

cart_tracker

by [Dominique Garmier](#) (c) 2020

App developed for the Kantonsspital Aarau, to track the status of laundry carts.

The app is meant to be used by the employees to check-off certain carts as "Done" once they are loaded. The information about all the carts is then written to a remote CSV file which can be read in real-time by an excel sheet.

How to install

[cart_tracker](#) only works for Windows (10). It can however be adapted for unix use. Namely by replacing the *.bat and *.vbs files with their respective unix counterparts.

First off [Python 3.8](#) or later needs to be installed alongside the required packages. Using [pip](#) you can install the required packages the following way:

```
$ pip install -r requirements.txt
```

It is also advised that you add [Python](#) to your PATH. Else you would need to adjust all the *.bat and *.vbs files. Since in their current state, they rely [Python](#) being in PATH.

If there is no [config.ini](#) file, run the [main.py](#) script once to generate it. It will close down immediately.

```
$ python main.py  
config.ini generated!
```

Next take a look at the [config.ini](#) file:

```
# config.ini  
  
[DEFAULT]  
data_path = ./data.csv  
debug = False  
  
[ExcelToPdf]  
pdf_folder_path = path/to/folder/  
excel_file_path = path/to/fild.xlsx  
excel_sheets = [1, 2]  
pdf_names = ['sheet-1', 'sheet-2']
```

- The section [\[DEFAULT\]](#) contains all the settings needed for the main application i.e. the the gui application.

- The option `data_path` is to specify where you want to store the raw data. In our intended usecase this would be a remote map drive.
- The `debug` option makes python print out entries and some other minor messages.
- The section `[ExcelToPdf]` contains all settings for the script to export the excel files automatically once a day.
 - `pdf_folder_path` specifies the folder in which the *.pdf files will be saved every day
 - `excel_file_path` sets the path to the *.xlsx file.
 - `excel_sheets` is a list of indices that specifies which sheet of the *.xlsx file should be exported.
 - `pdf_names` is a list of strings defining how those exported *.pdf files should be called. Each sheet having a different name.

For this specific usecase we want our *.csv file to be cleared once per day to reset the states of all the carts. for this there exists the `daily_task.bat` which intern runs the `daily_task.py` file. Here we can use the built in windows Task-Scheduler, or alternatively on unix a cron task, to trigger the *.bat file once per day.

Now finally create a shortcut of the `cart_tracker.vbs`. This will be primary way to start the programm.

How to use

Click on the shortcut you just created, a window will appear prompting you to enter cart numbers. You can use the autocomplete search to find valid cart names. You can select a cart name from the dropdown by either navigating to it using the arrow keys and then hitting the return or by double clicking it with your mouse. The Selected cart names will show up at the top. If you change your mind and afterall dont want to add a certain cart you can hit the X button next to the cart name to remove it.

To navigate between textfields hit the return button. Depending on the context you may need to hit the return button more than once (to select the entry in the autocomplete and then to skip ahead)

The second textfield asks you to provide a signature, type something like the first letters of your first and last name. It's to later on identify who loaded that cart.

After hitting return again you will be asked to confirm your entries. If you do so they will be saved in a specified CSV file where they can be read in real time by an Excel spreadsheet.

If you want to remove an entry you falsely entered, you can hit the "Korrektur" Button at the top left. This will open a new Page where you only have to enter the cart name you want to remove. This Textfield also has an autocomplete feature, only autocompleting to cart names already in the entries (else it works the same as the first autocomplete entry). After hitting enter the CSV file will be updated.

How to update

`cart_tracker` can be updated using `git`. If the repo was cloned as is, you can simply run `git pull`. The config file and the `cart_names.txt` file are both in the `.gitignore`. This can also be done using the `update.py` script, which will automatically try to pull from master. Simply run:

```
$ python udate.py
```