4. HomeyScript for storing energy day ahead prices from Tibber

This script will create a sheet like this: (Clik on it so see a large version)

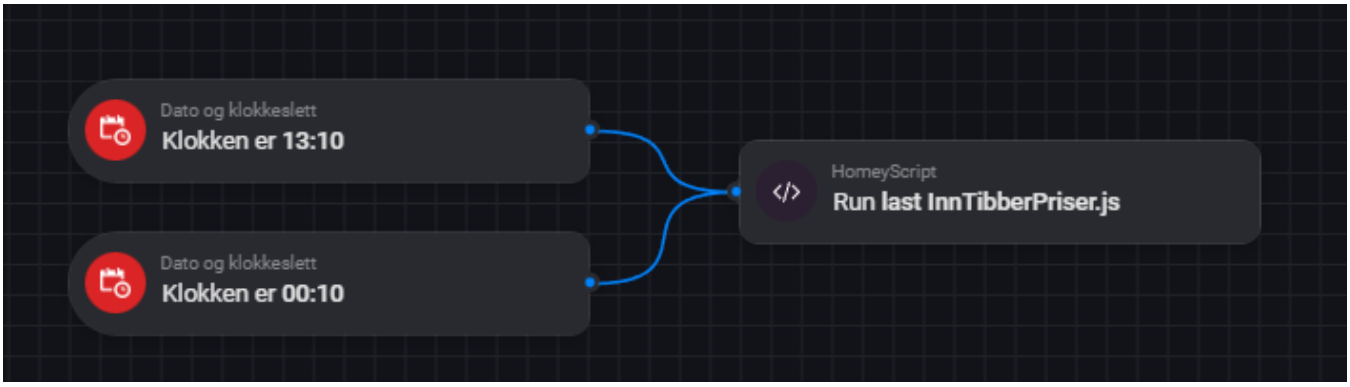
11.1.2023 kl. 06.00.03	11.01.2023 kl. 00.00.00	0,45	0,57
11.1.2023 kl. 06.00.04	11.01.2023 kl. 01.00.00	0,30	0,38
11.1.2023 kl. 06.00.05	11.01.2023 kl. 02.00.00	0,27	0,35
11.1.2023 kl. 06.00.07	11.01.2023 kl. 03.00.00	0,24	0,31
11.1.2023 kl. 06.00.08	11.01.2023 kl. 04.00.00	0,27	0,35
11.1.2023 kl. 06.00.08	11.01.2023 kl. 05.00.00	0,28	0,37
11.1.2023 kl. 06.00.09	11.01.2023 kl. 06.00.00	0,66	0,83
11.1.2023 kl. 06.00.09	11.01.2023 kl. 07.00.00	1,06	1,34
11.1.2023 kl. 06.00.10	11.01.2023 kl. 08.00.00	1,27	1,60
11.1.2023 kl. 06.00.10	11.01.2023 kl. 09.00.00	1,35	1,70
11.1.2023 kl. 06.00.11	11.01.2023 kl. 10.00.00	1,36	1,71
11.1.2023 kl. 06.00.11	11.01.2023 kl. 11.00.00	1,33	1,68
11.1.2023 kl. 06.00.11	11.01.2023 kl. 12.00.00	1,25	1,57
11.1.2023 kl. 06.00.12	11.01.2023 kl. 13.00.00	1,21	1,52
11.1.2023 kl. 06.00.12	11.01.2023 kl. 14.00.00	1,24	1,55
11.1.2023 kl. 06.00.13	11.01.2023 kl. 15.00.00	1,26	1,58
11.1.2023 kl. 06.00.13	11.01.2023 kl. 16.00.00	1,27	1,60

This is how you would run it:

It should be executed just past midnigth, becaus tomorrow is just become today, and tomorrows prices is empty at midnigth.

Next run should be aprox 14.00, (as the Tibber API sometimes seems to be late on getting the prices), because tomorrow prices will then have arrievd and are ready to be stored in the spreadsheet. The time of 13.10 below is just an example.

Note : *The name of the script is of your choice, so example picture below must be adjusted to match your name of the script*



Note ! This script now is utilizing the multi row feature in Google API, so if you have a Easy Logger delimiter set to **;** this script **WILL NOT WORK** . You **MUST** change the delimiter to another char, prefereably **|**

For Homey 2023

id must be :

```
id: 'homey:app:no.bwa.easy-logger:set-cell-delimited-data', instead of  id: 'set-cell-delimited-data',
id: 'homey:app:no.bwa.easy-logger:create-sheet', instead of  id: 'create-sheet',
```

```
/*

This script will create a sheet with sheetName into your Google Spreadsheet
configured from the Easy Logger App

You must have this App configured and Running
```

```
Script developed by Bjørn-Willy Arntzen bwa@bwa.no

versionlog :
- 04.01.2023.1 - Fix missing clear of tomorrowsdata column
- 11.01.2023.1 - Formatting of date from tibber
- 18.01.2023.1 - More effectice execution of Google API calls.
               - settable delimiter

*/

const startRow = 5; // the first row for real data in the sheet, should be at least 5
const sheetName = 'TibberPriceInfo';
const delimiter = '|';  // change this to the delimiter your app has, but its mandatory to NOT BE ;

// this is the demo token, replace this with your own Tibber Api Key
const tibberApiKey = '5K4MVS-OjfWhK_4yrjOlFe1F6kJXPVf7eQYggo8ebAE';

const locale = "no-NO";  // currently not implementet yet

Date.prototype.addHours= function(h){
    this.setHours(this.getHours()+h);
    return this;
}

function toTimestampString(d) {

        // typisk: 17.12.2022 kl. 18.07.41

        return  (d.getDate() < 10 ? '0' + d.getDate() : d.getDate() )+
                "." + ((d.getMonth()+1) < 10 ? '0' + (d.getMonth()+1) : (d.getMonth()+1)) +
                "." + d.getFullYear() + " kl. " +
                      (d.getHours()   < 10 ? '0'+d.getHours()     : d.getHours()) +
                "." + (d.getMinutes() < 10 ? '0' + d.getMinutes() : d.getMinutes()) +
                "." + (d.getSeconds() < 10 ? '0' + d.getSeconds() : d.getSeconds());


}

async function setCell(cellName, cellData) {


    try {
      await Homey.flow.runFlowCardAction({
        uri: 'homey:app:no.bwa.easy-logger',
        id: 'homey:app:no.bwa.easy-logger:set-cell-delimited-data',
        args: {
          'cell-name': cellName,
          'delimited-data': '' + cellData
        },
      });

      // console.log("setCell ", cellName, " data", cellData);
    } catch(err) {
      console.log(err);
    }
}


async function createSheet(sheetName) {


    try {
      await Homey.flow.runFlowCardAction({
        uri: 'homey:app:no.bwa.easy-logger',
        id: 'homey:app:no.bwa.easy-logger:create-sheet',
        args: {
          'sheet-name': sheetName
        },
      });

    } catch(err) {
```

```
      console.log(err);
    }
}

await createSheet(sheetName);

try {

  const query = '{ "query" \: "{ viewer { homes { currentSubscription{ priceInfo{ today { total energy tax
startsAt } tomorrow { total energy tax startsAt }}}}}" } ';

    const response = await fetch(
      'https://api.tibber.com/v1-beta/gql',
      {
        method: 'POST',
        headers: {
          'Authorization': 'Bearer ' + tibberApiKey,
          'Content-Type': 'application/json'
        },
        body: query
  });

  if(response.status !== 200) {
    throw Error("Tibber call failed. Details: ",response);
  }

  const body = await response.json();

  // the next lines may be handy if debug loggeing is needed
  //console.log(body.data.viewer);
  //console.log(body.data.viewer.homes[0]);

  let cellHeating = sheetName + "!"+ "A" + (startRow -3);
  await setCell(cellHeating, "Todays prices");

  cellHeating = sheetName + "!"+ "A" + (startRow -2);
  await setCell(cellHeating, "Time for price"+delimiter+"Energy"+delimiter+"Total cost");


  /*

    ={1;2;3}|={4;5;6}|={7;8;9}

    gir dette resultatet :

    1   4   7
    2   5   8
    3   6   9

  */
  let column1 = '={';
  let column2 = '={';
  let column3 = '={';

  // column 1
  for (let i = 0; i < body.data.viewer.homes[0].currentSubscription.priceInfo.today.length; i++) {
    let cell = sheetName + "!"+ "A" + (startRow + i);
    let d = new Date(body.data.viewer.homes[0].currentSubscription.priceInfo.today[i].startsAt).addHours(1);

    column1 += '"' +  toTimestampString(d) + '"';
    column2 += body.data.viewer.homes[0].currentSubscription.priceInfo.today[i].energy ;
    column3 += body.data.viewer.homes[0].currentSubscription.priceInfo.today[i].total ;



    if(i < body.data.viewer.homes[0].currentSubscription.priceInfo.today.length-1) {
      column1 += ";" ;
      column2 += ";" ;
      column3 += ";" ;
    }
```

```
  }
  column1 += "}";
  column2 += "}";
  column3 += "}";


  /*  TOMOROW   */

  let cell = sheetName + "!"+ "A" + (startRow );
  await setCell(cell, column1 + delimiter +"#" + column2 + delimiter +"#" + column3);


  cellHeating = sheetName + "!"+ "A" + (startRow -3 + 28);
  await setCell(cellHeating, "Tomorrows prices");

  cellHeating = sheetName + "!"+ "A" + (startRow -2 + 28);
  await setCell(cellHeating, "Time for price"+delimiter+"Energy"+delimiter+"Total cost");


  column1 = '={';
  column2 = '={';
  column3 = '={';

  // column 1
  for (let i = 0; i < 24; i++) {

    // ikke data klart
    if ( body.data.viewer.homes[0].currentSubscription.priceInfo.tomorrow.length == 0 ) {

      let cell = sheetName + "!"+ "A" + (startRow + i);

      column1 += '"- - - data is not ready until aprox 13.00 - - -"';
      column2 += '0';
      column3 += '0';



    } else {

      // data er klart

      let d = new Date(body.data.viewer.homes[0].currentSubscription.priceInfo.tomorrow[i].startsAt).addHours
(1);
      column1 += '"' +  toTimestampString(d) + '"';
      column2 += body.data.viewer.homes[0].currentSubscription.priceInfo.tomorrow[i].energy ;
      column3 += body.data.viewer.homes[0].currentSubscription.priceInfo.tomorrow[i].total ;
    }

    if(i < 23) {
      column1 += ";" ;
      column2 += ";" ;
      column3 += ";" ;
    }


  }
  column1 += "}";
  column2 += "}";
  column3 += "}";

  cell = sheetName + "!"+ "A" + (startRow+28 );
  await setCell(cell, column1 + delimiter + "#" + column2 + delimiter + "#" + column3);


} catch (err)  {
  console.log("Something bad happened");
  console.log(err)
}
```