

FR:Xapi

De OpenStreetMap Wiki

Langues disponibles

Aide

• Deutsch • English • **français** • 日本語 • português do Brasil

L'API étendue d'OSM (ou xapi, prononcer "zapi") est une API en lecture seule, basée sur une version modifiée de l'API standard, qui offre des capacités avancées de recherche et de récupération de données. Elle implémente la requête standard de carte, ainsi que plusieurs autres façons d'obtenir des données OSM en indiquant les tags recherchés. Xapi utilise une interface de type REST avec une touche de XPath.

Xapi gère uniquement les éléments actuels et ne retourne aucun élément historique ou effacé. La base de données utilisée est un miroir de la base de données principale d'OSM et elle est mise à jour par les dumps différentiels créés toutes les minutes. Les données ont en général moins de 10 minutes de retard par rapport à celles de la base de données principale.

Tous les résultats sont présentés dans le même format que le protocole standard, avec en plus quelques extensions de *namespaces*. Les requêtes de tags utilisent une syntaxe de type XPath pour spécifier les termes recherchés. L'attribut de visibilité est exclu de la réponse car il renvoie toujours une valeur vraie.

Sommaire

- 1 Serveurs
 - 1.1 Historiques
 - 1.2 API Overpass
 - 1.3 jXAPI
- 2 Map Query
 - 2.1 Exemple
 - 2.2 Notes
- 3 Tag Query
 - 3.1 Nœuds
 - 3.2 Chemins
 - 3.3 Relations
 - 3.4 Tous les Éléments
- 4 Predicats
 - 4.1 Prédicats de Tag
 - 4.2 Prédicats BBox
 - 4.3 Prédicats éléments enfants
 - 4.4 Caractère spéciaux
- 5 Tags
- 6 Utilisation

- 7 Limites
- 8 Extensions proposées
- 9 Alternatives
- 10 Voir également

Serveurs

Historiques

L'API XAPI historique (originelle) a été créée en utilisant le langage GT.M (auparavant MUMPS) par l'utilisateur User:80n. Il y avait plusieurs instances du service xapi qui tournaient sur plusieurs serveurs. Chaque serveur pouvait être attaqué directement ou par une redirection pointant la requête vers la machine la plus adaptée.

Tous les serveurs originels ont été démantelés et ne fonctionnent plus du tout.

API Overpass

L'Overpass API dispose maintenant d'une couche de compatibilité XAPI. (état: **La serveur a une faible charge.**). Le serveur gère la plupart des requêtes XAPI courantes ainsi que son propre langage de requête étendue. Vous trouverez plus d'informations dans la section couche de compatibilité XAPI.

jXAPI

En janvier 2011, Ian Dees a créé une nouvelle version Java de XAPI basée sur postGIS. Consultez son journal pour plus de détails (<http://www.openstreetmap.org/user/iandeas/diary/12916>) . Ce service est maintenant accessible sur une nouvelle machine par l'URL: <http://jxapi.openstreetmap.org/xapi/api/0.6/...>

Vous pouvez également l'essayer via "uixapi" (<http://openstreetmap.us/uixapi/xapi.html>) .

Veuillez noter que les requêtes bbox ont au maximum une emprise de 10 degrés carrés.

- <http://jxapi.openstreetmap.org/xapi/api/0.6/...> (http://jxapi.openstreetmap.org/xapi/api/0.6/*%5Bname=Sylt%5D) JXAPI par Ian Dees (état: **Fermé pour maintenance**, page de statistiques (<http://jxapi.openstreetmap.org/xapi/admin/stats>))
- <http://open.mapquestapi.com/xapi/api/0.6/...> (http://open.mapquestapi.com/xapi/api/0.6/*%5Bname=Sylt%5D) fournie par Open Mapquest (état: **OK, mais les relations ne sont pas toutes gérées**, page de statistiques (<http://open.mapquestapi.com/xapi/admin/stats>)). (Consultez service XAPI MapQuest (<http://developer.mapquest.com/web/products/open/xapi>))
- <http://jxapi.osm.rambler.ru/xapi/api/0.6/...> (http://jxapi.osm.rambler.ru/xapi/api/0.6/*%5Bname=Sylt%5D) fournie par Rambler (état: **OK, mais lent et les**

relations ne sont pas toutes gérées, page de statistiques
(<http://jxapi.osm.rambler.ru/admin/stats>)).

L'intérêt de jXAPI réside dans le fait que Java et PostGis sont moins exotiques que GT.M (dont peu de développeurs ont entendu parlé). Néanmoins, l'installation et la configuration sont quand même d'un bon niveau technique. Consultez le blog de paulnorman sur l'installation de jXAPI (<http://www.paulnorman.ca/blog/2011/11/loading-a-pgsnapshot-schema-with-a-planet-take-2/>)

Map Query

La Map query est identique à la map query de l'API principale et retourne:

- Tous les noeuds qui sont dans une bounding box donnée et toutes relations qui les référencent.
- Tous les chemins (ways) qui référencent au moins un noeud se trouvant dans une bounding box donnée, toutes relations qui les référencent [the ways] et tous les noeuds se trouvant à l'extérieur de la bounding box que les chemins référencent.

GET /api/0.6/map?bbox=gauche,bas,droite,haut

où:

- gauche est la longitude du coté gauche (le plus à l'ouest) de la bounding box, ou minlon.
- bas est la latitude du coté inférieur (le plus au sud) de la bounding box, ou minlat.
- droite est la longitude du coté droit (le plus à l'est) de la bounding box, ou maxlon.
- haut est la latitude du coté supérieur (le plus au nord) de la bounding box, ou maxlat.

Exemple

<http://www.informationfreeway.org/api/0.6/map?bbox=11.54,48.14,11.543,48.145>

Notes

- A la différence de l'API principale qui limite la question à un carré de 0.25 degrés, XAPI permet des requêtes plus étendues, jusqu'à 10,000,000 éléments (ce qui représente moins d'un tiers de la Californie) ou jusqu'à un carré de 100 degrés, ce qui est encore plus petit.
- La map query est fonctionnellement identique à la tag query /api/0.6 /*[bbox=gauche,bas,droite,haut]
- Il ne doit pas y avoir d'espaces dans la chaîne de caractères qui délimite la bbox.

Tag Query

Xapi implémente des requêtes pour les 3 sortes d'elements; noeuds, chemins et relations.

Nœuds

Une requête node retournera un document XML contenant des nœuds et les tags associés. L'URL pour une requête node prend la forme suivante:

[http://www.informationfreeway.org/api/0.6/node\[...\]](http://www.informationfreeway.org/api/0.6/node[...])

Cela retournera une réponse standard de l'API OSM contenant les nœuds et les tags. Par exemple:

```
<?xml version='1.0' standalone='no'?>
<osm version='0.6' generator='xapi: OSM Extended API'
  xmlns:xapi='http://www.informationfreeway.org/xapi/0.6'
  xapi:uri='/api/0.6/node[amenity=hospital]'
  xapi:planetDate='200803150826'
  xapi:copyright='2008 OpenStreetMap contributors'
  xapi:instance='zappy2'>
  <node id='672180' lat='48.2111685091189' lon='16.3035366605548' timestamp='200803150826'>
    <tag k='amenity' v='hospital' />
    <tag k='name' v='Wilhelminenspital' />
  </node>
  <node id='3596186' lat='53.4633699598014' lon='-2.22667910006381' timestamp='200803150826'>
    <tag k='amenity' v='hospital' />
    <tag k='name' v='Manchester Royal Infirmary' />
  </node>
  ...
</osm>
```

Chemins

L' URL pour une requête way a la forme suivante:

[http://www.informationfreeway.org/api/0.6/way\[...\]](http://www.informationfreeway.org/api/0.6/way[...])

Cela retourne un document XML contenant les chemins qui correspondent aux termes de la recherche. Pour chaque réponse satisfaisant les critères, les nœuds constituant le chemin sont également retournés. Un exemple de réponse pour la requête way ressemblera à ceci:

```
<?xml version='1.0' standalone='no'?>
<osm version='0.6' generator='xapi: OSM Extended API'
  xmlns:xapi='http://www.informationfreeway.org/xapi/0.6'
  xapi:uri='/api/0.6/way[landuse=residential]'
  xapi:planetDate='200803150826'
  xapi:copyright='2008 OpenStreetMap contributors'
  xapi:instance='zappy2'>
  <node id='218963' lat='52.5611324692581' lon='-1.79024812573334' timestamp='200803150826'>
  </node>
  <node id='331193' lat='53.7091237972264' lon='-1.50282510180841' timestamp='200803150826'>
    <tag k='highway' v='traffic_signals' />
  </node>
  ...
</osm>
```

```

    <tag k='source' v='Yahoo' />
  </node>
  ...
  <way id='4958218' timestamp='2007-07-25T01:55:35+01:00' version='3' changeset=
    <nd ref='218963' />
    <nd ref='331193' />
    ...
    <tag k='landuse' v='residential' />
    <tag k='source' v='landsat' />
  </way>
</osm>

```

Relations

L'URL pour une requête de type relation a la forme suivante:

[http://www.informationfreeway.org/api/0.6/relation\[...\]](http://www.informationfreeway.org/api/0.6/relation[...])

Elle retourne un document XML contenant les relations qui satisfont aux critères de recherche. Pour chaque relation, les nœuds et les chemins référencés sont également retournés. Un exemple de réponse pour une requête relation ressemble à ce qui suit:

```

<?xml version='1.0' standalone='no'?>
<osm version='0.6' generator='xapi: OSM Extended API'
  xmlns:xapi='http://www.informationfreeway.org/xapi/0.6'
  xapi:uri='/api/0.6/way[landuse=residential]'
  xapi:planetDate='200803150826'
  xapi:copyright='2008 OpenStreetMap contributors'
  xapi:instance='zappy2'>
  <node ...
  <way ...
  <relation id='2670' timestamp='2007-10-25T03:05:34Z' version='32' changeset=
    <member type='way' ref='3992472' role='' />
    <member type='way' ref='3992524' role='' />
    <member type='way' ref='4253050' role='' />
    <member type='way' ref='4253053' role='' />
    <member type='way' ref='4266813' role='' />
    <member type='way' ref='10285106' role='' />
    <tag k='name' v='Fonnereau Way' />
    <tag k='network' v='Ipswich footpaths' />
    <tag k='type' v='route' />
  </relation>
</osm>

```

Tous les Éléments

Tous les éléments (nœuds, chemins et relations) qui satisfont aux prédicats du filtre de recherche peuvent être recherchés en utilisant l'URL suivante:

[http://www.informationfreeway.org/api/0.6/*\[...\]](http://www.informationfreeway.org/api/0.6/*[...])

Cette requête renvoie un document XML contenant les nœuds, les chemins et les relations qui satisfont les critères de recherche. Pour chaque chemin satisfaisant aux critères, les nœuds et les relations référencés sont également renvoyés. Un exemple de réponse pour une requête de ce type ressemble à ce qui suit:

```
<?xml version='1.0' standalone='no'?>
<osm version='0.6' generator='xapi: OSM Extended API'
  xmlns:xapi='http://www.informationfreeway.org/xapi/0.6'
  xapi:uri='/api/0.6/*[amenity=hotel]'
  xapi:planetDate='200803150826'
  xapi:copyright='2008 OpenStreetMap contributors'
  xapi:instance='zappy2'>
  <node id='218963' lat='52.5611324692581' lon='-1.79024812573334' timestamp='
  </node>
  <node id='331193' lat='53.7091237972264' lon='-1.50282510180841' timestamp='
    <tag k='amenity' v='hotel' />
  </node>
  ...
  <way id='4958218' timestamp='2007-07-25T01:55:35+01:00' version='1' changeset=
    <nd ref='218963' />
    <nd ref='331193' />
    ...
    <tag k='amenity' v='hotel' />
    <tag k='building' v='hotel' />
  </way>
  <relation id='123456' timestamp='2007-10-25T03:05:34Z' version='32' changeset=
    <member type='node' ref='331193' role='' />
    <member type='node' ref='331194' role='' />
    ...
    <tag k='amenity' v='hotel' />
    <tag k='operator' v='Premier Inns' />
    <tag k='type' v='operators' />
  </relation>
</osm>
```

Predicats

Chaque requête API peut être suffixée par des *predicats* de sélection qui déterminent quel élément sélectionner. Par exemple [amenity=hospital] sélectionnera tous les éléments qui possèdent un tag amenity avec la valeur hospital. L'URL complète pour sélectionner tous les noeuds taggés Hospital sera:

[http://www.informationfreeway.org/api/0.6/node\[amenity=hospital\]](http://www.informationfreeway.org/api/0.6/node[amenity=hospital])

Une requête peut être suffixée avec de multiples prédicats, chacun d'entre eux contraignant le résultat (actuellement limité à un prédicat de tag et prédicat de bbox géographique). Par exemple:

[http://www.informationfreeway.org/api/0.6/node\[amenity=hospital\]\[bbox=-6,50,2,61\]](http://www.informationfreeway.org/api/0.6/node[amenity=hospital][bbox=-6,50,2,61])

Cette requête sélectionnera tous les noeuds possédant le tag amenity=hospital se

trouvant dans la bounding box englobant l'Angleterre, le Pays de Galles et l'Ecosse.

Prédicats de Tag

Un prédicat de sélection doit être de la forme `[key=value]`. Par exemple:

`node[amenity=hospital]` sélectionnera tous les noeuds qui possèdent la clé `amenity` et la valeur `hospital`, c'est à dire

```
<tag k="amenity" v="hospital"/>
```

Un opérateur d'union peut aussi être utilisé pour sélectionner des valeurs multiples. Par exemple pour sélectionner toutes les routes principales:

```
■ way[highway=motorway|motorway_link|trunk|primary]
```

L'opérateur union peut aussi être utilisé avec la clé. Par exemple pour sélectionner tous les parcours de golf:

```
■ node[amenity|leisure=golf_course]
```

Un joker (étoile) peut être utilisée pour la valeur (mais pas pour la clef), permettant d'écrire des prédicats comme les exemples suivants:

```
■ way[construction=*]  
■ way[highway=*]
```

Les jokers de clef étaient auparavant supportés mais ils ne sont plus maintenus.

Note: l'URL nécessite d'être encodée en UTF-8 en cas d'utilisation de caractères non-ascii.

Prédicats BBox

Une autre forme de prédicat est la pseudo clef `bbox`. Un prédicat de sélection de la forme `[bbox=gauche,bas,droite,haut]` définit un rectangle enveloppant qui est utilisé pour limiter l'étendue de la réponse. Seuls les éléments présents dans cette enveloppe feront partie de la réponse.

A la différence de l'API standard d'OSM, le prédicat `bbox` n'est pas limité en taille, tant qu'il est utilisé sans prédicat de tag au quel cas il est limité à un carré de 100 degrés.

L'étendue par défaut, si le prédicat `bbox` n'est pas spécifié, est la planète entière. Cela revient à écrire `[bbox=-180,-90,180,90]`.

Prédicats éléments enfants

Des objets peuvent être sélectionnés sur la présence ou l'absence d'éléments enfants. Par exemple, les chemins peuvent être sélectionnés sur le fait qu'ils n'aient pas un tag

donné ou qu'ils ne possèdent pas de nœuds. Cette opération est permise par l'utilisation d'un élément test de type XPATH. Plus précisément voici ce qui est implémenté:

- `/api/0.6/way[nd]` - sélectionne les chemins qui ont au moins un nœud
- `/api/0.6/way[tag]` - sélectionne des chemins qui ont au moins un tag
- `/api/0.6/way[not(nd)]` - sélectionne uniquement des chemins qui n'ont pas de nœuds
- `/api/0.6/way[not(tag)]` - sélectionne uniquement des chemins qui n'ont pas de tag
- `/api/0.6/node[way]` - sélectionne des noeuds appartenant à au moins un chemin
- `/api/0.6/node[not(way)]` - sélectionne des nœuds qui n'appartiennent à aucun chemin
- `/api/0.6/node[not(tag)]` - sélectionne les nœuds qui n'ont aucun tag
- `/api/0.6/relation[node]` - sélectionne les relations qui ont au moins un nœuds
- `/api/0.6/relation[way]` - sélectionne les relations qui ont au moins un chemin
- `/api/0.6/relation[relation]` - sélectionne les relations qui ont au moins un membre relation
- `/api/0.6/relation[not(node)]` - sélectionne les relations qui n'ont aucun nœuds
- `/api/0.6/relation[not(way)]` - sélectionne les relations qui n'ont aucun chemin
- `/api/0.6/relation[not(relation)]` - sélectionne les relations qui n'ont aucun membre relation

Notes:

1. En cas d'utilisation de wget, les parenthèses doivent être échappées, de sorte que pour un prédicat négatif on utilise `[not\ (way\)]` plutôt que `[not(way)]`.
2. Le schema xml d' OSM n'a pas réellement de ways comme éléments enfants. Cependant il y a logiquement, une relation de plusieurs à plusieurs entre les nœuds et les chemins et xapi implémente cette relation et permet de sélectionner des nœuds selon qu'ils appartiennent ou non à un chemin. Naturellement, le fichier xml produit est le même que normalement et n'inclus pas de way dans un élément node.

Caractère spéciaux

Les valeurs sans prédicat peuvent être exclu en ajoutant un anti-slash. Par exemple pour récupérer les éléments avec la valeur du tag contenant le caractère | comme `[route=46|46A]` vous devrez utiliser `[route=46\|46A]`.

Les caractères suivant nécessite l'anti-slash: `[] * / = () \` et espace

les espaces contenu dans la valeur du tag n'ont normalement pas besoin d'être précédé de l'anti-slash mais cela peut dépendre du client. Dans ce cas, il faut remplacer les espaces par la valeur `%20`. donc `[route=46 46A]` deviendra `[route=46%2046A]`.

Tags

La méthode principale pour interroger xapi est d'utiliser le couple tag/valeur dans un prédicat. En plus de tous les tags normaux que l'on peut trouver, les attributs suivants peuvent également être interrogés en utilisant `/api/0.6/*[@attribute=value]`

- @user - le nom d'utilisateur de la dernière personne à avoir modifié l'élément
- @uid - l'identifiant de la dernière personne à avoir modifié l'élément
- @changeset - le changeset dans lequel l'élément a été modifié

Ces requêtes fonctionnent mieux sur le serveur XAPI normal car le serveur jxapi les fait fonctionner très lentement.

Utilisation

Les URLs Xapi peuvent être utilisés à partir de n'importe quel navigateur web, cependant une simple requête peut retourner plusieurs Mb de données ce qui peut dépasser les capacités de certains navigateurs. Il est recommandé qu'un outil comme wget ou curl soit utilisé avec une sortie redirigée vers un fichier. Par exemple la commande suivante utilise wget pour sélectionner tous les hôpitaux et stocker le résultat dans un fichier appelé data.osm. A noter que wget a une valeur de timeout de 15 minutes ce qui peut parfois être insuffisant pour certaines requêtes. Assurez-vous d'utiliser l'option --timeout=0 pour gérer ce cas.

```
wget --timeout=0 http://www.informationfreeway.org/api/0.6/node[amenity=hospital
```

Note : si vous utilisez curl, il effectuera les redirections automatiquement et vous devrez interroger le serveur hôte directement ou utiliser l'option "--location". Vous devrez également utiliser l'option "--globoff", sinon curl tentera d'interpréter les crochets ("[" , "]") dans la ligne de commande:

```
curl --location --globoff "http://www.informationfreeway.org/api/0.6/node[amenit
```

Si vous utilisez un script pour récupérer les données depuis Xapi, il peut être utile de donner un User-Agent parlant. Cela nous permettra de mieux comprendre comment Xapi est utilisé.

Limites

- **Prédicats.** Actuellement, chaque requête est limitée à un prédicat de tag et un prédicat de bbox. A l'avenir, nous espérons implémenter des conditions plus élaborées. Les bbox multiples seraient très intéressantes pour les applications comme GpsMid - elle implémente déjà la récupération de données à travers xapi dans OsmGpsMidt pour les régions uniques ce qui est très pratique.
- **Grammaire de requête.** La grammaire de requête n'est pas formellement définie et elle n'est pas vérifiée par un analyseur syntaxique de sorte que les requêtes mal formatées peuvent retourner n'importe quoi.

Extensions proposées

- "Recherche par région": une fonctionnalité pour limiter les requêtes non plus à une

bouding box mais à des regions identifiées (ex: Pays, Comtés, continents). Proposé en janvier 2011.

- export JSON: Cette fonctionnalité est présente dans xappy.js? Elle peut être utilisée en indiquant "application/json" dans l'en-tête HTTP content-type. Quelques informations sur le format: Xappy.js#JSON output

Alternatives

- Osmosis Tag Filter.
- Taginfo/API
- API Overpass

Voir également

- <http://open.mapquestapi.com/xapi/>
- API v0.6
- Le code source Xapi (<http://xapi.openstreetmap.org/>) utilise GT.M une base de données à haute performance sans schéma. (jusqu'à janvier 2011)
- xappy.js une réimplémentation en Javascript

Récupérée de « <http://wiki.openstreetmap.org/w/index.php?title=FR:Xapi&oldid=906151> »

Catégories : OSM API | Technical | Data store

- Dernière modification de cette page le 21 mai 2013 à 03:47.
- Content is available under Creative Commons Attribution-ShareAlike 2.0 license.