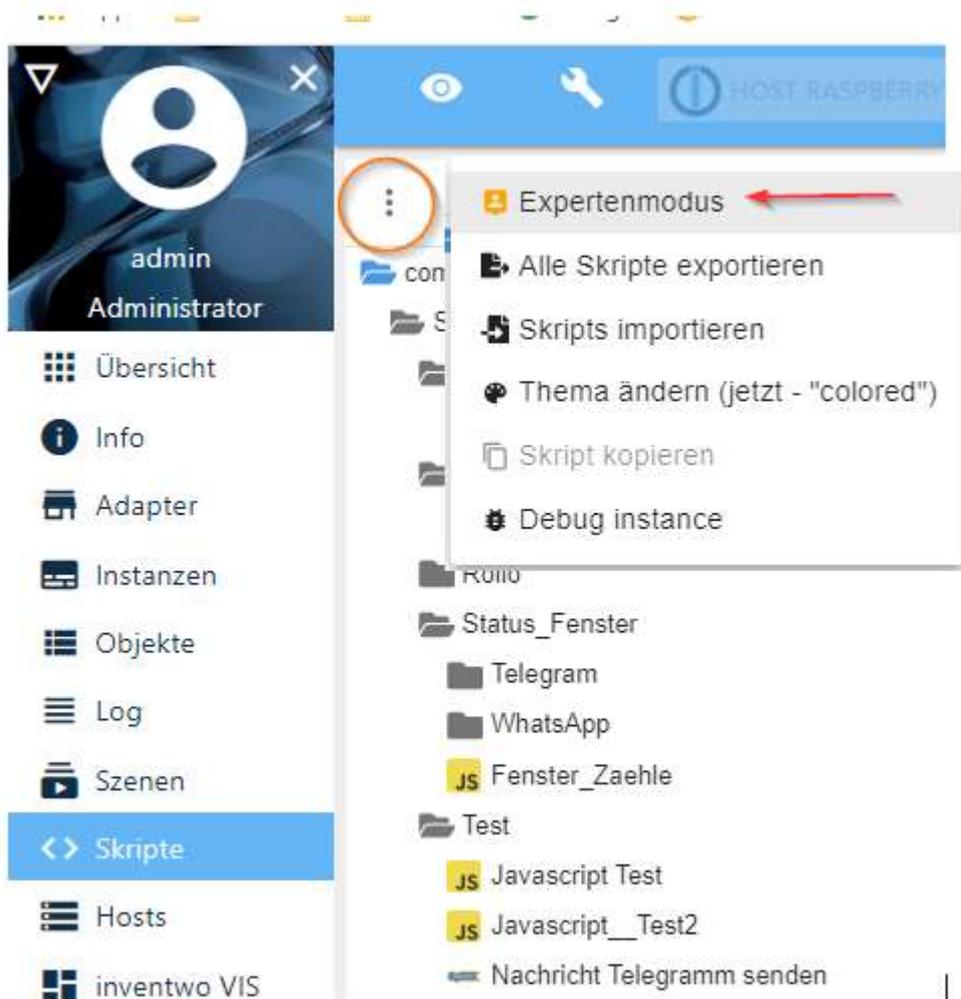


## Debuggen mit IOBroker Skripte

Um das Debuggen zu ermöglichen muss man erst einmal in den Expertenmodus schalten:

Die 3 Punkte anklicken und dann Expertenmodus.



Jetzt sieht man auch diesen Käfer:



Jetzt das Script auswählen und den Käfer drücken.

Jetzt kommt folgende Meldung

Bist du sicher?

Das Skript wird gestoppt und muss nach dem Debuggen manuell aktiviert werden. Fortsetzen?

JA    ABBRECHEN

JA -> weiter

Kann ein paar Sekunden dauern bis die nächste Anzeige kommt.

```
1  ((async () => {
2  var myMdwTheme = {};
3  myMdwTheme.sentry = {};
4  myMdwTheme.sentry.getId = function () { if(existsState("vis-materialdesign.0.sentry")) { re
5  myMdwTheme.sentry.getValue = function () { if(existsState("vis-materialdesign.0.sentry")) {
6
7
8
9  createState('_0_userdata.0.Test_Simulation.JavaScript_starten', false, {name: 'Skript bei VI
10
11 on('_0_userdata.0.Test_Simulation.JavaScript_starten', function (obj) {
12   if (!obj.state.ack && obj.state.val) {
13     // was soll passieren?
14     console.log('-----SelectorUNREACH-----');
15 console.log(Object.values(SelectorUNREACH));
16 console.log('-----Variable instanz ' + instanz);
17
18     // hier die Ausführung rein
19   }
20 });
21
22
```

STACK	CONSOLE
anonymous <i>common.Script.Test.Javascript__Test2 (2:0)</i>	+ LOCAL myMdwTheme=undefined (object)
anonymous <i>common.Script.Test.Javascript__Test2 (52:2)</i>	LOCAL instanz=undefined (undefined)
runInContext <i>vm.js (129:17)</i>	LOCAL idAnzahl=undefined (undefined)
runInNewContext <i>vm.js (134:16)</i>	LOCAL idText=undefined (undefined)
execute <i>main.js (1482:26)</i>	LOCAL SelectorUNREACH=undefined (undefined)
	LOCAL SelectorSTICKY_UNREACH=undefined (undefined)
	LOCAL SelectorLOWBAT=undefined (undefined)
	LOCAL SelectorCONFIG_PENDING=undefined (undefined)

Die Variablen aus dem Script werden automatisch eingetragen.

Zeilenweise ausführen



Die Inhalte der Variablen werden dann automatisch eingetragen.

```
25
26
27 var instanz = 'javascript.0.'
28 var idAnzahl = instanz + 'Servicemeldungen.Homematic.Anzahl',
29     idText = instanz + 'Servicemeldungen.Homematic.Text';
30
31 var SelectorUNREACH = $('channel[state.id=*.UNREACH]');
32 var SelectorSTICKY_UNREACH = $('channel[state.id=*.STICKY_UNREACH]');
33 var SelectorLOWBAT = $('channel[state.id=*.LOWBAT]');
34 var SelectorCONFIG_PENDING = $('channel[state.id=*.CONFIG_PENDING]');
35 console.log('-----Position 2-----');
36 console.log('Pause 1');
37 await sleep(1000);
38
```

STACK	CONSOLE
anonymous common.Script.Test.Javascript__Test2 (32:21)	+ LOCAL myMdwTheme=undefined (object)
anonymous common.Script.Test.Javascript__Test2 (52:2)	LOCAL instanz="javascript.0." (string)
runInContext vm.js (129:17)	LOCAL idAnzahl="javascript.0.Servicemeldungen.Homematic.Anzahl" (string)
runInNewContext vm.js (134:16)	LOCAL idText="javascript.0.Servicemeldungen.Homematic.Text" (string)
	LOCAL SelectorUNREACH=undefined (object)
	LOCAL SelectorSTICKY_UNREACH=undefined (object)
	LOCAL SelectorLOWBAT=undefined (undefined)
	LOCAL SelectorCONFIG_PENDING=undefined (undefined)

Leider kein Inhalt bei der Zuweisung von \$ Funktionen / Objekte.

z.B.: `var SelectorUNREACH = $('channel[state.id=*.UNREACH]');`

Ausgabe: `SelectorUNREACH =undefined(object)`

### **Breakpoint**

Links auf die Nummer klicken

```

32 var SelectorSTICKY_UNREACH = $('channel[state.id=*.STICKY_UNREACH]');
33 var SelectorLOWBAT = $('channel[state.id=*.LOWBAT]');
34 var SelectorCONFIG_PENDING = $('channel[state.id=*.CONFIG_PENDING]');
35 console.log('-----Position 2-----');
36 console.log('Pause 1');
37 await sleep(1000);
38
39 console.log('-----');
40 console.log('Ausgabe der Variable: idAnzahl = ' + idAnzahl);
41 // console.log('Ausgabe der Variable: SelectorUNREACH = ' + SelectorUNREACH);
42 // console.log('Ausgabe der Variable: SelectorUNREACH = ' + Object.values(SelectorUNREACH));

```

Dann hält er auch dort an wenn man „Resume execution“ drückt.



Beim Neustart scheint er dann wieder aus dem Debuggen auszusteigen