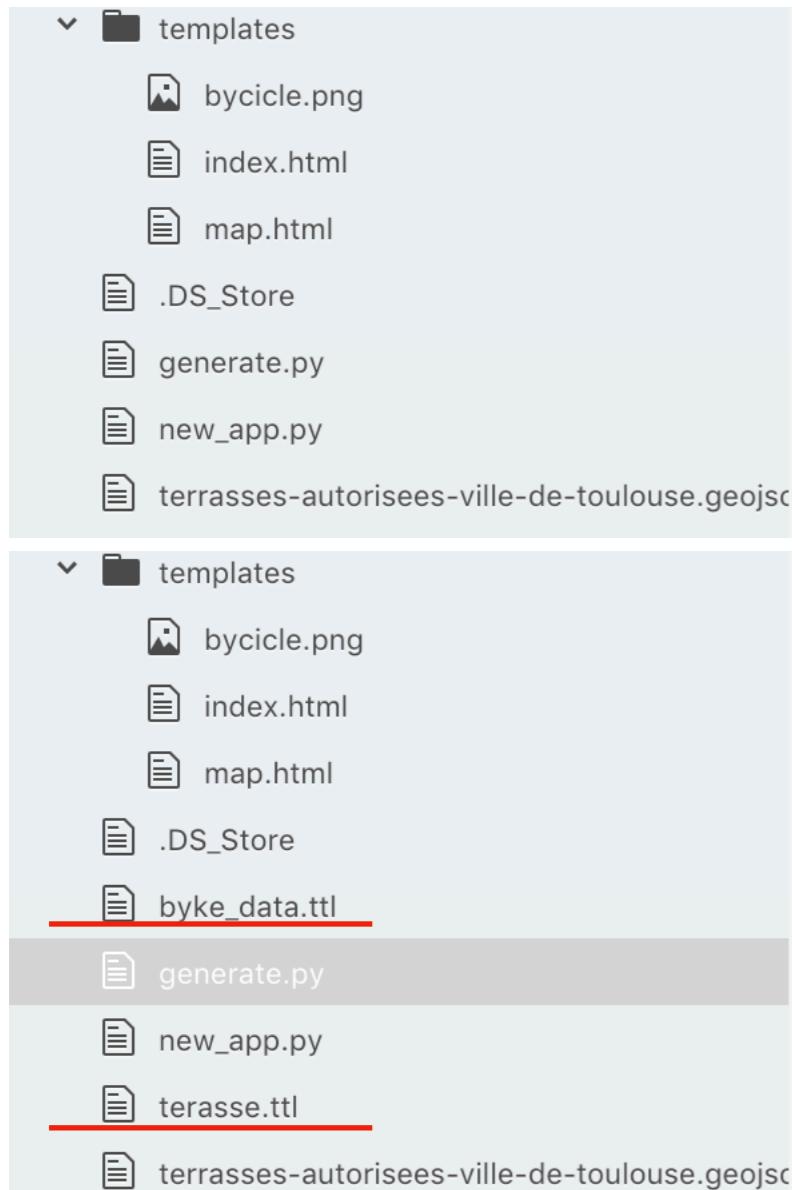


INITIAL Set Up and Configuration:

[RDFLIB](#), [Flask](#), [json](#), [urllib](#), [SPARQLEWrapper](#) - libraries for python.

Name of folder is [rdf_project_final](#). Inside folder:

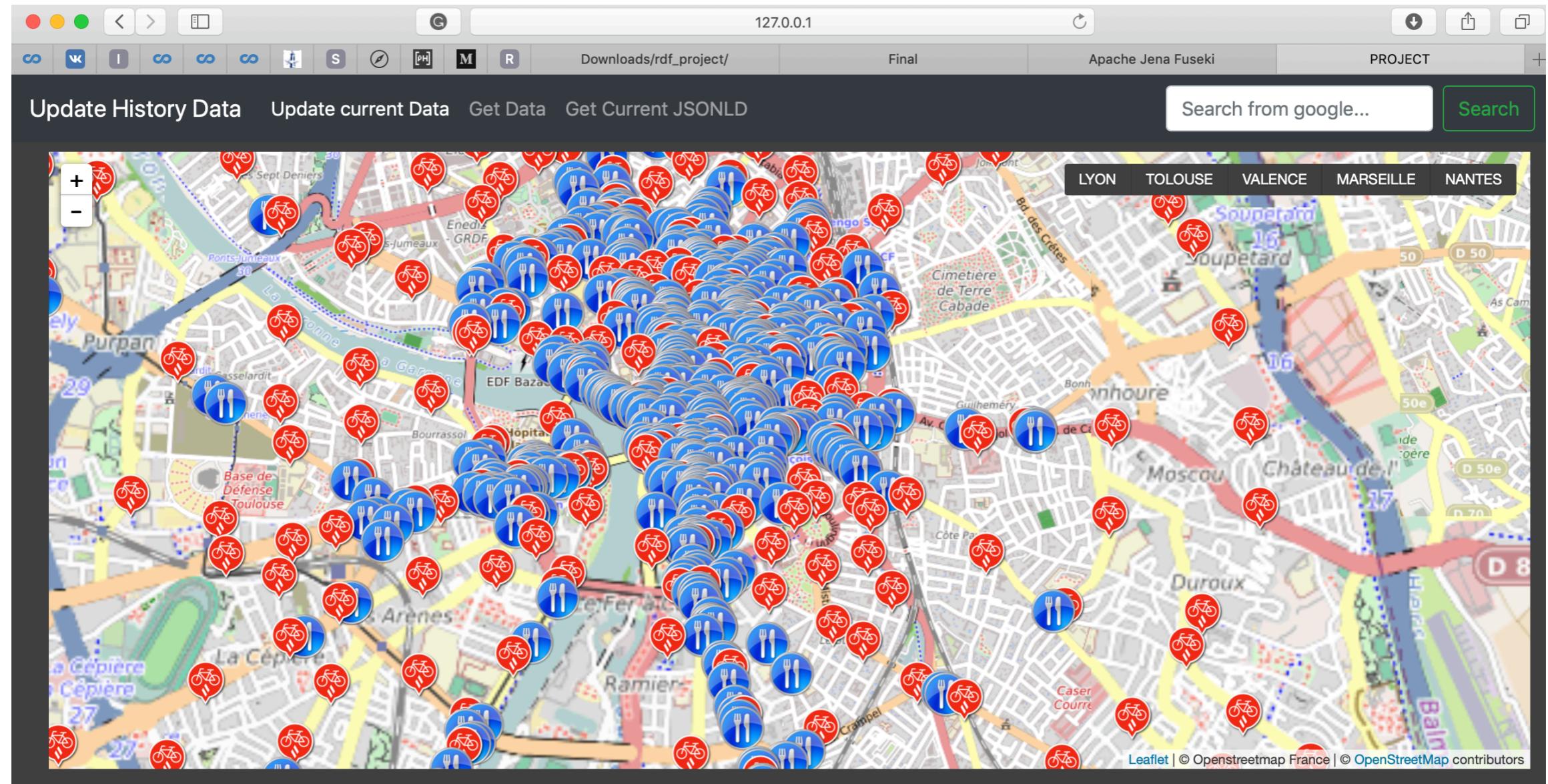


1. Run [generate.py](#)(it will generate starting data([byke_data.ttl](#), [terasse.ttl](#))!!! It may take sometime (for me 15-20 seconds)
2. Start Fuseki Server.
3. Create in Fuseki three data and name: [history](#), [current](#), [rest](#)
4. Upload the [byke_data.ttl](#) (generated data) to the [history](#) and to the [current](#)
5. Upload the [terasse.ttl](#) to the [rest](#)

Datasets on this server

dataset name	actions
/current	query add data info
/history	query add data info
/rest	query add data info

If previous steps done correctly run [new_app.py](#) and go to <http://127.0.0.1:5000> on your browser.



About me

[ZHANTIEUOV Eldiyar](#), Student at UJM/EMSE.

Phone: +33 7 82787411

E-mail: zhantieuov.eldiyar@gmail.com

Links: [Linkedin profile](#)

Address: 32 rue Emile Littré, SAINT ETIENNE

[View details](#)

Apache Jena Fuseki

Apache Jena Fuseki is a SPARQL server. It can run as a operating system service, as a Java web application (WAR file) , and as a standalone server. Fuseki is tightly integrated with TDB to provide a robust, transactional persistent storage layer, and incorporates Jena text query. It can be used to provide the protocol engine for other RDF query and storage systems

[Quick Start](#)

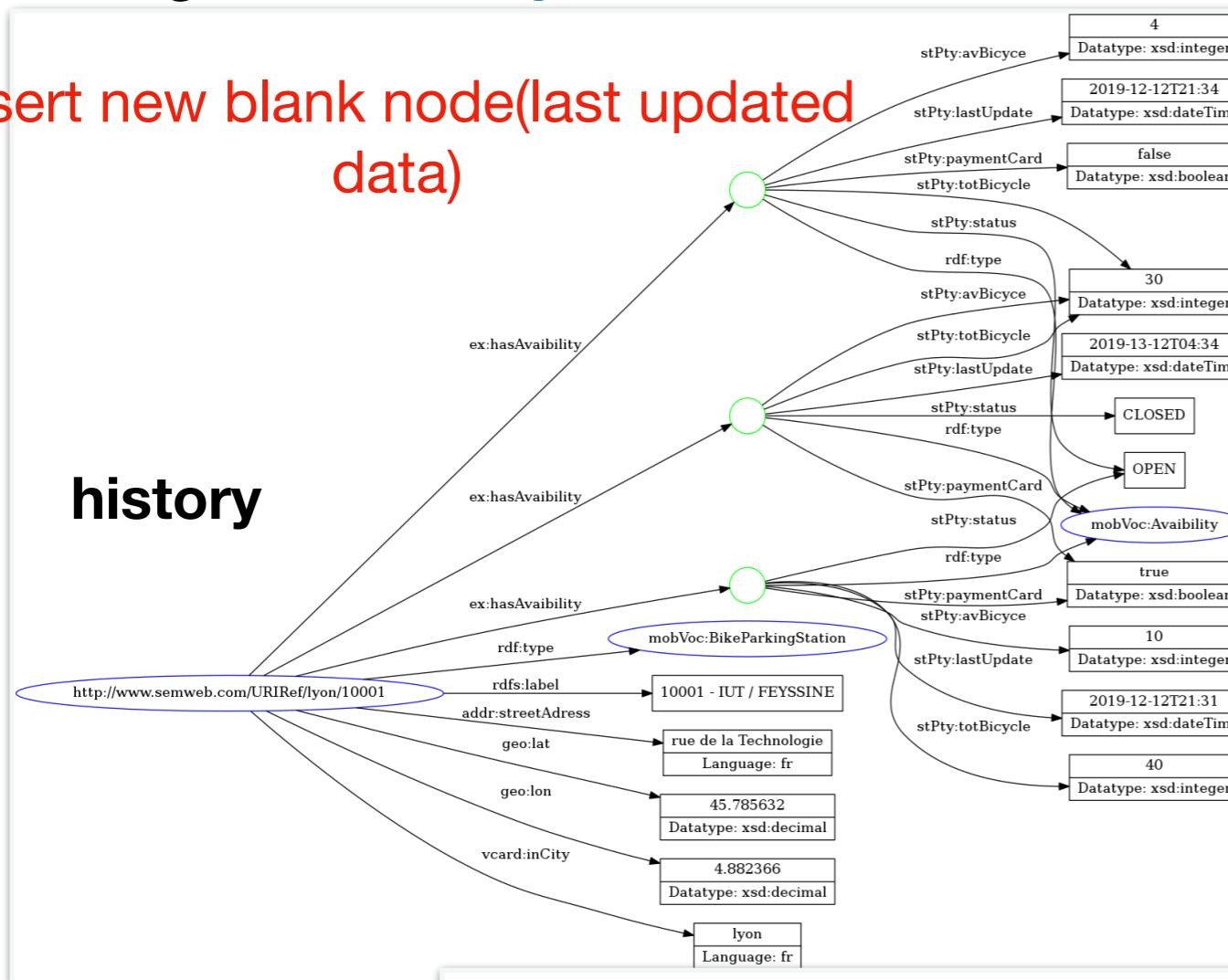
NavBar Buttons Info

Firs button "Update Fuseki history, we are adding last updated data from veloApi to our triple store". Button "Data", references to data used in this website(data from triple store, got by sparql query) "Update current" is updating our Fuseki server(updating current data)

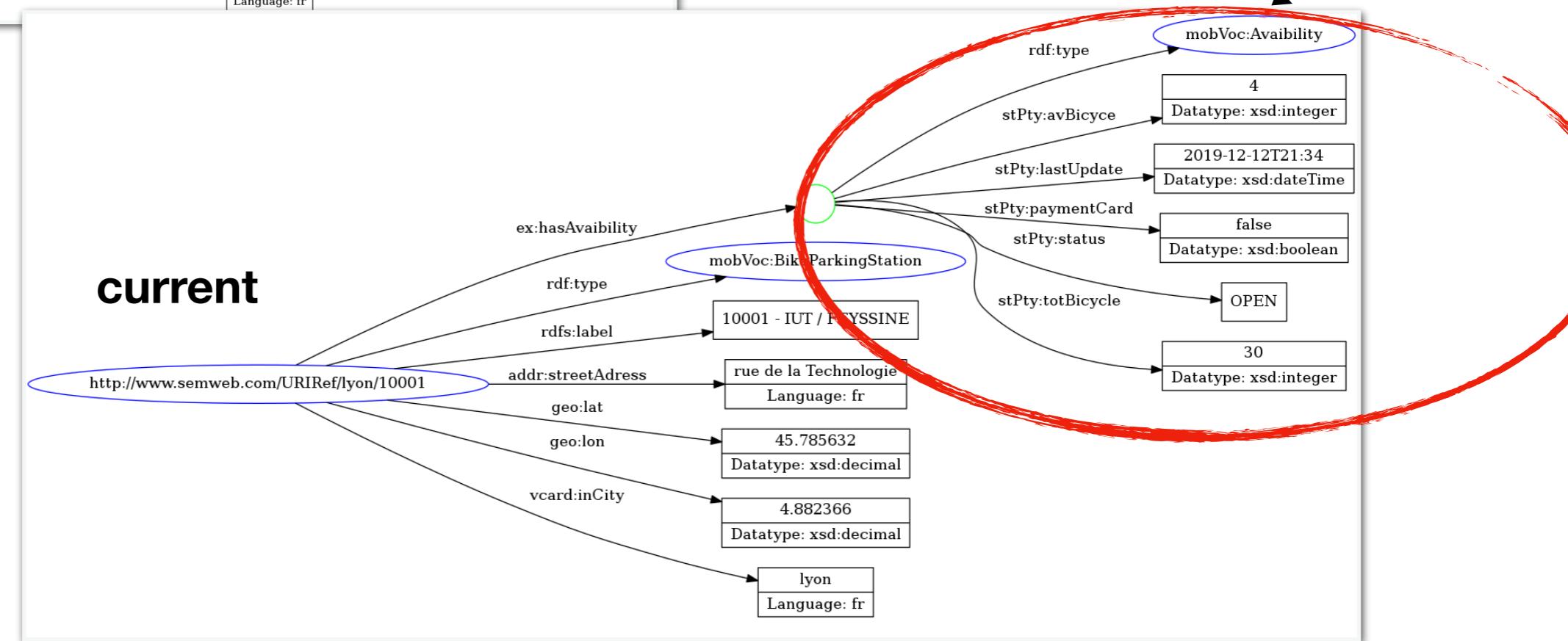
[Go to Fuseki Data](#)

The main goal of history and current data in TRIPLESTORE

Insert new blank node(last updated data)



history



current

Datasets on this server

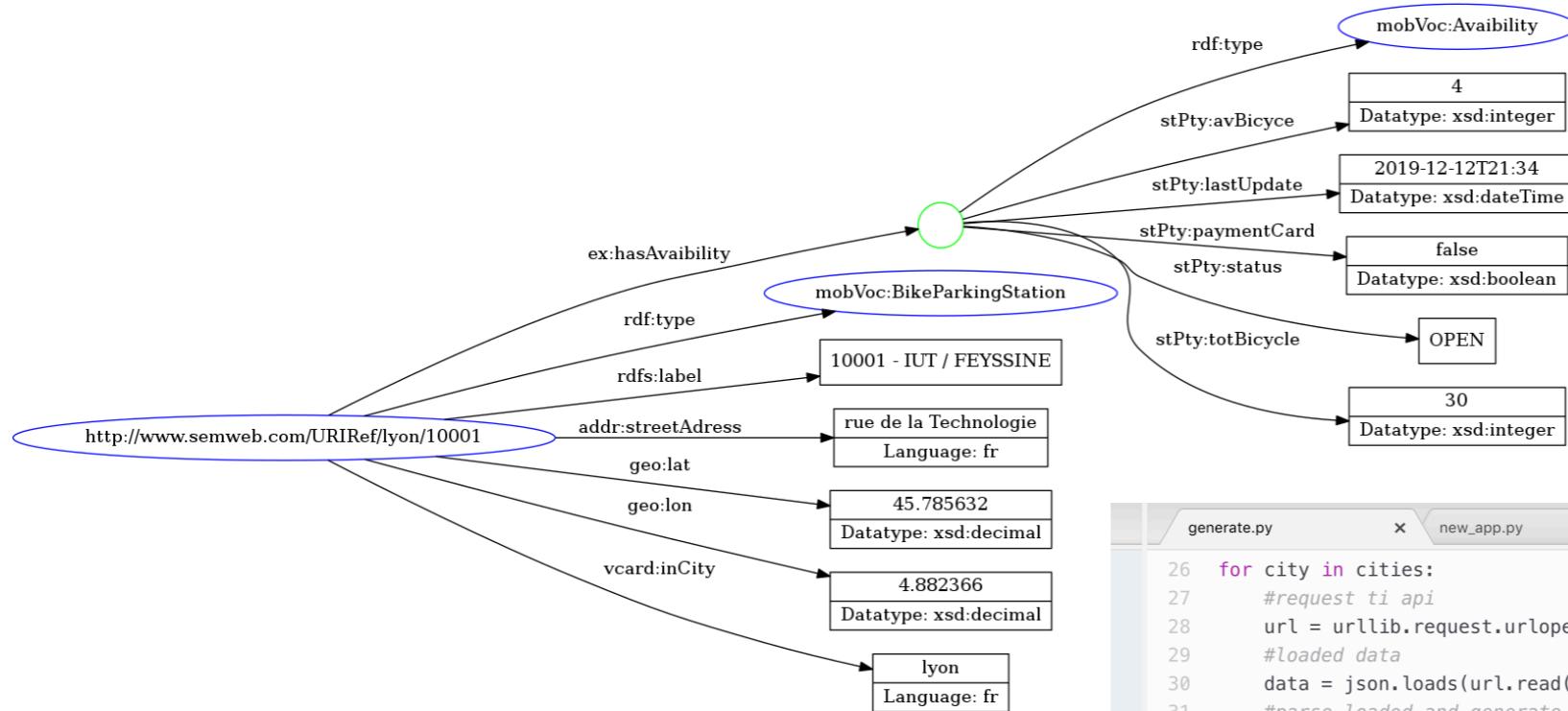
dataset name	actions
/current	query
/history	query

Delete old blank node and Insert new blank node (last_updated data)

Description of source codes

1. generate.py

This code runs at the beginning. It is generate initial data to triple store. It connects to bicycle api and get bicycle data in geoJson. For generating RDF in turtle format, I used RDFlib library. My generated data looks like:



```
generate.py  x new_app.py  x
26  for city in cities:
27      #request ti api
28      url = urllib.request.urlopen('https://api.jcdecaux.com/vls/v1/stations?contract=' + str(city) + '&apiKey=')
29      #loaded data
30      data = json.loads(url.read().decode(url.info().get_param('charset') or 'utf-8'))
31      #parse loaded and generate rdf turtle
32      for i in range(len(data)):
33          URIRef = URIRef('http://www.semweb.com/URIRef/' + data[i]['contract_name'] + '/' + str(data[i]['name']))
34          name = Literal(data[i]['name'], datatype=XSD.string)
35          city = Literal(data[i]['contract_name'], lang='fr')
36          address = Literal(data[i]['address'], lang='fr')
37          lat = Literal(data[i]['position']['lat'], datatype=XSD.decimal)
38          lon = Literal(data[i]['position']['lng'], datatype=XSD.decimal)
39          availability = BNode()
40
41          avail_bikes = Literal(data[i]['available_bikes'], datatype=XSD.integer)
42          total_bikes = Literal(data[i]['bike_stands'], datatype=XSD.integer)
43          banking = Literal(data[i]['banking'], datatype=XSD.boolean)
44          date = Literal("12-09-2019T13:05", datatype=XSD.date)
45          status = Literal(data[i]['status'], datatype=XSD.string)
46          last_update = Literal(datetime.fromtimestamp(data[i]['last_update'])/1000).strftime('%Y-%m-%dT')
47
48          g.namespace_manager.bind('geo', geo, override=False)
49          g.namespace_manager.bind('vcard', vcard, override=False)
```

Look at the code carefully.

1. new_app.py

This is core of our Web Application. I build this with Flask. This code is responsible for Select-Update-Insert-Delete our triplestore with real-time data from API. And it connects our backend with fronted.

Press the Update History button on navigating bar of website

Or if you go to <http://127.0.0.1:5000/update> update() function will send Sparql query for inserting last updated data our history dataset. It may take some time(for me 20-30 seconds)

```
71 @app.route('/update')
72 def update():
73     cities = ['valence', 'marseille', 'lyon', 'nantes', 'toulouse']
74 #sparql = SPARQLWrapper("http://localhost:3030/eldiyar/update")
75     for city in cities:
76         url = urllib.request.urlopen('https://api.jcdecaux.com/vls/v1/stations?contract=' + str(city) + '&
77 data = json.loads(url.read().decode(url.info().get_param('charset') or 'utf-8'))
78     for i in range(len(data)):
79         sparql = SPARQLWrapper("http://localhost:3030/history/")
80         URIRef = ('<http://www.semweb.com/URIRef/' + data[i]['contract_name']) + '/' + str(data[i]['numb
81         avail_bikes = data[i]['available_bikes']
82         last_update = datetime.fromtimestamp(data[i]['last_update']) / 1000).strftime('%Y-%m-%dT%R')
83         sparql.setQuery(''''PREFIX ex: <http://www.semweb.com/2001-schema#>
84                         PREFIX mobVoc: <http://schema.mobivoc.org/>
85                         PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
86                         PREFIX stPty: <http://www.semweb.org/2006/BycicleStation/property#>
87                         INSERT DATA {''' + URIRef + '''ex:hasAvailibility [ a mobVoc:Availabil
88                                         stPty:avBicyce %d
89                                         stPty:lastUpdate "%
90                                         stPty:paymentCard %s
91                                         stPty:status "%
92                                         stPty:totBicycle %d
93                                         ]
94                         }''' % ( avail_bikes, last_update, data[i]['banking'], data[i]['status'], dat
```

Before

```
Downloads/rdf_pr...  
  
@prefix mobVoc: <http://schema.mobivoc.org/> .  
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .  
@prefix ex: <http://www.semweb.com/2001-schema#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix xml: <http://www.w3.org/XML/1998/namespace> .  
@prefix geoNames: <http://www.geonames.org/ontology#> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .  
@prefix addr: <http://schemas.tails.com/2005#adresss/schema#> .  
@prefix stPty: <http://www.semweb.org/2006/BycicleStation/property#>  
  
<http://www.semweb.com/URIRef/lyon/10001>  
    a          mobVoc:BikeParkingStation ;  
    rdfs:label "10001 - IUT / FEYSSINE" ;  
    addr:streetAdress "rue de la Technologie"@fr ;  
    ex:hasAvailibility [ a          mobVoc:Availibility ;  
                          stPty:avBicyce 3 ;  
                          stPty:lastUpdate "2019-12-13T04:42:18"  
                          stPty:paymentCard false ;  
                          stPty:status "OPEN" ;  
                          stPty:totBicycle 30  
                      ] ;  
    geo:lat 45.785632 ;  
    geo:lon 4.882366 ;  
    vcard:inCity "lyon"@fr .
```

after

```
Downloads/rdf_pro...  
  
@prefix mobVoc: <http://schema.mobivoc.org/> .  
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .  
@prefix ex: <http://www.semweb.com/2001-schema#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix xml: <http://www.w3.org/XML/1998/namespace> .  
@prefix geoNames: <http://www.geonames.org/ontology#> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .  
@prefix addr: <http://schemas.tails.com/2005#adresss/schema#> .  
@prefix stPty: <http://www.semweb.org/2006/BycicleStation/property#> .  
  
<http://www.semweb.com/URIRef/lyon/10001>  
    a          mobVoc:BikeParkingStation ;  
    rdfs:label "10001 - IUT / FEYSSINE" ;  
    addr:streetAdress "rue de la Technologie"@fr ;  
    ex:hasAvailibility [ a          mobVoc:Availibility ;  
                          stPty:avBicyce 3 ;  
                          stPty:lastUpdate "2019-12-13T04:42:18"^^xsd:date  
                          stPty:paymentCard false ;  
                          stPty:status "OPEN" ;  
                          stPty:totBicycle 30  
                      ] ;  
    ex:hasAvailibility [ a          mobVoc:Availibility ;  
                          stPty:avBicyce 0 ;  
                          stPty:lastUpdate "2019-12-13T18:55"^^xsd:date  
                          stPty:paymentCard false ;  
                          stPty:status "OPEN" ;  
                          stPty:totBicycle 30  
                      ] ;  
    geo:lat 45.785632 ;  
    geo:lon 4.882366 ;  
    vcard:inCity "lyon"@fr .
```

fuseki-server POST

```
zantileuoveldiyar — fuseki-server — java -Xmx1200M -cp ~/Downloads/rdf_pr...  
[2019-12-13 19:02:35] Fuseki   INFO [432] 200 OK (6 ms)  
[2019-12-13 19:02:35] Fuseki   INFO [433] POST http://localhost:3030/history  
[2019-12-13 19:02:35] Fuseki   INFO [433] 200 OK (7 ms)  
[2019-12-13 19:02:35] Fuseki   INFO [434] POST http://localhost:3030/history  
[2019-12-13 19:02:35] Fuseki   INFO [434] 200 OK (6 ms)  
[2019-12-13 19:02:35] Fuseki   INFO [435] POST http://localhost:3030/history  
[2019-12-13 19:02:35] Fuseki   INFO [435] 200 OK (7 ms)  
[2019-12-13 19:02:35] Fuseki   INFO [436] POST http://localhost:3030/history  
[2019-12-13 19:02:35] Fuseki   INFO [436] 200 OK (12 ms)  
[2019-12-13 19:02:35] Fuseki   INFO [437] POST http://localhost:3030/history  
[2019-12-13 19:02:35] Fuseki   INFO [437] 200 OK (9 ms)  
[2019-12-13 19:02:35] Fuseki   INFO [438] POST http://localhost:3030/history  
[2019-12-13 19:02:35] Fuseki   INFO [438] 200 OK (17 ms)  
[2019-12-13 19:02:35] Fuseki   INFO [439] POST http://localhost:3030/history  
[2019-12-13 19:02:35] Fuseki   INFO [439] 200 OK (8 ms)  
[2019-12-13 19:02:35] Fuseki   INFO [440] POST http://localhost:3030/history  
[2019-12-13 19:02:35] Fuseki   INFO [440] 200 OK (11 ms)  
[2019-12-13 19:02:35] Fuseki   INFO [441] POST http://localhost:3030/history  
[2019-12-13 19:02:35] Fuseki   INFO [441] 200 OK (12 ms)  
[2019-12-13 19:02:35] Fuseki   INFO [442] POST http://localhost:3030/history  
[2019-12-13 19:02:35] Fuseki   INFO [442] 200 OK (10 ms)  
[2019-12-13 19:02:35] Fuseki   INFO [443] POST http://localhost:3030/history  
[2019-12-13 19:02:35] Fuseki   INFO [443] 200 OK (13 ms)
```

Press the Update Current Data on navigating bar of website

Or if you go to http://127.0.0.1:5000/update_current `update_current()` function will send Sparql query for deleting old blank node and insert new last updated blank node

```
def update_current():
    cities = ['valence', 'marseille', 'lyon', 'nantes', 'toulouse']
    #sparql = SPARQLWrapper("http://localhost:3030/eldiyar/update")
    for city in cities:
        url = urllib.request.urlopen('https://api.jcdecaux.com/vls/v'
        data = json.loads(url.read()).decode(url.info().get_param('cha'
        for i in range(len(data)):
            sparql = SPARQLWrapper("http://localhost:3030/current/")
            URIReff = ('<http://www.semweb.com/URIRef/' + data[i]['co'
            avail_bikes = data[i]['available_bikes']
            last_update = datetime.fromtimestamp(data[i]['last_update'])
            sparql.setQuery('''PREFIX ex: <http://www.semweb.com/200
                            PREFIX mobVoc: <http://schema.mobivoc.
                            PREFIX xsd:<http://www.w3.org/2001/XMI
                            PREFIX stPty: <http://www.semweb.org/200
                            DELETE {
                                ?s ex:hasAvailibility ?o .
                                ?o ?p1 ?o1 .
                            }
                            WHERE {
                                VALUES (?s) { ('' + URIReff
                                ?s ex:hasAvailibility ?o .
                                OPTIONAL {
                                    ?o ?p1 ?o1 .
                                }
                            }
                        ''')
            sparql.query()
            sparql.update()
    
```

Before

```
@prefix mobVoc: <http://schema.mobivoc.org/> .  
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .  
@prefix ex: <http://www.semweb.com/2001-schema#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix xml: <http://www.w3.org/XML/1998/namespace> .  
@prefix geoNames: <http://www.geonames.org/ontology#> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .  
@prefix addr: <http://schemas.tails.com/2005#adressss/schema#> .  
@prefix stPty: <http://www.semweb.org/2006/BycicleStation/property#> .  
  
<http://www.semweb.com/URIRef/valence/195>  
a mobVoc:BikeParkingStation ;  
rdfs:label "195_CALLE_GIORGETA" ;  
addr:streetAdress "Giorgeta - Roig de Corella"@fr ;  
ex:hasAvailability [ a mobVoc:Availibility ;  
stPty:avBicyce 11 ;  
stPty:lastUpdate "2019-12-13T04:47:54"^^xsd:dateTime ;  
stPty:paymentCard false ;  
stPty:status "OPEN" ;  
stPty:totBicycle 16  
] ;  
geo:lat 39.45935205521343 ;  
geo:lon -0.384372117626406 ;  
vcard:inCity "valence"@fr .
```

after

```
@prefix mobVoc: <http://schema.mobivoc.org/> .  
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .  
@prefix ex: <http://www.semweb.com/2001-schema#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix xml: <http://www.w3.org/XML/1998/namespace> .  
@prefix geoNames: <http://www.geonames.org/ontology#> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .  
@prefix addr: <http://schemas.tails.com/2005#adressss/schema#> .  
@prefix stPty: <http://www.semweb.org/2006/BycicleStation/property#> .  
  
<http://www.semweb.com/URIRef/valence/195>  
a mobVoc:BikeParkingStation ;  
rdfs:label "195_CALLE_GIORGETA" ;  
addr:streetAdress "Giorgeta - Roig de Corella"@fr ;  
ex:hasAvailability [ a mobVoc:Availibility ;  
stPty:avBicyce 8 ;  
stPty:lastUpdate "2019-12-13T19:10"^^xsd:dateTime ;  
stPty:paymentCard false ;  
stPty:status "OPEN" ;  
stPty:totBicycle 16  
] ;  
geo:lat 39.45935205521343 ;  
geo:lon -0.384372117626406 ;  
vcard:inCity "valence"@fr .
```

fuseki-server POST

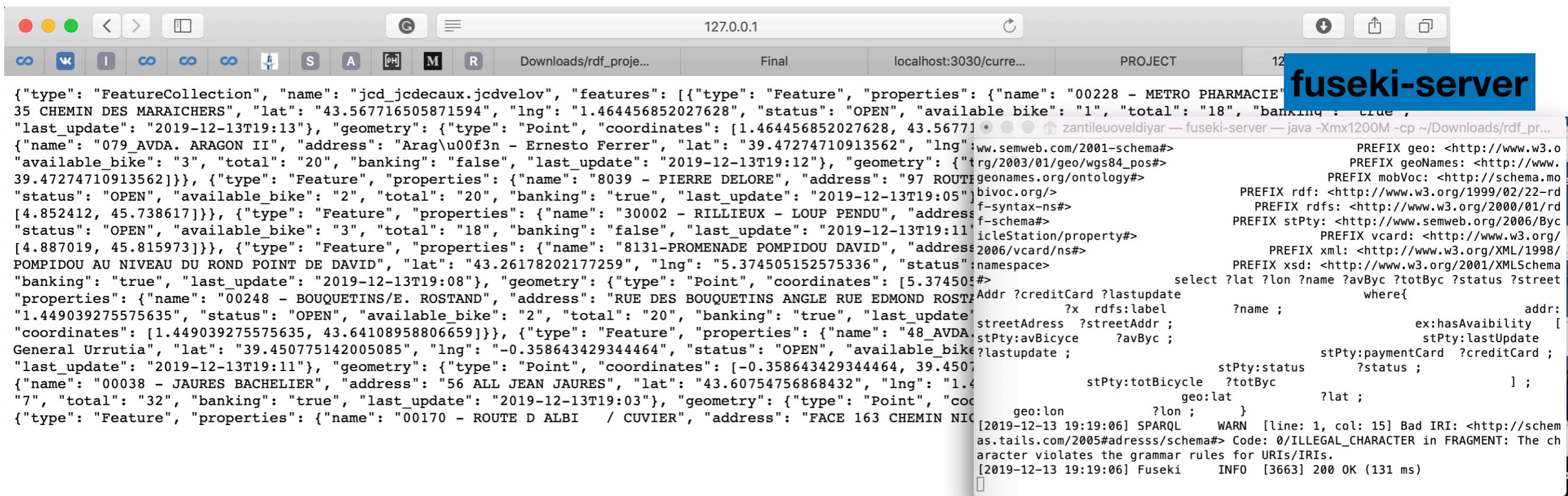
```
zantileuoveldiyar — fuseki-server — java -Xmx1200M -cp ~/Downloads/rdf_pr...  
/ [2019-12-13 19:13:23] Fuseki INFO [2039] 200 OK (5 ms)  
[2019-12-13 19:13:23] Fuseki INFO [2040] POST http://localhost:3030/current  
/ [2019-12-13 19:13:23] Fuseki INFO [2040] 200 OK (4 ms)  
[2019-12-13 19:13:23] Fuseki INFO [2041] POST http://localhost:3030/current  
/ [2019-12-13 19:13:23] Fuseki INFO [2041] 200 OK (5 ms)  
[2019-12-13 19:13:23] Fuseki INFO [2042] POST http://localhost:3030/current  
/ [2019-12-13 19:13:23] Fuseki INFO [2042] 200 OK (5 ms)  
[2019-12-13 19:13:23] Fuseki INFO [2043] POST http://localhost:3030/current  
/ [2019-12-13 19:13:23] Fuseki INFO [2043] 200 OK (7 ms)  
[2019-12-13 19:13:23] Fuseki INFO [2044] POST http://localhost:3030/current  
/ [2019-12-13 19:13:23] Fuseki INFO [2044] 200 OK (4 ms)  
[2019-12-13 19:13:23] Fuseki INFO [2045] POST http://localhost:3030/current  
/ [2019-12-13 19:13:23] Fuseki INFO [2045] 200 OK (4 ms)  
[2019-12-13 19:13:23] Fuseki INFO [2046] POST http://localhost:3030/current  
/ [2019-12-13 19:13:23] Fuseki INFO [2046] 200 OK (5 ms)
```

Press the Get Data button on navigating bar of website(for developers)

Or if you go to <http://127.0.0.1:5000/points> `data()` function will send query to Fuseki server and retrieve data from server in JSON-LD format. After inside function it will convert JSON-LD to geoJson and return to you.

```
@app.route('/points', methods=['GET'])
def data():
    sparql = SPARQLWrapper("http://localhost:3030/current/")
    sparql.setQuery('''' PREFIX addr: <http://schemas.tails.com/2005#adresss/schema#>
                      PREFIX ex: <http://www.semweb.com/2001-schema#>
                      PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
                      PREFIX geoNames: <http://www.geonames.org/ontology#>
                      SELECT ?addr ?ex ?geo ?name ?lat ?lon
                      WHERE { ?addr ex:type ?ex .
                            ?addr geo:lat ?lat .
                            ?addr geo:long ?lon .
                            ?addr geo:label ?name .
                            ?addr geo:point ?geo .
                            ?geo geo:lat ?lat .
                            ?geo geo:long ?lon }''')
```

result



Press the Get Current JSON-LD button on navigating bar of website(for developers)

Or if you go to http://127.0.0.1:5000/current_jsonld `data()` function will send query to Fuseki server and retrive data from server in JSON-LD format.

result

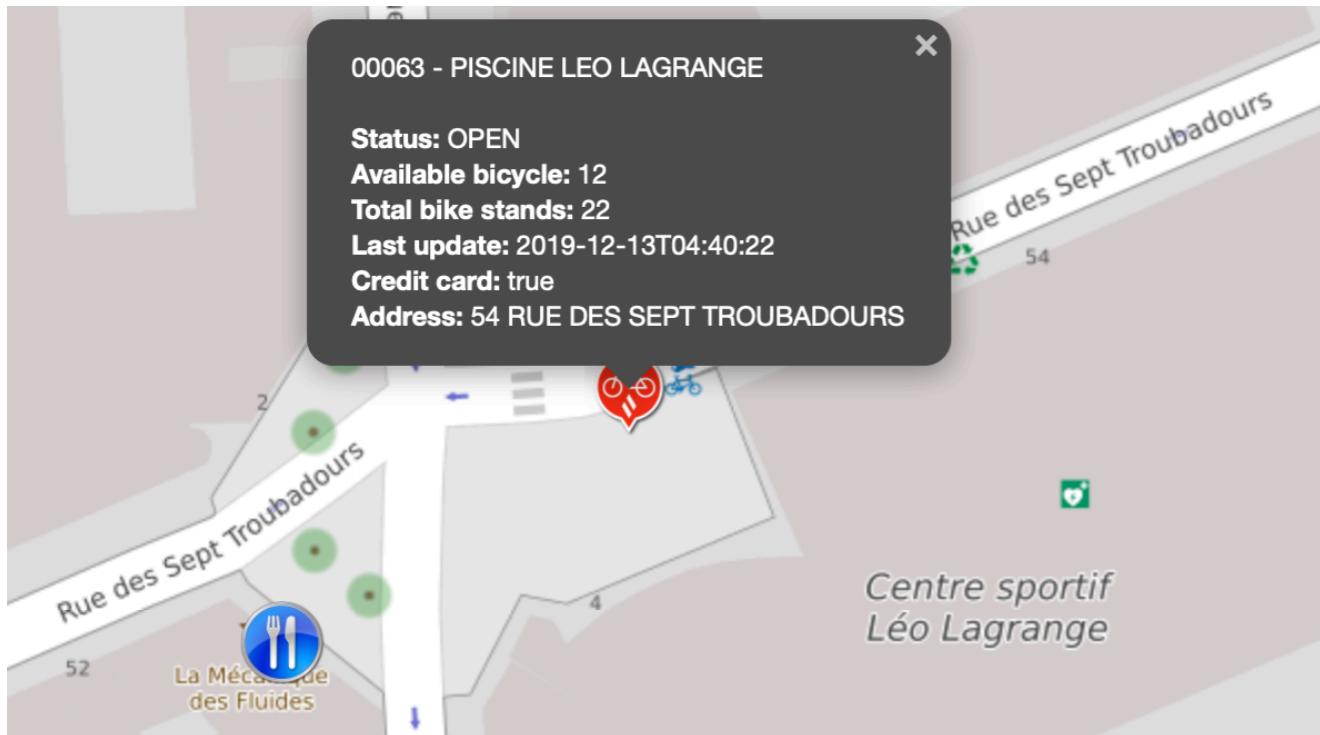
```
pp.route('/current_jsonld', methods=['GET'])
f current_jsonld():
    sparql = SPARQLWrapper("http://localhost:3030/current/")
    sparql.setQuery('''' PREFIX addr: <http://schemas.tails.com/#
PREFIX ex: <http://www.semweb.com/2001-#
PREFIX geo: <http://www.w3.org/2003/01/#
PREFIX geoNames: <http://www.geonames.o#
PREFIX mobVoc: <http://schema.mobivoc.o#
PREFIX rdf: <http://www.w3.org/1999/02/#
PREFIX rdfs: <http://www.w3.org/2000/01/#
PREFIX stPty: <http://www.semweb.org/200#
PREFIX vcard: <http://www.w3.org/2006/vc#
PREFIX xml: <http://www.w3.org/XML/1998/#
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#
select ?lat ?lon ?name ?avByc ?totByc ?status
where{
    ?x rdfs:label ?name ;
    addr:streetAdres
    ex:hasAvailibility
    ?x rdfs:label ?name ;
    stPty:streetAddr ;
    stPty:avBicycle ?avByc ;
    ?lastupdate ;
    stPty:status ?status ;
    stPty:totBicycle ?totByc
    geo:lat ?lat ;
    geo:lon ?lon ;
}'''')
    sparql.setReturnFormat(JSON)
    results = sparql.query().convert()
    return results
```

fuseki-server

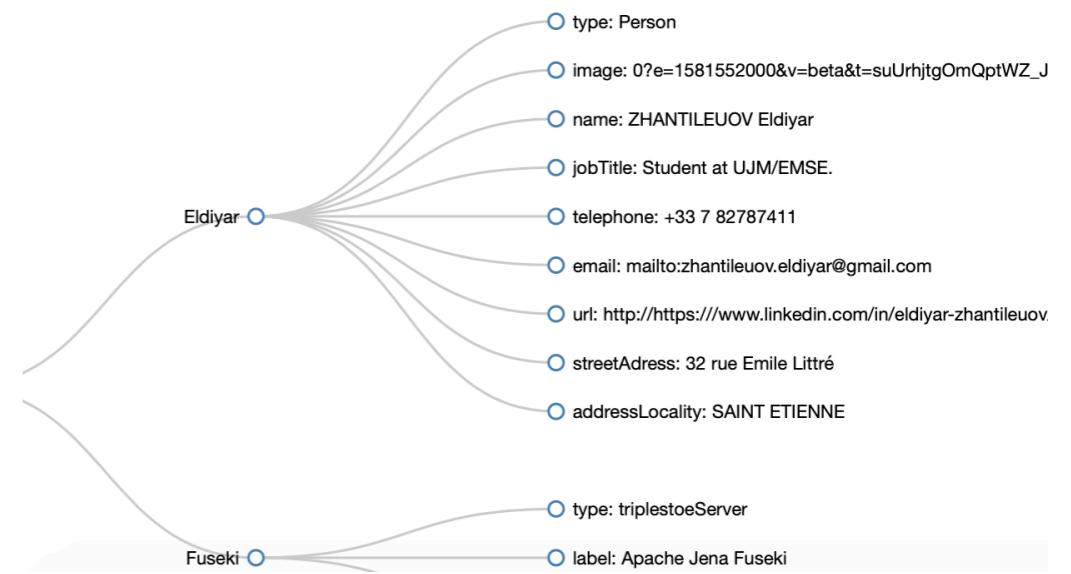
```
zantileuoveldiyar — fuseki-server — java -Xmx1200M -cp ~/Downloads/rdf.jar org.apache.jena.riot.RDFFormatFactory$ParserType$1 parse --format json-ld ~/Downloads/rdf.jsonld
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX geoNames: <http://www.geonames.org/ontology#>
PREFIX mobVoc: <http://schema.mobivoc.org/ns#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX stPty: <http://www.semweb.org/2006/09/property#>
PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>
PREFIX xml: <http://www.w3.org/XML/1998/namespace>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
select ?lat ?lon ?name ?avByc ?totByc ?status ?lastupdate ?stPty:paymentCard ?creditCard ?stPty:lastUpdate ?stPty:status ?totByc ?geo:lat ?lat ?geo:lon ?lon
where{
    ?x rdfs:label ?name ;
    addr:streetAdres
    ex:hasAvailibility
    ?x rdfs:label ?name ;
    stPty:streetAddr ;
    stPty:avBicycle ?avByc ;
    ?lastupdate ;
    stPty:status ?status ;
    stPty:totBicycle ?totByc
    geo:lat ?lat ;
    geo:lon ?lon ;
}
[2019-12-13 19:27:20] SPARQL   WARN [line: 1, col: 15] Bad IRI: <http://schemas.tails.com/2005#adresss/schema#> Code: 0/ILLEGAL_CHARACTER in FRAGMENT: T
[2019-12-13 19:27:21] Fuseki   INFO [3664] 200 OK (115 ms)
```

1. map.html

It is the front end of our application. Inside of this file is included bootstrap, leaflet.js, query.js and our RDFA data.



RDFA



```
var counties = $.ajax({
  url: "http://127.0.0.1:5000/points",
  dataType: "json",
  success: console.log("County data successfully loaded."),
  error: function(xhr) {
    alert(xhr.statusText)
  }
});
var rest = $.ajax({
  url: "http://127.0.0.1:5000/rest_tolous",
  dataType: "json",
  success: console.log("rest data successfully loaded."),
  error: function(xhr) {
    alert(xhr.statusText)
  }
});
```