

# Format des données ouvertes TEMPS-RÉEL



Tous droits réservés © Ville de Montréal

Tout transfert ou reproduction de ce document, exploitation ou communication de son contenu sont interdits, sauf consentement écrit préalable.

Tableau 1 - Historique des révisions

<b>Historique des révisions</b>			
<b>No</b>	<b>Date (aaaa-mm-jj)</b>	<b>Auteur</b>	<b>Commentaires</b>
01	2017-05-12	Julien Ciesla	Version initiale
01A	2017-05-17	Julien Ciesla	Mise à jour du format de l'heure dans les données d'horodatage
01B	2017-10-18	Julien Ciesla	Correction du fonctionnement de l'expiration « ExpireUtc ». Remplacement du caractère d'abonnement de topic « * » à « # ».
01C	2017-10-24	Julien Ciesla	Clarification de l'usage des majuscules/minuscules dans les valeurs de topics et dans les nœuds JSON.
01D	2017-10-25	Julien Ciesla	Clarification de l'usage des majuscules/minuscules dans les nœuds JSON (Bis), pour les unités de mesure et les acronymes.

Tableau 2 - Références documentaires

<b>Références documentaires</b>	
<b>No</b>	<b>Description</b>

Tableau 3 - Termes et acronymes

<b>Termes et acronymes</b>	
<b>No</b>	<b>Description</b>
ODTF1	Open Data Topic Format Version 1
ODNF1	Open Data Node Format Version 1
CGMU	Centre de Gestion de la Mobilité Urbaine
MQTT	Message Queuing Telemetry Transport

## 1 Table des matières

1	TABLE DES MATIÈRES .....	3
2	LISTE DES FIGURES.....	4
3	LISTE DES TABLEAUX.....	4
4	AVANT-PROPOS .....	5
5	PORTÉE DU DOCUMENT .....	5
6	LES RUBRIQUES .....	5
6.1	Abonnement à des rubriques.....	6
7	ARBORESCENCE DES DONNÉES .....	6
7.1	Format .....	7
7.2	Country .....	7
7.3	Province .....	7
7.4	City.....	7
7.5	Theme .....	7
7.6	System .....	7
7.7	Subsystem.....	7
7.8	Class.....	7
7.9	Instance .....	8
7.10	Node .....	8
8	LE FORMAT DES NŒUDS .....	8
8.1	Format .....	9
8.1.1	Desc .....	9
8.1.2	CreateUtc .....	9
8.1.3	ExpireUtc.....	9

8.1.4	Unit.....	9
8.1.5	Status.....	9
8.1.6	Value .....	9

## 2 Liste des figures

Figure 1 - La rubrique et le séparateur .....	6
Figure 2 - Arborescence des données .....	6
Figure 3 - Format ODNF1, nœud d'une valeur simple.....	8
Figure 4 - Nœud avec donnée personnalisée.....	10
Figure 5 - Nœud complexe comportant une liste de nœuds enfants .....	10

## 3 Liste des tableaux

Tableau 1 - Historique des révisions.....	2
Tableau 2 - Références documentaires.....	2
Tableau 3 - Termes et acronymes .....	2

## 4 Avant-propos

Dans le contexte de la ville intelligente, la ville de Montréal souhaite publier ses données à tous afin d'initier un levier bénéfique à l'ensemble de la communauté dans différents secteurs. Les activités qui suivent découlent d'initiatives proposées par le centre de gestion de la mobilité urbaine (CGMU).

Extrait du site <http://donnees.ville.montreal.qc.ca/>: En ouvrant ses données à tous, la Ville de Montréal permet qu'elles soient réutilisées à différentes fins, incluant des fins commerciales. Les résultats de cette réutilisation peuvent ensuite être partagés dans la communauté, ce qui crée un effet démultiplicateur. Les données libérées et réutilisées génèrent ainsi des bénéfices à la fois dans les sphères économiques, culturelles, sociales et technologiques.

## 5 Portée du document

Dans un contexte de publication de données temps-réel provenant de différents systèmes de la ville intelligente, ce document décrit une arborescence standardisée permettant de classer les données, ainsi que le standard dans lesquelles elles sont formatées. Puisqu'il n'y a pas de dialogue envisagé entre les producteurs de données (ex : la ville) et les consommateurs (les citoyens), un patron de communication « publish/subscribe » sera utilisé. En effet, ce patron définit un mécanisme de publication et d'abonnement de messages dans lequel les émetteurs ne destinent pas les messages à des destinataires. À la place, une rubrique est associée aux messages émis sans savoir s'il y a des destinataires. De la même manière, les destinataires souscrivent aux rubriques qui les intéressent, et ne reçoivent que les messages correspondant, sans savoir qui sont les émetteurs.

Le format proposé dans ce document facilitera l'intégration entre les différents intervenant, autant pour la production des données par la Ville, que pour leur utilisation par les citoyens. Le format est conçu pour être le plus évolutif que possible, lui permettant d'évoluer à des notions qui ne sont pas connues pour le moment.

## 6 Les rubriques

Une rubrique (en anglais « topic ») est une chaîne de caractères qui permet au citoyen de filtrer les données qu'il désire recevoir. Une rubrique consiste à un ou plusieurs niveaux, dont chacun est divisé des autres par un caractère spécial appelé le séparateur. Dans le document qui suit, le séparateur sera identifié comme le caractère barre oblique (« slash » en anglais), se référant au séparateur utilisé par plusieurs protocoles communs, tel que MQTT. Il est important de mentionner que le séparateur peut différer selon la technologie utilisée, et qu'il n'est d'aucune valeur dans la spécification ci-jointe.

Les niveaux des rubriques sont écrits en utilisant le jeu de caractères suivants :

- Les chiffres de 0 à 9
- Les lettres minuscules « a » à « z »
- Le trait d'union ( - )

Les espaces et tout autre caractère non spécifié dans la liste ci-dessus ne sont pas permis, permettant ainsi de faciliter le codage et le décodage des messages sur tous types de plateformes. Voici un exemple de rubrique :

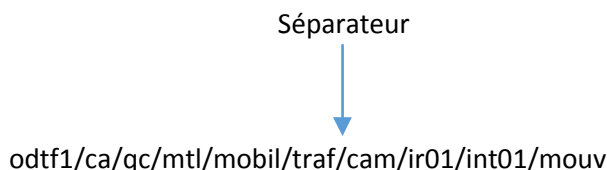


Figure 1 - La rubrique et le séparateur

## 6.1 Abonnement à des rubriques

Un citoyen peut s'abonner à une ou plusieurs rubriques spécifiques. Lors de l'abonnement, il est aussi possible d'utiliser des caractères spéciaux pour s'abonner à un ensemble de rubriques. Les caractères spéciaux diffèrent cependant de la technologie utilisée, de la même manière que le séparateur. Si le protocole MQTT est utilisé, 2 caractères spéciaux sont disponibles :

- Le caractère plus ( + ) permet de s'abonner à l'ensemble des valeurs d'un niveau donné.
- Le caractère dièse ( # ) permet de s'abonner à l'ensemble des valeurs d'un niveau et des sous-niveaux subséquents. Le caractère étoile n'est pas recommandé et devrait être évité dans la mesure du possible.

Voici un exemple pour récupérer tous les messages de détection de mouvement des caméras de trafic pour l'ensemble des villes du Québec :

odtf1/ca/qc/+ /mobil/traf/cam/+ /+ /mouv

Note : Il n'est pas dans l'intérêt du citoyen de s'abonner à trop grand nombre de données, car le logiciel les recevant ne sera pas en mesure de les traiter, et dans un tel cas, les données accumulées sur une trop longue période de temps seront perdues. Il est donc important de maîtriser son besoin, et de dimensionner son système en conséquence (ressources, bande passante, etc.).

## 7 Arborescence des données

Les rubriques décrivent une arborescence permettant au citoyen de retrouver rapidement une donnée qui l'intéresse. Elle est décrite selon la structure suivante :

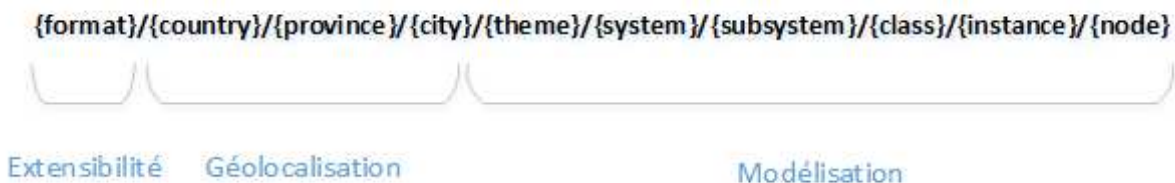


Figure 2 - Arborescence des données

## 7.1 Format

Puisque l'innovation est au cœur des projets de la ville intelligente, il n'y a pas de standard bien établi à ce jour pour l'organisation des données à grande échelle. Le champ format permet de prévoir une extensibilité vers d'autres formats éventuels des rubriques. Pour le moment, la seule valeur possible sera décrite comme une chaîne de caractère « **odtf1** » (Open Data Topic Format Version 1). À noter que cette spécification ne décrit pas la liste des valeurs possibles dans chacun des champs, mais suggère plutôt une méthode de travail pour organiser les données.

## 7.2 Country

Le champ {country} doit suivre la codification à 2 lettres de la norme **ISO 3166-1 Alpha 2**. Pour le Canada, le code utilisé est « **ca** » (réf [https://fr.wikipedia.org/wiki/ISO\\_3166-1](https://fr.wikipedia.org/wiki/ISO_3166-1)).

## 7.3 Province

Le champ {province} doit suivre la codification de la norme **ISO 3166-2**, sans inclure le préfixe du pays. Pour la province du Québec, le code ISO 3166-2 étant CA-QC, le champ {province} sera égal à « **qc** » (réf [https://fr.wikipedia.org/wiki/ISO\\_3166-2:CA](https://fr.wikipedia.org/wiki/ISO_3166-2:CA))

## 7.4 City

Le nom de la ville, ici « **mtl** ».

## 7.5 Theme

Le thème décrit le secteur dans lequel la donnée est impliquée. Voici des thèmes envisagés : « **mobil** » (mobilité), « **gov** » (gouvernance), « **energy** » (énergie), « **educ** » (éducation), « **health** » (santé), « **eco** » (économie), « **env** » (environnement), « **water** » (eau).

## 7.6 System

Le système permet de diviser le thème en grandes catégories. Par exemple, en mobilité urbaine « mobil », il pourrait y avoir les transports routiers « **traf** », le transport en commun « **transit** », etc.

## 7.7 Subsystem

Le sous-système permet de décrire la catégorie d'équipement qui fournit de l'information au système. Par exemple, pour le système « traf » de la mobilité « mobil », il pourrait y avoir des sous-systèmes tels que « **detector** » pour des détecteurs de véhicules, ou « **camera** » si cela concerne des caméras thermiques.

## 7.8 Class

Ce champ permet d'identifier le type d'équipement producteur d'information. Cela peut être un équipement physique (ex : un détecteur modèle Z installé à une intersection, et qui contient une structure de données particulière), mais aussi un équipement virtuel qui pourrait produire des données sur la base d'autres données récupérées ailleurs (ex : un générateur de données statistiques, un serveur informatique, etc.).

## 7.9 Instance

Il s'agit d'un identifiant unique permettant d'identifier qui est le producteur de la donnée.

## 7.10 Node

Le nœud est un élément d'information atomique et indissociable qui décrit une information produite par un fournisseur. Le nœud peut décrire une information simple (température, humidité, image) mais aussi une information complexe (statistiques, liste de nœuds, etc.). Si un producteur transmet une donnée toutes les 30 secondes contenant un groupe de valeurs, alors le nœud correspond à ce groupe de valeurs.

# 8 Le format des nœuds

Les nœuds sont encodés en format JSON. Il s'agit d'un format texte dérivé de la notation des objets du langage JavaScript. La raison de ce choix repose sur le fait qu'une donnée, pour être considérée valide dans un système, doit être nécessairement accompagnée de son contexte. Le contexte doit contenir suffisamment d'information pour que le citoyen qui la consomme puisse interpréter la donnée et juger que la qualité de l'information, et ce sans faire appel au producteur de la donnée. Aucun doute ne doit subsister sur la pertinence de l'information.

En ce qui concerne la casse (organisation des majuscules et minuscules), la convention de casse Pascal devrait être respectée dans la mesure du possible, à moins d'avis contraire, selon les règles suivantes :

- Les acronymes sont en majuscules (Ex : « ODNF1 », « MTL »).
- La première lettre des mots devrait être majuscule si elle n'est pas précédée d'un espace (Ex : « CreationTime », « Creation time », « CreateUTC »).

De la même manière que le format des rubriques, une propriété {Format} sera insérée pour décrire les champs inclus dans un nœud. Le format actuel sera appelé « **ODNF1** » (Open Data Node Format Version 1).

```
{  
  "Format": "ODNF1"  
  "Desc": "Average speed"  
  "CreateUTC": "2017-05-12T08:22:02"  
  "ExpireUTC": "2017-05-12T08:22:15"  
  "Unit": "km/h"  
  "Status": "Good"  
  "Value": 34.21  
}
```

Figure 3 - Format ODNF1, nœud d'une valeur simple



## 8.1 Format

Le format réfère à la liste des propriétés qui décrivent le contexte de la donnée. Le format actuel est le « **Odnf1** ».

### 8.1.1 Desc

Décrit textuellement la donnée présentée, qu'elle soit simple ou complexe.

### 8.1.2 CreateUtc

Date de production de la donnée, selon le format triable ISO 8601 yyyy'-MM'-'dd'T'HH':mm':ss (ref [https://fr.wikipedia.org/wiki/ISO\\_8601](https://fr.wikipedia.org/wiki/ISO_8601)), et pour lequel

- La date est décrite comme suit : yyyy = année, MM = mois, dd = jour
- L'heure est décrite comme suit : HH = heure au format 24h, mm = minute, ss = seconde

Exemple : « 2017-05-17T10:58:23 »

### 8.1.3 ExpireUtc

Moment où la donnée ne peut plus être considérée valide. C'est le producteur de la donnée qui connaît la période à laquelle les données sont transmises. Si le producteur n'est plus en mesure de produire la donnée, le citoyen doit pouvoir juger de la fraîcheur de la donnée. Si la donnée est valide en tout temps, alors la valeur inscrite doit être égale à la valeur « CreateUtc ».

### 8.1.4 Unit

L'unité de mesure permet au consommateur du nœud d'interpréter la valeur afin de la décoder. Si la valeur est une liste d'autres nœud, la valeur à indiquer ici est « **Array** ». Si la valeur est un objet, alors la valeur à indiquer est « **Object** ». Dans tous les autres cas, l'unité de mesure représentera une valeur élémentaire. Si l'unité élémentaire est décrite dans son format abrégé, alors elle sera en minuscules mais, si le nom de l'unité dérive d'un nom propre, la première lettre du symbole est majuscule : N pour newton, J pour joule, Hz pour hertz, V pour volt, A pour ampère, etc. (Ex : « km/h », « Kilomètres par heure », « N/m », « Newton par mètre »). Pour plus d'information sur la nomenclature des unités de mesure, référez-vous à [https://fr.wikipedia.org/wiki/Unit%C3%A9\\_de\\_mesure](https://fr.wikipedia.org/wiki/Unit%C3%A9_de_mesure).

### 8.1.5 Status

Une donnée peut avoir une durée de vie relativement longue, voir même sans expiration (voir le champ ExpireUtc). Dans l'éventualité où le producteur détecte une anomalie avec la donnée, il est responsable d'indiquer à l'utilisateur que cette valeur n'est plus utilisable, dans les délais les plus courts que possibles. Les valeurs possibles sont **Bad**, **Uncertain** et **Good**

### 8.1.6 Value

La valeur d'un nœud peut être soit simple (chaîne de caractères, valeur numérique, null, true/false), soit complexe (un objet), ou soit un tableau de nœuds. Le format des nœuds pouvant évoluer, un nœud parent de format ODNF1 pourrait contenir des nœud enfants d'un format ultérieur, ou vice versa.

```
{
  "Format": "ODNF1"
  "Description": "Statistics data"
  "CreateUTC": "2017-05-12T08:22:02"
  "ExpireUTC": "2017-05-12T08:22:15"
  "Unit": "Object"
  "Status": "Good"
  "Value": {
    "Prop1": "Val1"
    "Prop2": "Val2"
  }
}
```

*Figure 4 - Nœud avec donnée personnalisée*

```
{
  "Format": "ODNF1"
  "Description": "Statistics data"
  "CreateUTC": "2017-05-12T08:22:02"
  "ExpireUTC": "2017-05-12T08:22:15"
  "Unit": "Array"
  "Status": "Good"
  "Value": [
    {node}
    {node}
    {node}
  ]
}
```

*Figure 5 - Nœud complexe comportant une liste de nœuds enfants*

En utilisant les nœud complexes, le producteur a tout la liberté de produire les structures de données de son choix, tout en respectant la contrainte d’avoir à fournir le contexte pour chaque information de la structure.

**\*\* FIN DU DOCUMENT \*\***