Nettoyage de la Base de données SQLITE

Recherche des capteurs

Donc, on va en profiter pour identifier s'il y a des capteurs qui ne nous intéressent pas et qui prennent beaucoup de place, et on va dire au système de les exclure.

Dans l'onglet **Exécuter SQL,** nous allons exécuter la requête SQL suivante qui nous montrera les 10 entités qui ont le plus d'enregistrements dans le fichier téléchargé. Nous écrivons ou copions simplement la requête et exécutons

```
SELECT entity_id, COUNT(*) as count FROM states GROUP BY entity_id ORDER BY count DESC LIMIT
10
```

Suppression des capteurs

Maintenant il faut supprimer les capteurs volumineux dans la DB, pour ce faire exécutez la commande suivante

```
SELECT *
FROM "states"
where entity_id='sensor.humidity'
```

La commande `VACUUM` nettoie votre base de données.

Nettoyons et voyons les résultats

Eh bien, nous pouvons maintenant dire à notre système quelles entités nous ne voulons pas pour l'optimisation de la base de données.

Cela peut se faire directement dans le fichier configuration.yaml, mais petit à petit ce fichier devient difficile à lire s'il est très long. Au lieu de cela, nous vous dirons d'inclure le contenu de quelqu'un d'autre. Dans **configuration.yaml,** nous écrirons :

```
recorder: !include recorder.yaml
```

Et nous allons créer, dans le même répertoire de configuration, le fichier **recorder.yaml** .

Gardez à l'esprit que vous devez mettre vos capteurs, ne copiez pas cet exemple :

```
  purge_keep_days: 30
  auto_purge: true
  commit_interval: 10
  exclude:
    entities:
      - sensor.router_packets_received
      - sensor.router_b_received
      - sensor.router_packets_sent
      - sensor.router_b_sent
      - sensor.router_packets_s_sent
      - sensor.router_packets_s_received
      - sensor.router_kib_s_sent
      - sensor.enchufe_blitzwolf_1_linkquality
      - sensor.enchufe_blitzwolf_3_linkquality
      - sensor.enchufe_blitzwolf_2_linkquality
      - sensor.enchufe_blitzwolf_1_voltage
      - sensor.enchufe_blitzwolf_3_voltage
      - sensor.enchufe_blitzwolf_1_current
      - sensor.enchufe_blitzwolf_2_voltage
      - sensor.date_time
      - sensor.date_time_iso
      - sensor.date_time_utc
      - sensor.time
      - sensor.time_date
      - sensor.time_utc
```

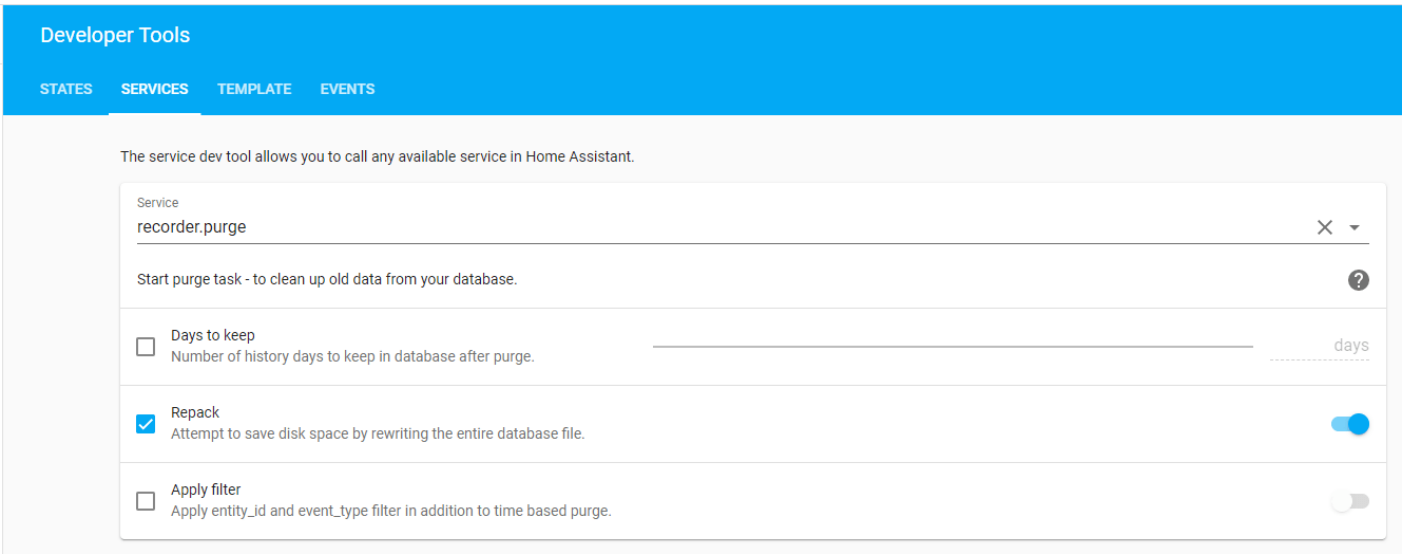
Ici, nous nous arrêtons un peu et expliquons les valeurs que nous avons choisies pour configurer l'enregistreur lorsqu'il s'agit d'optimiser la base de données sqlite :

- **purge_keep_days : 30** - On vous dit de gagner 30 jours… au total, maintenant on va l'optimiser et on a une place. Par défaut, la base de données enregistre 10 jours.
- **auto_purge: true** - Pour dire à Home Assistant de purger la base de données 1 fois/jour et de ne pas la développer indéfiniment.
- **commit_interval : 10** - Par défaut, cette valeur est de 1 seconde. Nous préférons que vous écriviez des blocs toutes les 10 secondes. La dégradation des mémoires flash (comme une carte **microSD**) est connue lorsqu'un nombre donné d'opérations de lecture/écriture est atteint (généralement environ 100 000). Nous préférons avoir l'historique des événements et des états avec 10 secondes de retard que de voir comment notre système se dégrade prématurément.

Réduire la taille du fichier

Compte tenu de tout ce qui précède à propos de l'optimisation de la base de données sqlite, il se peut qu'il nous reste un fichier de base de données assez volumineux. Merci Luis pour vos contributions sur ce sujet.

Dans ce cas, nous pouvons appeler le service avec l'option repack dans la section Outils de développement de notre Home Assistant :



Après avoir utilisé manuellement l'option repack, le fichier peut être considérablement réduit, et cela a pris beaucoup de temps (plusieurs heures) dans un RPI 4, cela dépend de chaque cas particulier.

Repack supprime les enregistrements précédemment marqués comme supprimés par le service de purge (auto_purge) chaque nuit, mais il le fait en créant une copie de travail de la base de données et en la refaisant, il doit donc y avoir suffisamment d'espace pour les fichiers temporaires créés également (home -assistant_v2.db-wal et home-assistant_v2.db-shm).

Révision #3
Créé 2021-10-17 00:00:04 UTC par garfieldtux
Mis à jour 2021-10-17 00:38:17 UTC par garfieldtux