

Variablen

Variablen können dynamisch zur Laufzeit vergeben werden und in der App, per [Konfigurationstool](#) integriert werden oder zur Laufzeit aus der Datei vars.json geladen werden. Falls Fehler in der Datei vorhanden sind, wird die App mit entsprechendem Hinweis beendet. Zum bearbeiten empfiehlt sich eine Editor wie Notepad++ oder ein [JSON online Editor](#).

Benutzung von Variablen

Variablen können auf drei wege gesetzt werden. Der Aufruf von variablen erfolg über: &+name

Als Beispiel:

- Einer Variable mit dem Namen „test“ wurde der Wert 5 zugewiesen.
- Der Kommando scan &test würde einen scan mit der Dauer 5 durchführen.

Variablen können jeden Wert enthalten, auch Kommandos. Der Name von Variablen darf keine Leerzeichen enthalten!

Laden von Variablen aus Datei

Falls das laden von externen Variablen aktiviert ist, können diese über die Benutzeroberfläche geladen werden. Zum laden der Datei ist der Menüpunkt „Load Vars“ zu benutzen. Hier kann eine Datei ausgewählt werden die die benötigten variablen enthält.

Variablen über Kommando setzen

In Standardkonfiguration, setzt der Kommando set eine Variable. Der Syntax ist:

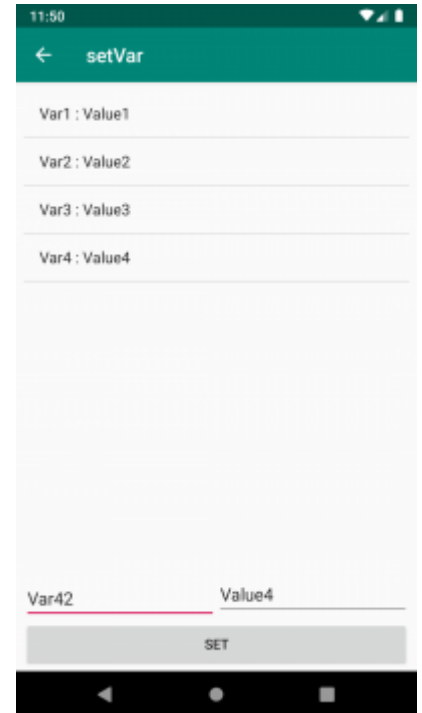
```
set Name Wert;
```

(siehe Kommandos)

Variablen über Oberfläche setzen

Die Oberfläche zur Variablen Verwaltung lässt sich über den Menüpunkt „Add Var“ öffnen. Hier lassen

sich alle aktuellen variablen anzeigen,löschen oder neue hinzugefügt werden.



Links des : ist der Name, rechts der Wert der Variable.

Variable hinzufügen:

Hinzugefügt werden Variablen über die zwei Textfelder. Im linken wird der Name der Variable definiert, im rechten ihr Wert. Mit druck auf den Button „Set“ wird sie hinzugefügt

Variable löschen:

Zum löschen muss die gewünschte Variable, in der Liste, gedrückt und gehalten werden.

Umgang mit der Datei vars.json

Die vars.json Datei ist eine Auflistung einzelner Blöcke.

Block

Einzelne Variablen sind in Blöcken organisiert. Ein block sieht wie folgt aus:

```
{
  "name": "Name",
  "cont": "wert"
}
```

Diese Variable würden den Namen. „Name“ tragen und den Wert „wert“ enthalten.

Mehrere Variablen

Einzelne Blöcke werden jeweils per Kommata getrennt, eine Ausnahme hierbei stellt der letzte Block dar.

Beispiel:

```
[
  {
    Block1
  },
  {
    Block2
  }
]
```

Als letztes ist es wichtig, alle Blöcke in [] zu verschachteln.

Fertige Datei

Als Beispiel für eine syntaktisch korrekte vars.json, hier eine Datei die fünf Variablen definiert.

```
[
  {
    "name": "Var1",
    "cont": "Val1"
  },
  {
    "name": "Var2",
    "cont": "scan 5;"
  },
  {
    "name": "Var3",
    "cont": "XX:XX:XX:XX:XX:XX;"
  },
  {
    "name": "var4",
    "cont": "2a00;"
  },
  {
    "name": "var5",
    "cont": "'test'"
  }
]
```

Eine Datei mit nur einer Variable:

```
[  
  {  
    "name": "eineVariable",  
    "cont": "Wert"  
  }  
]
```

From:

<https://ble-toolkit2.fynnhr.com/> - **BLE-Toolkit 2.0**

Permanent link:

<https://ble-toolkit2.fynnhr.com/doku.php?id=variablen>

Last update: **2019/05/24 14:27**

