Processing England's Live Traffic Information with Python

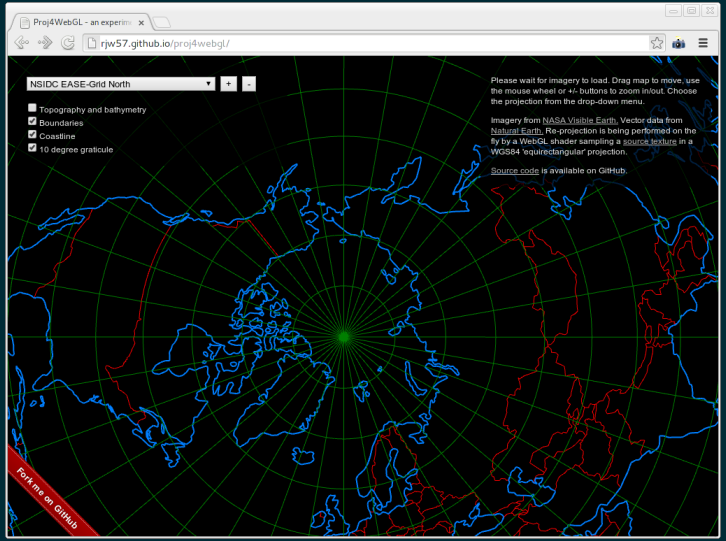
Rich Wareham

Department of Engineering
University of Cambridge

29 April 2013


MISSION CONTROL CENTER
esa
JAXA

http://rjw57.github.io/proj4webgl/

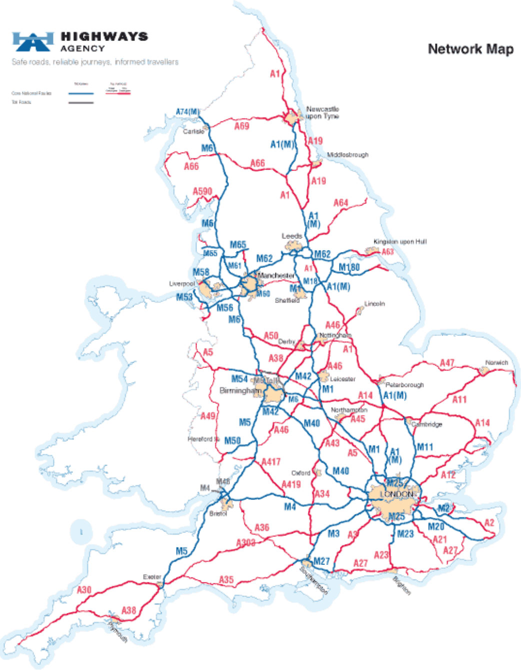



HIGHWAYS
AGENCY

HIGHWAYS AGENCY

Safe roads, reliable journeys, informed travellers

Network Map




A5
M6(T)
A38
M42
M54
M6
A5
M6(T)
M42
LEICESTER
M69
M6
M5
QUINTON
RCC
A38M
CMPG
M6
M69
M6
A46
M42
M40
A46
A45
M5
M40
M50
GLOUCESTERSHIRE
THAM
VALL
© Toby Smith

Search for a location: Postcode / House number Find

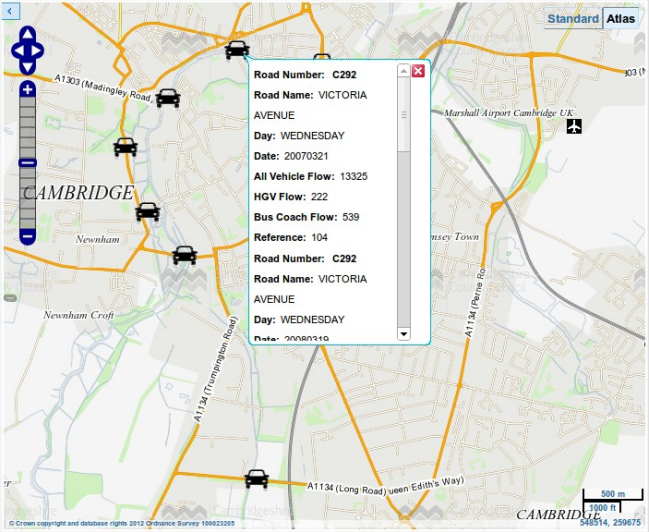
Report a problem

By selecting a location Cambridgeshire County Council will place cookies on your computer to remember your location.

Find Nearest

Show Map Categories

- + Council and Democracy 0/6
- + Environment and Planning 0/2
- + Leisure and Culture 0/10
- + NHS Choices 0/5
- + Schools and Learning 0/7
- + Transport and Streets 1/14
- + Highways Faults 0/8
- + Public Sector Assets 0/9
- + Flood Risk Assets 0/13



http://data.gov.uk/

HM Government

Log in or sign up

DATA.GOV.UK Beta

Opening up Government

Home | Data | Participate | Data requests | Apps | Location | Linked Data | Library | About

What is Linked Data?

An introduction to Linked Data

CODE OF PRACTICE [DATASETS] CONSULTATION RESPONSES

SEARCH AND PREVIEW

New map widgets

BATHING QUALITY DATA AS LINKED DATA

A BLOG BY Dr IAN DICKINSON

data.gov.uk technologies

New features to get at the data!

DEPARTMENTAL OPEN DATA STRATEGIES

9361

search data

Search

search other content

Search

Live traffic information from the Highways Agency

View | Resources (9) | History

Description

Live traffic information data showing traffic information on the strategic road network in England, maintained by the Highways Agency.

The separate datasets are referred to as "information products" and presented as XML, conforming to a Datex 2 data specification. A supporting document is provided to support understanding of the Datex 2 schema.

Information products consist of: Current Roadworks updated every 10 mins (1 M bytes) Future Roadworks updated every 6 hours (2 M bytes) Current Planned Events updated every 10 mins (10 K bytes) Future Planned Events updated every 6 hours (200K bytes) Unplanned Events updated every 10 mins (50 K bytes) All Traffic Data updated every 10 mins (50 M bytes) Journey Time Data updated every 10 mins (950 K bytes) Variable Message Signs updated every 10 mins (3 M bytes) The Variable Message Signs (VMS) information product will contain the status of all currently set or faulty VMS on the HA network. If the next publication no longer contains a VMS then either the setting or fault has been cleared. The payload file will contain VariableMessageSignSetting elements.

Matrix Signals updated every 10 mins (9 M bytes) The Matrix Signal information product will contain the status of all currently set or faulty Matrix Signals on the HA network. If the next publication no longer contains a particular Matrix Signal, then either the setting or fault has been cleared. The payload file will contain MatrixSignalSetting elements.

Data Resources (9)

Current roadworks	XML	Preview	Download
Future roadworks	XML	Preview	Download
Current planned events	XML	Preview	Download
Future planned events	XML	Preview	Download
Unplanned events	XML	Preview	Download
Journey time data	XML	Preview	Download
Traffic data	XML	Preview	Download
Variable Message Signs (VMS)	XML	Preview	Download
Matrix signals	XML	Preview	Download

Additional Information

Openness score:	★★☆
Theme	Transportation
Temporal coverage	21/12/2011
Geographic coverage	England
Date added to data.gov.uk	21/12/2011
Date updated on data.gov.uk	13/04/2013
Date update future	2012-01-06
Precision	longitude/latitude reference
Update frequency	live data updated automatically
Temporal granularity	point
Taxonomy URL	http://www.datex2.eu/sites/www.datex2.eu/files/NTIS_DATEX2v2_0_0.pdf
Mandate	No value

Developer Tools

The information on this page (the dataset metadata) is also available in JSON format:

http://data.gov.uk/api/rest/dataset/live-traffic-information-from-the-highways-agency-road-network

Read more about this site's CKAN API. About the API

This dataset has a permanent URI:

http://data.gov.uk/dataset/live-traffic-information-from-the-highways-agency-road-network

Login to post comments

Licence

- UK Open Government Licence (OGL)

Contact

Enquiries:
Paul Knights
- Email: paul.knights@highways.gsi.gov.uk
- Phone: 0113 2836846

FOI Contact:
HAIL
- Email: ha_info@highways.gsi.gov.uk

Tags

Transportation | highways-agency | traffic-information

Social

Tell the world about this dataset!

Tweet 0 | Share 0 | Google +1 0

Developer tools

JSON, API and URI for this dataset record

Do more with this data

- Share your app
- Share an idea
- Request new data

Publisher

Highways Agency

Data Resources (9)

Current roadworks	XML	Preview	Download
Future roadworks	XML	Preview	Download
Current planned events	XML	Preview	Download
Future planned events	XML	Preview	Download
Unplanned events	XML	Preview	Download
Journey time data	XML	Preview	Download
Traffic data	XML	Preview	Download
Variable Message Signs (VMS)	XML	Preview	Download
Matrix signals	XML	Preview	Download

http://hatrafficinfo.dft.gov.uk/feeds/datex/England/PredefinedLocationLinks/content.xml

http://hatrafficinfo.dft.gov.uk/feeds/datex/England/TrafficData/content.xml

DATEX

EASYWAY

Search

Search this site:

Search

User login

Username: *

Password: *

Log in

Create new account

Request new password

DATEX II newsletter

In Evidence

DATEX User Forum 2012 - Stockholm

The 2012 edition of the DATEX User Forum have been held in Stockholm on 20th-21st March

Session Presentations are now available from this link.

News

- DATEX II v2.1 has been released on DATEX II website
  DATEX II v2.1 is finalised and available for download on datex website www.datex2.eu Please visit

```
<script>
  <character>
    <name>James Kirk</name> <actor>William Shatner</actor>
  </character>
  <character>
    <name>Spock</name> <actor>Leonard Nimoy</actor>
  </character>
</script>
```

```
import lxml.objectify as lo
fobj = open('document.xml')
document = lo.parse(fobj)
script = document.getroot()
print(script.character[0].name)
# prints 'James Kirk'
print([c.actor for c in script.character])
# prints ['William Shatner', 'Leonard Nimoy']
```

PredefinedLocationLinks

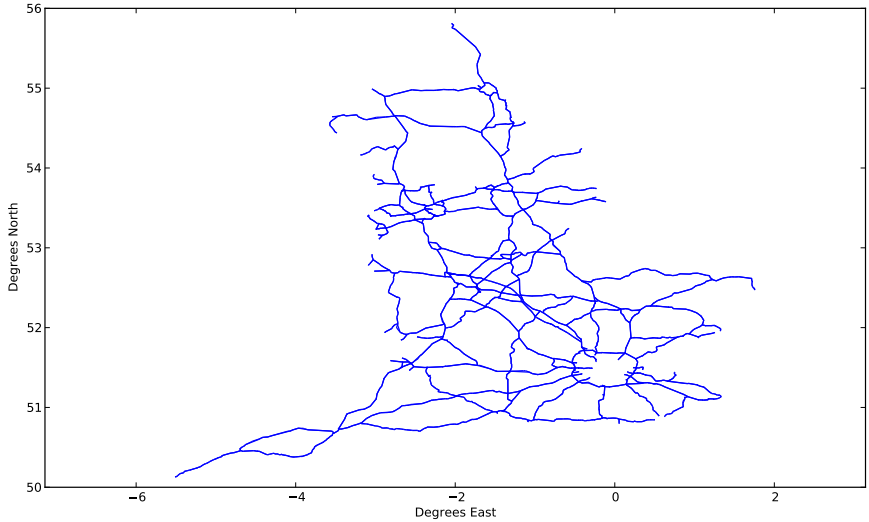
```
<?xml version="1.0" ?>
<d2LogicalModel>
  <payloadPublication>
    <publicationTime>
      2013-04-18T12:18:34+01:00
    </publicationTime>
    <predefinedLocationSet>
      <predefinedLocation>
        <!-- ... -->
      </predefinedLocation>
      <!-- etc, etc -->
    </predefinedLocationSet>
  </payloadPublication>
</d2LogicalModel>
```

```
<predefinedLocation id="Link12345">
  <predefinedLocation>
    <tpeglinearLocation>
      <to>
        <pointCoordinates>
          <latitude>52.892544</latitude>
          <longitude>-1.775278</longitude>
        </pointCoordinates>
      </to>
      <from>
        <pointCoordinates>
          <latitude>52.891373</latitude>
          <longitude>-1.768254</longitude>
        </pointCoordinates>
      </from>
    </tpeglinearLocation>
  </predefinedLocation>
</predefinedLocation>
```

```
pub = model.payloadPublication
location_sets = pub.predefinedLocationSet

link_segments = {}
for loc_set in location_sets:
    for loc_elem in loc_set.predefinedLocation:
        loc = loc_elem.predefinedLocation
        linear = loc.tpeglinearLocation

        to_coord = linear.to.pointCoordinates
        to = ( float(to_coord.longitude),
               float(to_coord.latitude) )
        from_coord = linear['from'].pointCoordinates
        from_ = ( float(from_coord.longitude),
                  float(from_coord.latitude) )

        link_segments[loc_elem.attrib['id']] = (from_, to)
```

```
link_segments = {
    "Link12345": (
        (52.892544, -1.775278),
        (52.891373, -1.768254)
    ),

    # etc, etc
}
```

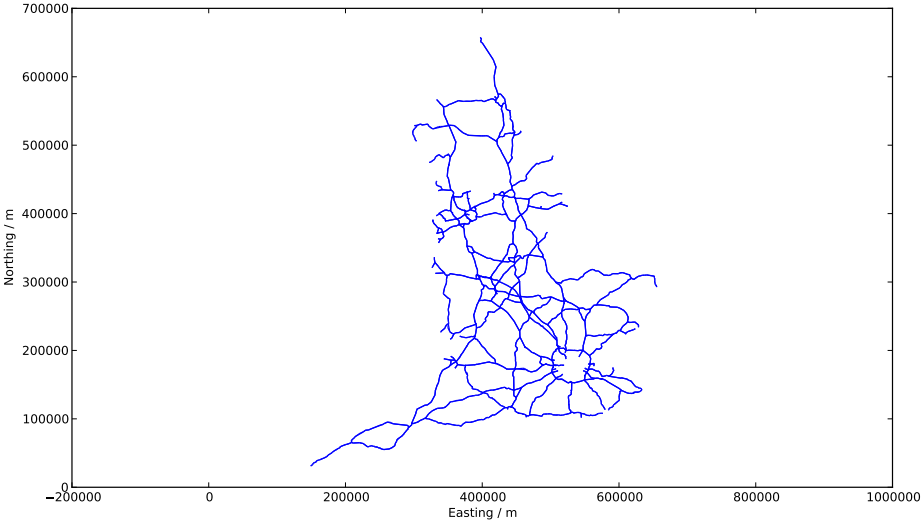
```
print(link_segments.values()[:5])

[((1.174898, 51.1075), (1.234404, 51.106983)),
 ((-1.519843, 53.73263), (-1.519193, 53.732506)),
 ((0.313035, 51.43044), (0.320784, 51.430897)),
 ((-2.023598, 52.672565), (-2.031822, 52.674973)),
 ((-1.204398, 52.617138), (-1.20728, 52.621113))]

from matplotlib.collections import LineCollection
lc = LineCollection(link_segments.values())
gca().add_collection(lc)
xlabel('Degrees East')
ylabel('Degrees North')
axis('equal')
savefig('wgs84-links.pdf')
```


56
55
54
53
52
51
50
Degrees North
−6
−4
−2
0
2
Degrees East

```
from osgeo import osr

wgs84 = osr.SpatialReference()
wgs84.ImportFromEPSG(4326)  # Longitude, Latitude (degrees)
bng = osr.SpatialReference()
bng.ImportFromEPSG(27700)   # OS grid Easting, Northing (metres)
trans = osr.CoordinateTransformation(wgs84, bng)
```

```
segment = link_segments.values()[0]
print(segment)
((1.174898, 51.1075), (1.234404, 51.106983))

print(trans.TransformPoints(segment))
[(622341.8470642003, 139045.90819568178, -43.13303761463612),
 (626508.8899003072, 139169.91645833637, -43.055007888004184)]

print([p[:2] for p in trans.TransformPoints(segment)])
[(622341.8470642003, 139045.90819568178),
 (626508.8899003072, 139169.91645833637)]
```

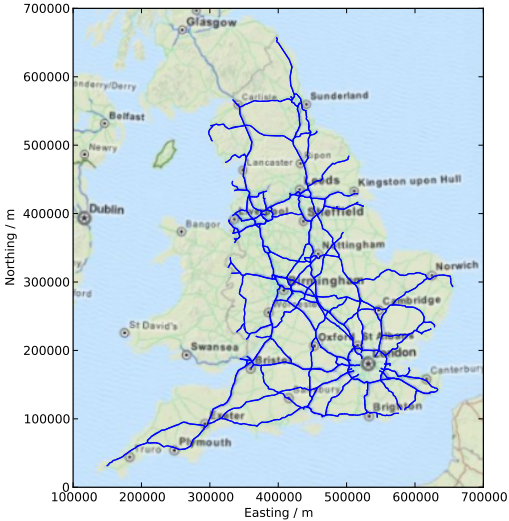
```
transformed_points = list(
    [x[:2] for x in trans.TransformPoints(segment)]
    for segment in link_segments.values()
)

lc = LineCollection(transformed_points)
gca().add_collection(lc)
xlabel('Easting / m')
ylabel('Northing / m')
axis('equal')
savefig('bng-links.pdf')
```


700000
600000
500000
400000
300000
200000
100000
0
Northing / m
−200000
0
200000
400000
600000
800000
1000000
Easting / m

https://github.com/rjw57/foldbeam/

```
foldbeam-render -p 27700 \
  --left 100000 --right 700000 \
  --top 700000 --bottom 0 \
  --height 900 -o bng-map.png
```

```
im = imread('bng-map.png')
imshow(im, extent=(1e5, 7e5, 0, 7e5))
lc = LineCollection(transformed_points)
gca().add_collection(lc)
xlabel('Easting / m')
ylabel('Northing / m')
axis('image')
savefig('bng-links-2.pdf')
```


Northing / m
Easting / m
700000
600000
500000
400000
300000
200000
100000
0
100000
200000
300000
400000
500000
600000
700000
Glasgow
Belfast
Dublin
Sunderland
Leeds
Kingston upon Hull
Sheffield
Nottingham
Norwich
Cambridge
Oxford
London
Swansea
Bristol
Brighton
Plymouth
Canterbury
St David's
Bangor
Lancaster

TrafficData

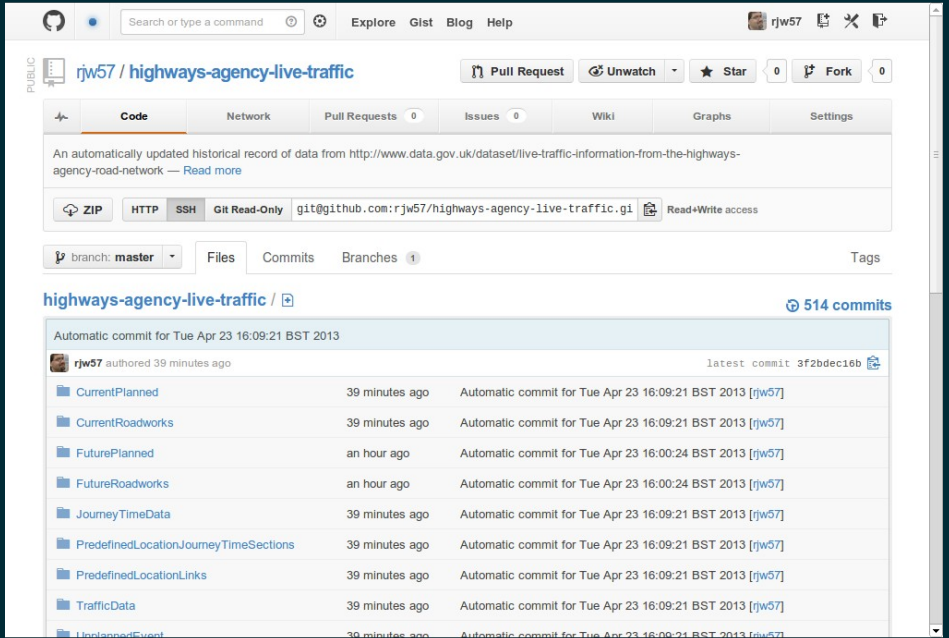
```
<elaboratedData id="GUID-58528643-101000101">
  <basicDataValue>
    <affectedLocation>
      <locationContainedInGroup>
        <predefinedLocationReference>
          Link101000101
        </predefinedLocationReference>
      </locationContainedInGroup>
    </affectedLocation>
    <averageVehicleSpeed>47.0</averageVehicleSpeed>
  </basicDataValue>
</elaboratedData>
```

```
link_data = {}
for datum in pub.elaboratedData:
    value = datum.basicDataValue
    location_id = value.affectedLocation.\
            locationContainedInGroup.\
            predefinedLocationReference.text

    # Get, or create, the appropriate link datum
    try:
        link_datum = link_data[location_id]
    except KeyError:
        link_datum = {}
        link_data[location_id] = link_datum

    link_datum['location_id'] = location_id
    link_datum['speed'] = float(value.averageVehicleSpeed.text)
```

```
link_data = {
    'Link1234': {
        'location_id': 'Link1234',
        'speed': 48.5
    },

    # etc, etc
}
```

```
# provides MATLAB-like linear algebra operations
import numpy as np

speeds = np.array([d['speed'] for d in link_data.values()])
hist(speeds)
```

Movie



Search or type a command

Explore Gist Blog Help

rjw57

PUBLIC

rjw57 / highways-agency-live-traffic

Pull Request Unwatch Star 0 Fork 0

Code Network Pull Requests 0 Issues 0 Wiki Graphs Settings

An automatically updated historical record of data from http://www.data.gov.uk/dataset/live-traffic-information-from-the-highways-agency-road-network — Read more

ZIP HTTP SSH Git Read-Only git@github.com:rjw57/highways-agency-live-traffic.gi Read+Write access

branch: master Files Commits Branches 1 Tags

highways-agency-live-traffic /

514 commits

Automatic commit for Tue Apr 23 16:09:21 BST 2013

rjw57 authored 39 minutes ago latest commit 3f2bdec16b

CurrentPlanned	39 minutes ago	Automatic commit for Tue Apr 23 16:09:21 BST 2013 [rjw57]
CurrentRoadworks	39 minutes ago	Automatic commit for Tue Apr 23 16:09:21 BST 2013 [rjw57]
FuturePlanned	an hour ago	Automatic commit for Tue Apr 23 16:00:24 BST 2013 [rjw57]
FutureRoadworks	an hour ago	Automatic commit for Tue Apr 23 16:00:24 BST 2013 [rjw57]
JourneyTimeData	39 minutes ago	Automatic commit for Tue Apr 23 16:09:21 BST 2013 [rjw57]
PredefinedLocationJourneyTimeSections	39 minutes ago	Automatic commit for Tue Apr 23 16:09:21 BST 2013 [rjw57]
PredefinedLocationLinks	39 minutes ago	Automatic commit for Tue Apr 23 16:09:21 BST 2013 [rjw57]
TrafficData	39 minutes ago	Automatic commit for Tue Apr 23 16:09:21 BST 2013 [rjw57]
UnplannedEvent	39 minutes ago	Automatic commit for Tue Apr 23 16:09:21 BST 2013 [rjw57]

```
im = imread('bng-map.png')
imshow(im, extent=(1e5, 7e5, 0, 7e5))
lc = LineCollection(transformed_points, cmap=cm.spectral)
lc.set_array(speed)
gca().add_collection(lc)
xlabel('Easting / m')
ylabel('Northing / m')
axis('image')
```

Movie

```
import ephem

def sun_altitude(date):
    """
    Return the altitude, in radians, of
    the Sun on a given date ant time as
    observed from London.
    """
    london = ephem.city('London')
    london.date = date
    return ephem.Sun(london).alt
```

Conclusions

- Python is a universal ‘glue’ language.
- The IPython web notebook is amazing.
- Plotting and numerical processing on a par with MATLAB.
- GPU acceleration is easily exposed as needed by PyOpenCL.
- Playing with data is fun.
- You too can make ‘mission control’.

http://gplus.to/richwareham rjw57@cam.ac.uk