UUID avec PostgreSQL : Pourquoi ? Comment ?

Nicolas Raidelet – IRSTEA

Webcast RDBB – 14 décembre 2017

Qu’est ce que c’est ?

Un UUID : « Identifiant Universel Unique »

Ou plutôt Identifiant Universel à l’unicité très probable mais sans garantie

C’est un entier sur 128-bit avec un formatage particulier

Ça peut se représenter comme ça :

379dae82-5a2b-4c4b-8193-b8e7749a3495

C’est normé : RFC 4122 de 2005

Elle définit 5 variantes (généralement on utilise la 4ème version : aléatoire)

Pourquoi faire ?

Pour identifier de manière unique une ressource sans avoir besoin d'un registre central

=> très utile dans le cas de systèmes répartis

Exemples :

- base de données d'embarquées
- besoin de synchro entre différentes instances d'une même application

Le type PostgreSQL UUID

Type natif de PostgreSQL

Permet de stocker et de comparer des UUID

Ne permet pas de les générer nativement

Format standard :

a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11

Formats en entrée :

A0EEBC99-9C0B-4EF8-BB6D-6BB9BD380A11

{a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11}

a0eebc999c0b4ef8bb6d6bb9bd380a11

a0ee-bc99-9c0b-4ef8-bb6d-6bb9-bd38-0a11

{a0eebc99-9c0b4ef8-bb6d6bb9-bd380a11}

Les avantages et inconvénients

- Pas de registre central
- Aléatoire donc non séquentiel et non prévisible
- Entier 128-bit donc rapide et peu coûteux
- Peu lisible par un humain

Comment en générer avec linux ?

$ uuidgen

31ea437c-010c-43ed-8d82-be2643abbb4f

Doc :

https://linux.die.net/man/1/uuidgen

Comment en générer avec php ?

Installer la lib : ramsey/uuid

```
$uuid = \Ramsey\Uuid\Uuid::uuid4();
echo $uuid→toString();
```

Doc :

https://github.com/ramsey/uuid/wiki/Ramsey%5CUuid-Cookbook

Comment en générer avec Postgres ? 1/2

Utiliser le lib uuid-ossp

Sous debian, elle est disponible dans le paquet

Postgresql-contrib-9.6

On peut en suite l'activer avec la commande

```
postgres@lyp4029:~$ psql -d mypgdatabase
mypgdatabase=# CREATE EXTENSION "uuid-ossp" ;
```

Doc :

https://www.postgresql.org/docs/9.6/static/uuid-ossp.html

Comment en générer avec Postgres ? 2/2

```
INSERT
INTO test_uuid (uuid, nom)
VALUES (uuid_generate_v4(), 'toto');
```

Conclusion

C’est simple et très recommandé en cas de base de données « multiple »