

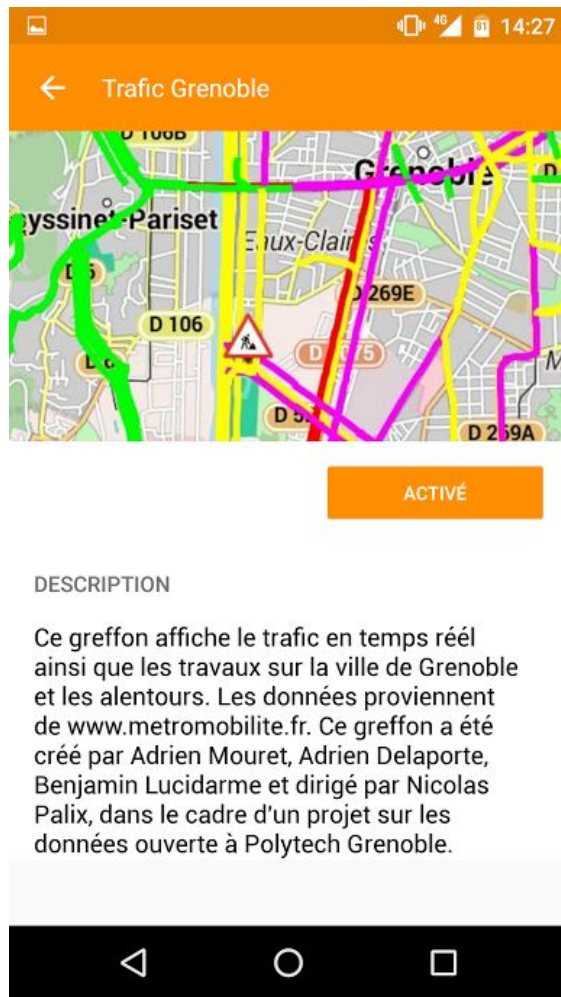
# Grenoblois Futé



DELAPORTE Adrien - LUCIDARME Benjamin -  
MOURET Adrien - RICM4

# Sommaire

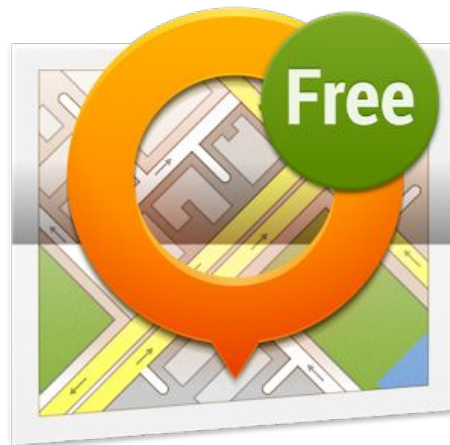
- 1°/ Présentation du projet
- 2°/ Création du Plugin
- 3°/ Réalisation des Parsers
  - a°/ JSON / GEOJSON
  - b°/ XML
- 3°/ Affichage sur OsmAnd
- 4°/ Problèmes rencontrés
- 5°/ Perspectives du projet



# Présentation du projet

- Contexte & contraintes :

- OsmAnd
- Métromobilité











- Buts :

- Trafic en temps réel
- Indication des travaux



# Présentation du projet

- Quantification du projet existant :
  - 16 langages de programmation différents
  - 695 956 lignes de code

Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
 XML	346,647	4,491	1.3%	15,972	367,110	 45.8%
 Java	224,981	19,348	7.9%	30,878	275,207	 34.4%
 C++	108,081	5,952	5.2%	19,119	133,152	 16.6%
 Python	5,045	1,623	24.3%	1,215	7,883	 1.0%

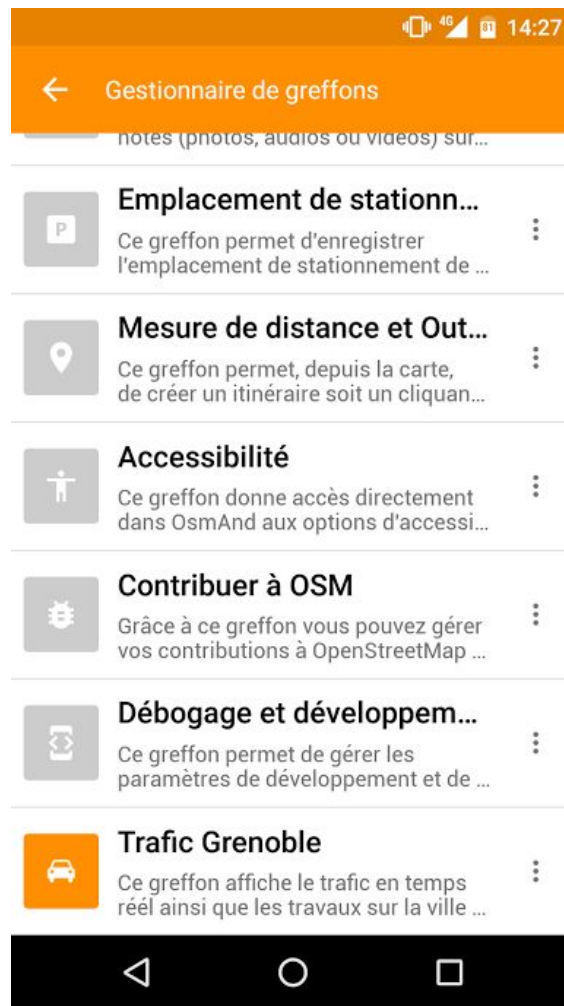
# Gestion du projet

Deux parties majeures :

- Gestion du Plugin :
  - Création du Plugin
  - Utilisation des données dans le Plugin
- Gestion des données ouvertes :
  - Trafics : JSON / GEOJSON
  - Travaux : XML

# Création du Plugin

- Affichage dans le Gestionnaire de Greffons
  - Class osmandPlugin.java - initPlugin(...)
  - getName(), getLogoResourceId(), getDescription()
- Methodes essentielles :
  - registerLayers(...)
  - updateLayers(...)



# Parser JSON / GEOJSON

A screenshot of a JSON viewer interface. The window has a title bar with 'Viewer' and 'Text' tabs. The main content area shows a tree view of a JSON object. The root is a JSON object with a 'features' array. The first feature has properties 'CODE', 'NIVEAU', and 'NSV\_ID', and a 'geometry' object with a 'MultiLineString' type and a 'coordinates' array. The 'coordinates' array contains three sub-arrays, each representing a line of coordinates. The first line has two points: (5.67619, 45.21778).

```
{
  "eacute": "\u00c5",
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "CODE": "T002",
        "NIVEAU": 1,
        "NSV_ID": 0
      },
      "geometry": {
        "type": "MultiLineString",
        "coordinates": [
          [
            [
              [
                0: 5.67619,
                1: 45.21778
              ]
            ]
          ]
        ]
      }
    }
  ]
}
```

Donn\u00e9es statiques

A screenshot of a JSON viewer interface. The window has a title bar with 'Viewer' and 'Text' tabs. The main content area shows a tree view of a JSON object. The root is a JSON object with a 'date' property and a 'features' array. The first feature has properties 'CODE' and 'NSV\_ID'. The remaining features are represented by empty JSON objects in an array, indicating a dynamic or large set of data.

```
{
  "eacute": "\u00c5",
  "date": "05/04/2016 13:16:30",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "CODE": "N1_999",
        "NSV_ID": 1
      }
    },
    {},
    {},
    {},
    {},
    {},
    {},
    {},
    {}
  ]
}
```

Donn\u00e9es dynamiques

# Parser XML

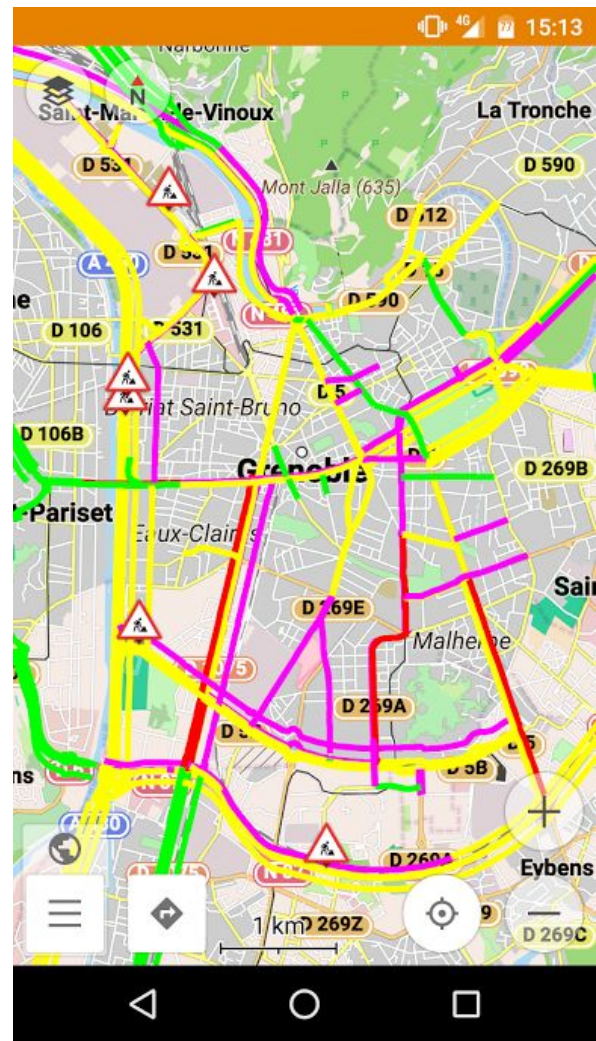
```
▼<Evts>
  ▼<Evt>
    <Code>GAM_EVT_41402</Code>
    <Nat>CHANTIER</Nat>
    <SsNat>CHANTIER</SsNat>
    <DDebut>04/01/2016 09:00:00</DDebut>
    <DFin>30/04/2016 18:00:00</DFin>
    <HDebut>00:00:00</HDebut>
    <HFin>00:00:00</HFin>
    <WE>2</WE>
    <Com>GRENOBLE</Com>
    <Loc>CHANTIER: Avenue des Martyrs</Loc>
    <Imp>50</Imp>
    <Lat>45.2003</Lat>
    <Lon>5.707</Lon>
  ▼<Comment>
    ▼<![CDATA[
      CHANTIER: Avenue des Martyrs<BR> Du 04/01/2016 09:00<BR> au
    ]]>
    </Comment>
  </Evt>
```

- Utilisation des bibliothèques DOM et SAX
- Implémentation des fonctions pour récupérer toutes les données



# Affichage sur OsmAnd

- Intégration des analyseurs syntaxiques
- Création du layer de trafic
- Appel du layer par le greffon de trafic



# Problèmes rencontrés

- Création du module
- GIT
- Intégration des Parsers sous Android Studio
- Affichage des tronçons

# Conclusion

- Technologies utilisées :
  - Android Studio
    - Java, Gradle
  - XML
    - Parser, Librairies DOM et SAX
  - JSON, GEOJSON
    - Parser, Librairie JSON.org
  - JOSM
    - Affichage de fichiers GPX

➔ 885 lignes de codes : 82 lignes de XML & 803 lignes de Java

➔ 9 classes concernées

# Conclusion

- Plugin obtenu :
  - Affichage du trafic en temps réel
  - Indication des travaux en temps réel
  
- Suites possibles du projet :
  - Extension à d'autres villes
  - Routage
  - Réglages supplémentaires (infos travaux, optimisation de l'affichage ...)
  - Mode hors ligne