

loadTibberPrices.js

```
/*

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 effective execution of Google API calls.
                  - settable delimiter
- 23.01.2023.1 - using tibberApiKey from global variables
- 19.06.2023.1 - Parsing dates from Tibbes as localDateTime, will probably fixe timesone and summertime issues

*/

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 ;
const tibberApiKey = global.get('tibberApiKey'); // you need to execute the script her before use: https://wiki.bwa.no/display/HEL/setTibberApiKey.js
const locale = "no-NO"; // currently not implementet yet

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 logging is needed
  //console.log(body.data.viewer);
  //console.log(body.data.viewer.homes[0]);

  let cellHeating = sheetName + "!" + "A" + (startRow -3);
  await setCell(cellHeating, "Today's 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);

    // substring(0,19) creates '2023-01-01T14:00:00' from 2023-01-01T14:00:00.000+01:00
    let localDateTimeString =body.data.viewer.homes[0].currentSubscription.priceInfo.today[i].startsAt.substring(0,19);
    let d = new Date(localDateTimeString);

    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 += " ";

/* TOMORROW */

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

let dataSize = body.data.viewer.homes[0].currentSubscription.priceInfo.tomorrow.length;
for (let i = 0; i < dataSize; 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

        // substring(0,19) creates '2023-01-01T14:00:00' from 2023-01-01T14:00:00.000+01:00
        let localDateTimeString = body.data.viewer.homes[0].currentSubscription.priceInfo.tomorrow[i].startsAt.
substring(0,19);
        let d = new Date(localDateTimeString);

        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 < (dataSize -1)) {
        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)
}
```