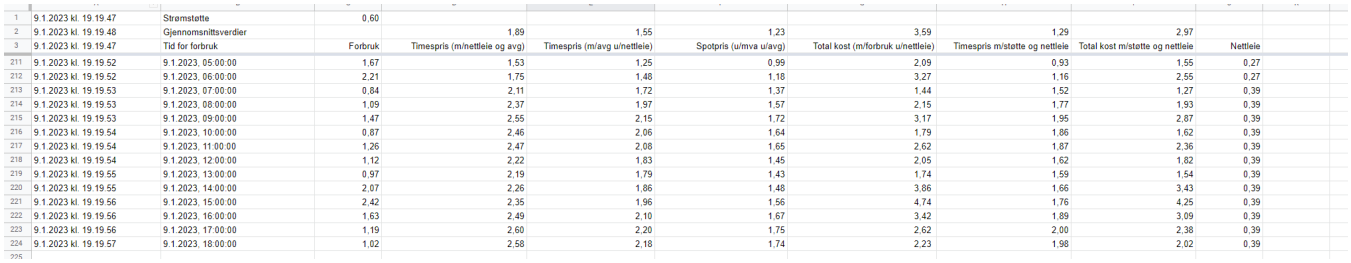
3. HomeyScript for storing energy consumption data from Tibber

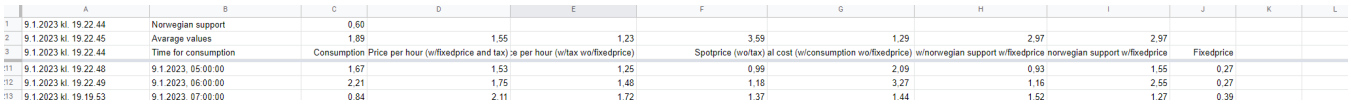
This script will create a a sheet for each month of the year, like this: (Click on image to see large version)

Norwegian version :



1	9.1.2023 kl. 19.19.47	Strømstøtte	0,60							
2	9.1.2023 kl. 19.19.48	Gjennomsnittsverdier		1,89	1,55	1,23	3,59	1,29	2,97	
3	9.1.2023 kl. 19.19.47	Tid for forbruk	Forbruk	Timespris (m/nettleie og avg)	Timespris (m/avg u/nettleie)	Spotpris (u/mva u/avg)	Total kost (m/forbruk u/nettleie)	Timespris m/støtte og nettleie	Total kost m/støtte og nettleie	Nettleie
211	9.1.2023 kl. 19.19.52	9.1.2023, 05:00:00	1,67	1,53	1,25	0,99	2,09	0,93	1,55	0,27
212	9.1.2023 kl. 19.19.52	9.1.2023, 06:00:00	2,21	1,75	1,48	1,18	3,27	1,16	2,55	0,27
213	9.1.2023 kl. 19.19.53	9.1.2023, 07:00:00	0,84	2,11	1,72	1,37	1,44	1,52	1,27	0,39
214	9.1.2023 kl. 19.19.53	9.1.2023, 08:00:00	1,09	2,37	1,97	1,57	2,15	1,77	1,93	0,39
215	9.1.2023 kl. 19.19.53	9.1.2023, 09:00:00	1,47	2,55	2,15	1,72	3,17	1,95	2,87	0,39
216	9.1.2023 kl. 19.19.54	9.1.2023, 10:00:00	0,87	2,46	2,06	1,64	1,79	1,86	1,62	0,39
217	9.1.2023 kl. 19.19.54	9.1.2023, 11:00:00	1,26	2,47	2,08	1,65	2,62	1,87	2,36	0,39
218	9.1.2023 kl. 19.19.54	9.1.2023, 12:00:00	1,12	2,22	1,83	1,45	2,05	1,62	1,82	0,39
219	9.1.2023 kl. 19.19.55	9.1.2023, 13:00:00	0,97	2,19	1,79	1,43	1,74	1,59	1,54	0,39
220	9.1.2023 kl. 19.19.55	9.1.2023, 14:00:00	2,07	2,26	1,86	1,48	3,86	1,66	3,43	0,39
221	9.1.2023 kl. 19.19.56	9.1.2023, 15:00:00	2,42	2,35	1,96	1,56	4,74	1,76	4,25	0,39
222	9.1.2023 kl. 19.19.56	9.1.2023, 16:00:00	1,63	2,49	2,10	1,67	3,42	1,89	3,09	0,39
223	9.1.2023 kl. 19.19.56	9.1.2023, 17:00:00	1,19	2,60	2,20	1,75	2,62	2,00	2,38	0,39
224	9.1.2023 kl. 19.19.57	9.1.2023, 18:00:00	1,02	2,58	2,18	1,74	2,23	1,98	2,02	0,39

English version:



	A	B	C	D	E	F	G	H	I	J	K	L
1	9.1.2023 kl. 19.22.44	Norwegian support	0,60									
2	9.1.2023 kl. 19.22.45	Avarage values	1,89	1,55	1,23	3,59	1,29	2,97	2,97			
3	9.1.2023 kl. 19.22.44	Time for consumption	Consumption	Price per hour (w/fixedprice and tax)	e per hour (w/tax wo/fixedprice)	Spotprice (wo/tax)	al cost (w/consumption wo/fixedprice)	w/norwegian support w/fixedprice	norwegian support w/fixedprice	Fixedprice		
11	9.1.2023 kl. 19.22.48	9.1.2023, 05:00:00	1,67	1,53	1,25	0,99	2,09	0,93	1,55	0,27		
12	9.1.2023 kl. 19.22.49	9.1.2023, 06:00:00	2,21	1,75	1,48	1,18	3,27	1,16	2,55	0,27		
13	9.1.2023 kl. 19.19.53	9.1.2023, 07:00:00	0,84	2,11	1,72	1,37	1,44	1,52	1,27	0,39		

This has a prerequisite that you have

- a Tibber subscription and a ApiKey of your own
- Installed and configured Homey Easy Logger
- You **must replace** the demo token (**5K4MVS-OjfWhK_4yrjOlFe1F6kJXPVf7eQYggo8ebAE**) with you own
- You must change the delimiter from **;** to **|**
- Row 2 has now average value of all columns
- Tip: You may want to lock the 3 first rows in the sheet
- Tip: You may also rigth justify the row3 to get better alignments
- You should put 'nettleie' into `function calulateNettLeieFor(time)` with your corresponding value
  - in the eample this is programmed as 0.28 on daytime and 0.16 at nigth :

```
function calulateNettLeieFor(time) {

  // Glitre Energi sine priser
  if (time < 22 && time > 06) {
    //console.log("calulateNettLeieFor", time, " > ", 28)
    return 0.28;
  } else {
    //console.log("calulateNettLeieFor", time, " > ", 16)
    return 0.16;
  }

}
```

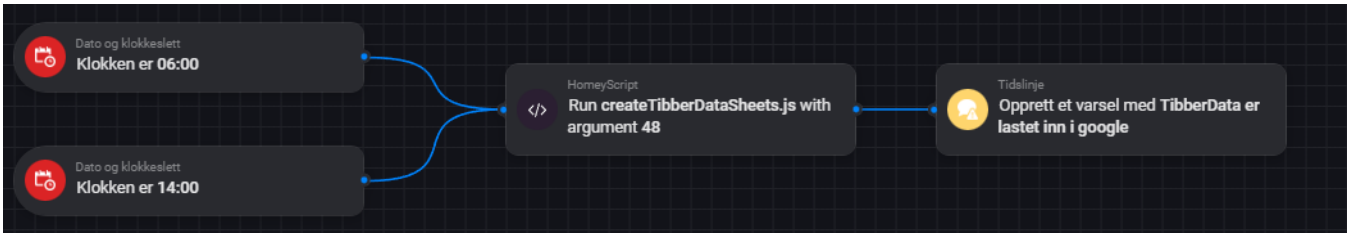
  - the value and hours of day should be adjusted to your local prices and hours

This is how you may run this script. A couple of times each day.

How to determine when to run ?

Tibber will return

If you want to get smaller pieces more often, change the argument to 2 and run every hour.



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',
```

Please note that this is in develop, so comments and other testcode is not removed

```
/*

This script will create a sheet with sheetNamePrefix into your Google Spreadsheet

configured from the Easy Logger App

You must have this App configured and Running

You should correct the calulateNettLeieFor function to reflect your NettLeie

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

version : 03.01.2023.1
version : 09.01.2023.1
 - major change: delimiter is | which is a better value than ;
 - several improvements
 - language support
version : 09.01.2023.2
 - new column : Total cost wo/norwegian support w/fixed price
*/

const locale = "no-NO";
const language = "no"; // or set "en" to get english headings
const sheetNamePrefix = "TibberData-";
const startRow = 5; // use value 5 or higher

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

console.log("args", args);

var countOfHours = parseInt(args)

if (isNaN(countOfHours)) {
  countOfHours = 24; // the default is 24 if argument is not set
}

console.log("countOfHours", countOfHours);

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);
    }
}


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);
    }
}

function calulateNettLeieFor(time) {

  // Glitre Energi sine priser
  if (time < 22 && time > 06) {
    //console.log("calulateNettLeieFor", time, " > ", 28)
    return 0.394;
  } else {
    //console.log("calulateNettLeieFor", time, " > ", 16)
    return 0.274;
  }

}


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

console.log("Getting data from Tibber API");


try {
  const response = await fetch(
    'https://api.tibber.com/v1-beta/gql',
    {
      method: 'POST',
      headers: {
        'Authorization': 'Bearer ' + tibberApiKey,
        'Content-Type': 'application/json'
      },
      body: '{"query":"{viewer{homes{address{address1}consumption(resolution:HOURLY, last: '+countOfHours+')
{nodes{from to cost unitPrice unitPriceVAT consumption consumptionUnit}}}}}"}'
  });

  //console.log("raw-response",response);
  //console.log("response status ",response.status);
```

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

  const body = await response.json();

  //console.log("homes",body.data.viewer.homes);
  //console.log("consumption",body.data.viewer.homes[0].consumption);



  var _from;
  var _consumption;
  var sheet = '';
  for (let i = 0; i < body.data.viewer.homes[0].consumption.nodes.length; i++) {

    _from = body.data.viewer.homes[0].consumption.nodes[i].from;
    _consumption = body.data.viewer.homes[0].consumption.nodes[i].consumption;

    if(_consumption === null)
      _consumption = 0;

    // må legge til en time for at klokka skal bli rett
    // from:  2023-01-01T14:00:00.000+01:00  >>  2023-01-01T14:00:00.000Z
    let d = new Date(_from).addHours(1);

  // starter på 0
    let time_offset = d.getHours();
    let dag_offset = d.getDate();
    let month_offset = d.getMonth()+1;
    if (month_offset < 10) month_offset = '0'+ month_offset;
    let year_offset = d.getFullYear();
    let dag_skew = 24 * (dag_offset - 1 );


    var _sheet =  sheetNamePrefix + month_offset + "-" + year_offset;

    if(_sheet !== sheet) {
      sheet = _sheet;
      await createSheet(sheet); // Blir opprettet om det ikke finnes, ellers feiler silent
      let cellHeating = sheet + "!"+ "A" + (startRow -2);
      if(language==="no")
        await setCell(cellHeating, "Tid for forbruk|Forbruk|Timespris (m/nettleie og avg)|Timespris (m/avg u
/nettleie)|Spotpris (u/mva u/avg)|Total kost (m/forbruk u/nettleie)|Timespris m/støtte og nettleie|Total kost m
/støtte og nettleie|Nettleie|Total kost u/støtte m/nettleie");
      else
        await setCell(cellHeating, "Time for consumption|Consumption|Price per hour (w/fixedprice and tax)
|Price per hour (w/tax wo/fixedprice)|Spotprice (wo/tax)|Total cost (w/consumption wo/fixedprice)|Price per
hour w/norwegian support w/fixedprice|Total cost w/norwegian support w/fixedprice|Fixedprice|Total cost wo
/norwegian support w/fixedprice");

      cellHeating = sheet + "!"+ "A" + (startRow -4);
      if(language==="no")
        await setCell(cellHeating, "Strømstøtte|=(SUM(F6:F800) / count(F6:F800) -0,7)*0,9*1,25");
      else
        await setCell(cellHeating, "Norwegian support|=(SUM(F6:F800) / count(F6:F800) -0,7)*0,9*1,25");

      cellHeating = sheet + "!"+ "A" + (startRow -3);
      if(language==="no")
        await setCell(cellHeating, "Gjennomsnittsverdier||=SUM(D6:D800) / count(D6:D800)|=SUM(E6:E800) / count
(E6:E800)|=SUM(F6:F800) / count(F6:F800)|=SUM(G6:G800) / count(H6:H800)|=SUM(H6:H800) / count(H6:H800)|=SUM(I6:
I800) / count(I6:I800)||=SUM(K6:K800) / count(K6:K800)");
      else
        await setCell(cellHeating, "Average values|=SUM(D6:D800) / count(D6:D800)|=SUM(E6:E800) / count(E6:E800)
|=SUM(F6:F800) / count(F6:F800)|=SUM(G6:G800) / count(H6:H800)|=SUM(H6:H800) / count(H6:H800)|=SUM(I6:I800) /
count(I6:I800)||=SUM(K6:K800) / count(K6:K800)");

    }

    let num = (startRow + dag_skew + dag_offset+ time_offset);
```

```
    let cell = sheet + "!"+ "A" + num;

    let unitPrice = body.data.viewer.homes[0].consumption.nodes[i].unitPrice;
    let spotPrice = (body.data.viewer.homes[0].consumption.nodes[i].unitPrice-0.01)/1.25;

    let nettleie = calulateNettLeieFor(time_offset);

    console.log("from: ",_from, " cons " + _consumption , " d", d);

    let totalCost = body.data.viewer.homes[0].consumption.nodes[i].cost;
    if(totalCost===null)
      totalCost = 0;

    await setCell(cell, d.toLocaleString(locale) +
      "|#" + _consumption +   // forbruk
      "|#" + (unitPrice + nettleie) +   // Pris (m/nettleie)
      "|#" + unitPrice +  // Pris (u/nettleie)
      "|#" + spotPrice +   // Spotpris
      "|#" + totalCost + // total kost (u/nettleie)
      "|=D"+ num +"-$C$1" + // pris m/støtte og nettleie
      "|=H"+ num + "*C"+num + // Total kost m/støtte og nettleie
      "|#" + nettleie +  // Nettleie
      "|#=G" + num + "+J" + num   // Total kost u/støtte m/nettleie
      ); // # tvinger tallformat



  }




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


console.log("DONE");
```