

Stockage des données contextuelles dans le système de fichiers

Les données stockées dans les variables de contexte, de flux et globales sont appelées données de contexte et sont normalement stockées en mémoire.

Cela signifie que si vous redémarrez le flux, les données sont perdues.

Cependant à partir de la version 0.19 il est possible de stocker les données de contexte dans le système de fichiers.

Pour ce faire, vous devrez modifier le fichier de paramètres et ajouter l'entrée suivante :

[context.js](#)

```
contextStorage: {
  default: "memoryOnly",
  memoryOnly: { module: 'memory' },
  file: { module: 'localfilesystem' }
},
```

Peu importe où vous le placez dans le fichier de paramètres et selon la version de node-red avec laquelle vous avez commencé, il se peut que vous ayez déjà une entrée commentée.

Les paramètres ci-dessus configurent node-red pour utiliser le magasin par défaut en mémoire ainsi que le système de fichiers pour le magasin de fichiers.

Nous avons donc deux magasins.

Lorsque vous enregistrez des données dans les variables de contexte ou que vous en récupérez des données, vous devrez spécifier le magasin dans lequel elles se trouvent. La valeur par défaut est en mémoire.

Donc si vous utilisez :

```
var count=context.get("count");
```

Vous récupérerez la variable count de la mémoire et pour obtenir la variable count du magasin de fichiers, utilisez :

```
context.get("count", "file");
```

Pour stocker des données, utilisez :

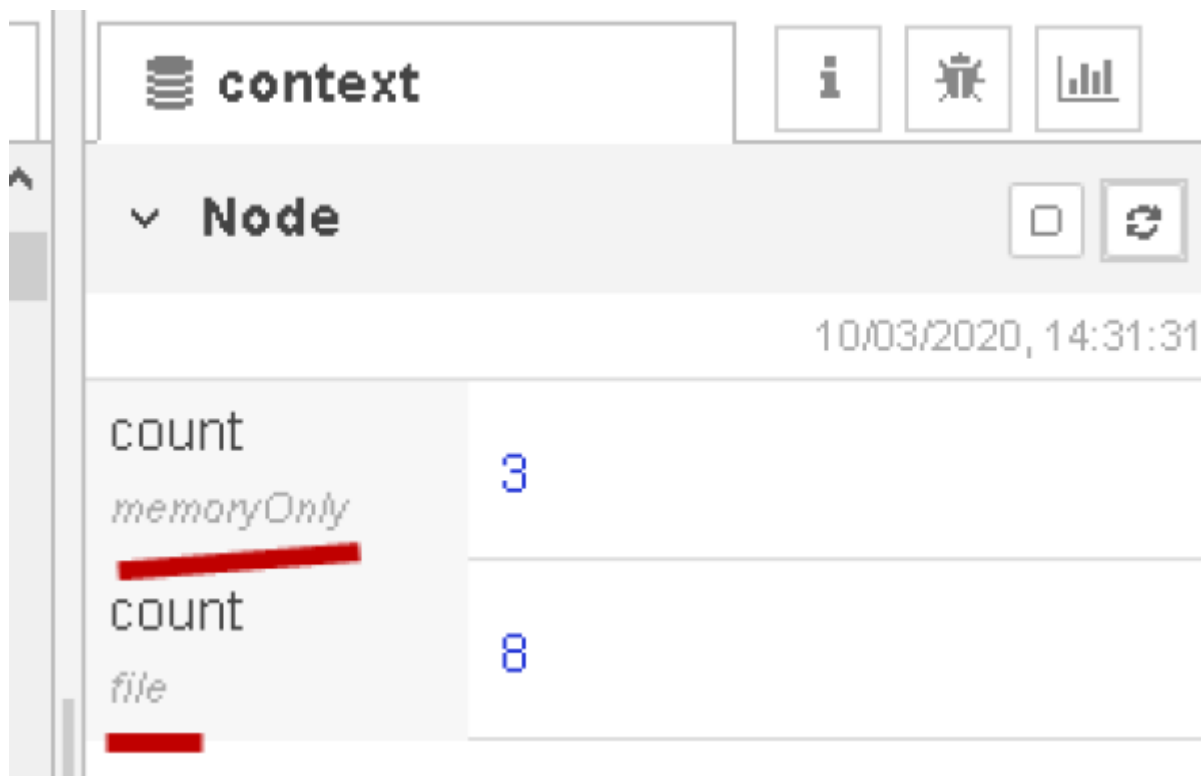
```
context.set("count", count,"file");
```

Le système stocke les variables dans un fichier JSON dans un dossier appelé contexte sous le dossier .node-red .

Même si vous stockez des données dans le système de fichiers, il est toujours possible de perdre des données car elles ne sont vidées vers le système de fichiers que toutes les 30 secondes.

Vous pouvez modifier cela (pas vraiment besoin) et d'autres options de configuration, voir la documentation ici .

Vous devez également noter que vous pouvez avoir deux variables portant le même nom mais stockées dans la mémoire et dans le système de fichiers, comme indiqué dans la capture d'écran ci-dessous :



The screenshot shows a Node.js dashboard with a 'context' header and a 'Node' section. The 'Node' section displays two variables: 'count' with a value of 3, labeled 'memoryOnly', and 'count' with a value of 8, labeled 'file'. The 'file' label is highlighted with a red underline. The dashboard also shows a timestamp '10/03/2020, 14:31:31' and several icons (info, bug, chart) in the top right corner.

Variable	Value
count (memoryOnly)	3
count (file)	8

système de fichiers de données de contexte de magasin

Si vous disposez de plusieurs magasins de contexte, lorsque vous utilisez des nœuds tels que le nœud de modification, un choix vous sera présenté comme indiqué ci-dessous :

magasin de contexte

Cependant, vous souhaitez peut-être utiliser uniquement le magasin de fichiers et pour éviter toute confusion, il est alors plus facile d'activer uniquement cette option en supprimant l'option de mémoire dans le fichier settings.js, comme indiqué ci-dessous :

```
contextStorage : { par défaut : "fichier", fichier : { module : 'localfilesystem' } },
```

Vous devez être très prudent lorsque vous utilisez plusieurs magasins, car cela rend les flux difficiles à déboguer.

From:

<https://www.magenealogie.chanterie37.fr/www/fablab37110/> - Castel'Lab le Fablab MJC de Château-Renault

Permanent link:

<https://www.magenealogie.chanterie37.fr/www/fablab37110/doku.php?id=start:raspberry:nodered:stokagedonnees&rev=1708964749>

Last update: **2024/02/26 17:25**

