
IncidentsTransports Documentation

Release 2.0

Olivier Girardot

October 14, 2011

CONTENTS

Contents:

INTRODUCTION

1.1 History

This project began as a small idea to create a platform allowing people to interact and share information about incidents in Public Transportations.

1.2 Why

API

2.1 Cities

2.2 Lines

2.3 Incidents

2.4 Votes

Votes are linked to incidents, therefore you can only interact with them in relation to an incident, you can :

- get the current number of votes for an incident;
- vote for an incident.

Votes are available using three types of actions :

- Plus : meaning you confirm there is an incident still going on ;
- Minus : meaning you infirm there is an incident or ever was;
- End : meaning you know that there was an incident which is now over.

The full access point as a Django URL is defined as such ::

```
r'api/latest/incident.(?P<emitter_format>[a-z]+)/vote/(?P<incident_id>[\d]+)/(?P<action>[a-z])', vot
```

Where :

- the emitter_format : can be *xml*, *json*;
- the incident_id : is pretty much self explanatory;
- and the action : as defined earlier can take the values *plus*, *minus* or *end*.

2.4.1 Examples

Get the number of vote

So for example to get the number of vote for a given *incident_id* (id=1) ::

GET : `http://www.incidents-transport.com/api/incident.json/vote/1/end`

The answer will be ::

```
{  
  "number": 0  
}
```

Another example would be to get the number of *minus* votes for the same *incident_id* ::

GET : `http://www.incidents-transport.com/api/incident.json/vote/1/minus`

The answer will be ::

```
{  
  "number": 6  
}
```

Vote for an Incident

DEVELOPMENT

3.1 How to get involved

3.2 Technology

3.3 Issues

3.4 Contact

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*