Table of Contents:

1. Summary

This guide describes the installation and operation of the connect-web-logger app on Windows 10 PCs. Tests have shown the Google Chrome browser to be the best choice for scraping information from the Fröling website. Tests have also proven headless mode to be very unreliable (both for Chrome and Firefox) on Ubuntu.

If you need reliable long term measurements then a Windows PC or MacBook is the best way to go.

2. Prerequistes

- PC with Windows 10 installed (latest version)
- Connection to the Internet (fast connection preferred)
- Fröling PE1 pellet heating
- Valid account at connect-web.froeling.com

Note:
> designates a command line (cmd)
# designates a comment line
... designates a response

3. Installation Guide

3.1 Google Chrome

Install or update the latest stable release of the Chrome browser. For this guide we used version 87.0.4280.
Note:

- If you update or reinstall the Google Browser, some of the dependant modules (e.g. Selenium) can be deinstalled / removed.
- The connect-web-logger will no longer run as advertised.

Now we need to install **chromedriver**, the interface between the Chrome browser and Selenium. Download the proper version (here 87.0.4280) from the chromium website https://chromedriver.chromium.org.

- Copy the downloaded zip file to the desktop and unzip it.
- Make a new folder **WebDriver** in C: and add a subfolder **bin**.
- Copy chromedriver.exe to folder C:\WebDriver\bin\chromedriver.exe
- These file and folders are referenced in the application, the names are mandatory.

3.2 Python

Install Python from the Microsoft Store (free). For this guide we used version 3.9.1.

3.3 Python Dependencies

Open the command line tool (CMD) and install the following dependencies. Some may already be present, this will be reported:

```
> pip --version
20.2.3
> pip install --upgrade pip
...
> pip --version
20.3.3
> pip install setuptools
...
> pip install selenium
...
> pip install phantomjs
...
> pip install schedule
...
> pip install matplotlib
...
```

3.4 Git

Download Git (Windows version) from git-scm.com and install.
For this guide we used version 2.30.0.windows.1.

3.5 connect-web-logger

The application resides on github.com and is downloaded as follows.
Open the command line tool (CMD):

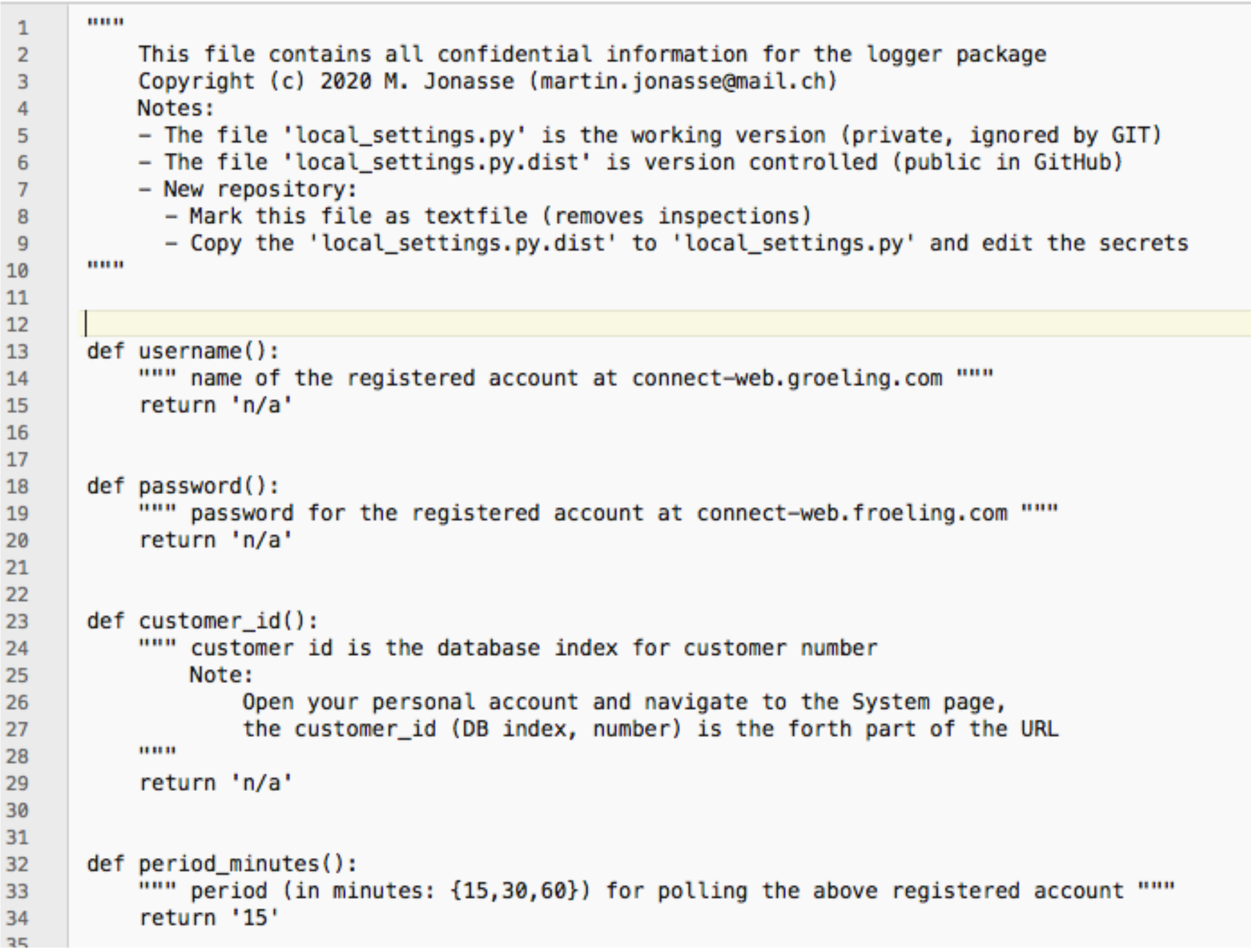
```
# set current directory (folder)
> D:
> cd myprojects

# clone repository to subfolder connect-web-logger
> git clone https://github.com/majo48/connect-web-logger.git
...
> cd connect-web-logger
> mkdir database
> cd logger
> copy local_settings.py.dist local_settings.py
```

3.6 Add confidential information

Open the Windows Explorer and navigate to file local_setting.py. Edit the file with a text editor and change 'n/a' to a valid text (username, password and customer_id).

The value for period_minutes is preset to 15 minutes, but can be changed to 30 or 60 minutes. Any other values are not supported by the application and cause Exceptions.

If you are using an IDE, you might need to mark the local_settings.py.dist file as a text file (due to inspection error messages).



```
"""
    This file contains all confidential information for the logger package
    Copyright (c) 2020 M. Jonasse (martin.jonasse@mail.ch)
    Notes:
    - The file 'local_settings.py' is the working version (private, ignored by GIT)
    - The file 'local_settings.py.dist' is version controlled (public in GitHub)
    - New repository:
      - Mark this file as textfile (removes inspections)
      - Copy the 'local_settings.py.dist' to 'local_settings.py' and edit the secrets
"""


def username():
    """ name of the registered account at connect-web.groeling.com """
    return 'n/a'


def password():
    """ password for the registered account at connect-web.froeling.com """
    return 'n/a'


def customer_id():
    """ customer id is the database index for customer number
        Note:
            Open your personal account and navigate to the System page,
            the customer_id (DB index, number) is the forth part of the URL
    """
    return 'n/a'


def period_minutes():
    """ period (in minutes: {15,30,60}) for polling the above registered account """
    return '15'
```

Congratulations, you are now ready to run.

4. Run Scripts

The application provides some scripts used for testing and some python modules which represent the application as such:

- the **logger**: captures data from the Fröling website into a database
- the **plotter**: displays captured data from the database to multiple plots
- the **database**: contains the captured data for custom queries

4.1 Run testChrome.py

This is a simple script which opens the python.org website and prints the page title. This uses practically all the components installed previously.

4.1.1 Windows

- Open Windows CMD
- Navigate to D:\...\connect-web-logger folder
- Navigate to scripts subfolder
- Enter: **py scripts/testChrome.py**

```
Test Selenium & Chrome...
Title: Welcome to Python.org
```

Please note:

- This Windows version of Chrome produces many error messages complaining about the "usb_service_win". This is a bug in Chrome and should be fixed in version 90.
- Reference: https://stackoverflow.com/questions/65080685/usb-usb-device-handle-win-cc1020-failed-to-read-descriptor-from-node-connectio

4.1.2 MacOS

- Open Terminal
- Navigate (cd) to connect-web-logger folder
- Enter: **python3 scripts/testChrome.py**

```
(venv) macbook:connect-web-logger mart$ python3 scripts/testChrome.py
Test Selenium & Chrome...
Title: Welcome to Python.org
(venv) macbook:connect-web-logger mart$
```

- Note: (venv) stands for virtual environment of the IDE

4.2 Run logger module

The logger module runs periodically (see local_settings.py), scrapes information from the account at the Fröling website and stores the data in a SQLite database.

The logger app has zero, one or three arguments:

- **py -m logger** # runs periodically, using attributes from local_settings.py
- **py -m logger unittest** # runs once, using attributes from local_settings.py
- **py -m logger <username> <password> <period_minutes>** # runs periodically

4.2.1 Windows:

- Open Windows CMD
- Navigate to D:\...\connect-web-logger folder
- Enter: **py -m logger**
  Each output line starts with a timestamp >>> and information.
  Errors are designated with timestamp >>> Error: ...

Please note:

- This Windows version of Chrome produces many error messages complaining about the "usb_service_win". This is a bug in Chrome and should be fixed in version 90.
- Reference:
  https://stackoverflow.com/questions/65080685/usb-usb-device-handle-win-cc1020-failed-to-read-descriptor-from-node-connectio

4.2.2 MacOS:

- Open Terminal
- Navigate (cd) to connect-web-logger folder
- Enter: **python3 -m logger**
  Each output line starts with a timestamp >>> and information.
  Errors are designated with timestamp >>> Error: ...

4.3 Run plotter module

The plotter module has been build to provide some relevant visual data from the SQLite database.

The plotter app has none (from the beginning) or one argument:

- from_date: plot from yyyy-mm-dd hh:mm:ss
- to_date: plot including yyyy-mm-dd hh:mm:ss
- None: plot from the beginning to the end of the database

Windows:

- Open Windows CMD
- Navigate to D:\...\connect-web-logger folder
- Enter: **py -m plotter "2021-01-06 12:00:00" "2021-01-07 12:00:00"**

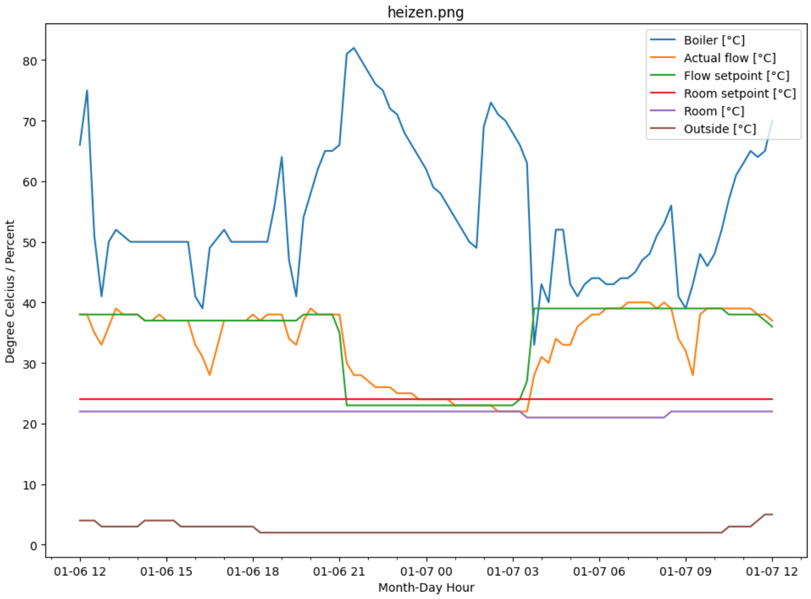
```
First timestamp: 2021-01-06 12:00:00
Last timestamp: 2021-01-07 12:00:00
Created file: rauchgas.png
Created file: heizen.png
Created file: warmwasser.png
```

MacOS:

- Open Terminal
- Navigate (cd) to connect-web-logger folder
- Enter: **python3 -m plotter "2021-01-06 12:00:00" "2021-01-07 12:00:00"**

```
First timestamp: 2021-01-06 12:00:00
Last timestamp: 2021-01-07 12:00:00
Created file: rauchgas.png
Created file: heizen.png
Created file: warmwasser.png
```

Example:


heizen.png
Boiler [°C]
Actual flow [°C]
Flow setpoint [°C]
Room setpoint [°C]
Room [°C]
Outside [°C]
Degree Celcius / Percent
Month-Day Hour