Jour 5 – Expressions régulières

Dans le tutoriel précédent un exercice consistait à chercher les magasins de vêtements (`shop=clothes`) et de chaussures (`shop=shoes`). Ajoutez à cela les vêtements en cuir (`shop=leather`), les galeries marchandes (`shop=mall`), la liste de requêtes à inclure dans l'union peut vite s'étoffer.

Lorsqu'on cherche *plusieurs valeurs possibles d'une même clef*, une alternative à l'union consiste à utiliser une **expression régulière** dans un filtre de tag. Pour cela la clef et la valeur doivent être séparées par le caractère tilde ~ et la valeur doit être placée entre quotes (simples ou doubles). Notez que la valeur **doit** être placée entre quotes s'il ne s'agit pas d'un simple mot (si elle contient un espace, un tiret ou tout autre caractère spécial). Il en est de même pour la clef, notamment si elle contient le caractère `:` délimitant un préfixe ou post-fixe.

Les expressions régulières peuvent prendre de nombreuses formes. Dans le cas qui nous intéresse ici, les différentes valeurs recherchées sont placées entre parenthèses et séparées par une barre verticale : `[key~"(val1|val2|val3|…)"]` :

```
[bbox:{{bbox}}];
nwr[shop~"(shoes|clothes|leather|mall)"];
out geom;
```

Soyons précis : cette syntaxe trouve tous les éléments dont la valeur de la clef shop **contient** l'un des mots shoes, clothes, leather ou mall.

La forme la plus simple d'une expression régulière est en réalité `[key~"value"]` : ainsi la requête suivante retourne les éléments pour lesquels la valeur de shop contient 'food' : seafood, frozen_food, health_food, etc. :

```
nwr[shop~'food']({{bbox}});
out geom;
```

Les expressions régulières sont intéressantes sur des clefs comme name. On peut par exemple chercher toutes les rues dont le nom contient le mot “Saint” :

```
way[highway][name~'Saint']({{bbox}});
out geom;
```

On peut préciser si la valeur **commence** par le mot recherché, en plaçant ^ en premier caractère. La requpête suivante trouve toutes les rues dont le nom **commence** par “Allée” :

```
way[highway][name~'^Allée']({{bbox}});
out geom;
```

Et celle-ci, qui utilise $ en dernier caractère, toutes les rues dont le nom **termine** par “Eau” :

```
way[highway][name~'Eau$']({{bbox}});
out geom;
```

Enfin l'option `i` permet d'**ignorer la casse** des caractères, majuscules ou minuscules:

```
way[highway][name~'eau$',i]({{bbox}});
out geom;
```

Ces syntaxes peuvent être combinées. L'exemple suivant trouve les rues dont le nom finit par “saint-pierre”, “saint-denis” ou “saint-jean”, avec ou sans majuscules :

```
[bbox:{{bbox}}];
way[highway][name~'saint-(pierre|denis|jean)$',i];
out geom;
```

Exercices

* Trouvez les restaurants dont le nom commence par “le” ou “la”. * Trouvez les bars, pubs et cafés, dont le tag `website` pointe vers un compte Facebook. * Utilisez une expression régulière pour trouver les magasins de vente ou de réparation de voitures (shop=car ou car_repair) mais pas ceux de caravanes (shop=caravan) ou de tapis (shop=carpet) !