit.org/WebAppGrafo/sparql>

Details on SPARQL 1.1 can be found at <https://www.w3.org/TR/sparql11-overview/>

At http://log.disit.org/sparql_query_frontend/ a query user interface can be found to play with SPARQL queries with some examples. Moreover the knowledge graph can be navigated using the Linked Open Graph viewer available at <http://log.disit.org>

5 Bibliografia

- ..

6 Acronimi

- API: Application Program Interface
- AVL: Automatic vehicle location
- AVM: Automatic Vehicle Monitoring
- BDaaS: Big Data as a Service
- CAP principle: Consistency Availability Partition Tolerance principle
- CBB: Content Based Billing
- CBB: Content Based Billing
- CEN: European Committee for Standardization
- DBMS: database management system
- FCD: Floating Cellular Data
- GPRS: General packet radio service
- GPS: Global positioning System
- GSM: Global System for Mobile
- ICT: Information and Communication Technologies
- ITS: Intelligent Transport Systems
- LCD: liquid-crystal display
- LOD: linked open data
- MC: Mobile Collector
- MMS: Multimedia Messaging Service
- NLP: Natural Language Processing
- NoSQL: no SQL database
- OD: open data
- OD: Open Data

- OGC: Open Geospatial Consortium
- OWL: Web Ontology Language
- PA: Pubblica Amministrazione
- PMI: Piccola e Media Impresa
- PMS: Private Mobile Systems
- POS: part-of-speech
- RDF: Resource Description Framework
- RFID: Radio Frequency IDentification o Identificazione a radio frequenza
- RTTI: Real-time Travel & Traffic Information
- SDI: Spatial Data Infrastructures
- SII: sistema di interoperabilità integrato
- SIMONE: progetto Simone
- SMS: Short Message Service
- SN: social networking, oppure sensor network
- SOA: Service Oriented Architecture
- SOAP: Simple Object Access Protocol
- SSAMM: Agenzia per la Mobilità Metropolitana strumenti di supporto, TOSCANA
- TPEG: Transport Protocol Experts Group
- TPL: gestore trasporto pubblico locale
- UML: Unified Modeling Language
- UMTS: Universal Mobile Telecommunications System
- UTC: Urban Traffic Control
- UUDI: Universal Description Discovery and Integration
- V2I: Vehicle-to-Infrastructure
- V2V: vehicle-to-vehicle
- VMS: Variable Message Sign
- VWSN: Vehicular Wireless Sensor Networks
- W3C: World Wide Web Consortium
- WSD: Word Sense Disambiguation
- WSDL: Web Services Description Language
- WSN: Wireless Sensor Networks
- XMI: XML Metadata Interchange standard di OMG
- XML: Extensible Markup Language
- ZTL: Zona a Traffico Limitato