PyGCE

API Documentation

April 17, 2017

Contents

1 Package pygce

1.1 Modules

- **analysis** *(Section 2, p. 4)*
  - **cli** *(Section 3, p. 5)*
  - **models** *(Section 4, p. 6)*
- **cli** *(Section 5, p. 12)*
- **models** *(Section 6, p. 14)*
  - **bot** *(Section 7, p. 15)*
  - **garmin** *(Section 8, p. 18)*
    - **activities** *(Section 9, p. 19)*
    - **timeline** *(Section 10, p. 20)*
    - **utils** *(Section 11, p. 33)*

1.2 Variables

Name	Description
__package__	**Value:** `None`

2 Package pygce.analysis

2.1 Modules

- **cli** *(Section 3, p. 5)*
- **models** *(Section 4, p. 6)*

2.2 Variables

Name	Description
__package__	**Value:** `None`

3 Module pygce.analysis.cli

3.1 Functions

create_args()

```
:return: ArgumentParser
    Parser that handles cmd arguments.
```

parse_args(*parser*)

```
:param parser: ArgumentParser
    Object that holds cmd arguments.
:return: tuple
    Values of arguments.
```

check_args(*folder_path*)

```
:param folder_path: str
    Path to folder with data files to analyse
```

main()

4 Module pygce.analysis.models

4.1 Class GarminDataFilter

object

pygce.analysis.models.GarminDataFilter

```
Parses and fixes raw data
```

4.1.1 Methods

__init__(*self*, *dataset_file*)

```
:param dataset_file: str
    Path to folder with data to analyse
```

Overrides: object.__init__

parse_csv(*self*)

```
:return: tuple [], [] of []
    Headers of csv file and data
```

convert_time_columns(*headers*, *headers_to_convert*, *data*)

```
:param headers: [] of str
    Column names of data
:param headers_to_convert: [] of str
    Column names of data to convert from time format to float
:param data: [] of []
    Raw data
:return: [] of []
    Input data but with converted time columns
```

fix_floats(*headers*, *headers_to_fix*, *data*)

```
:param headers: [] of str
    Column names of data
:param headers_to_fix: [] of str
    Column names of data to fix the float format
:param data: [] of []
    Raw data
:return: [] of []
    Input data but with fixed floats in columns
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

4.1.2 Properties

Name	Description
Inherited from object	
__class__	

4.2 Class StatsAnalysis

object

pygce.analysis.models.GarminDataFilter

pygce.analysis.models.StatsAnalysis

```
Computes correlation of data
```

4.2.1 Methods

__init__(*self*, *dataset_file*)

```
:param dataset_file: str
    Path to folder with data to analyse
```

Overrides: object.__init__

show_correlation_matrix(*self*, *title_image*, *headers_to_analyze*)

```
:param title_image: str
    Title of output image
:param headers_to_analyze: [] of str
    Compute correlation matrix of only these headers
:return: void
    Shows correlation matrix of data of files in folder
```

Inherited from pygce.analysis.models.GarminDataFilter(Section 4.1)

convert_time_columns(), fix_floats(), parse_csv()

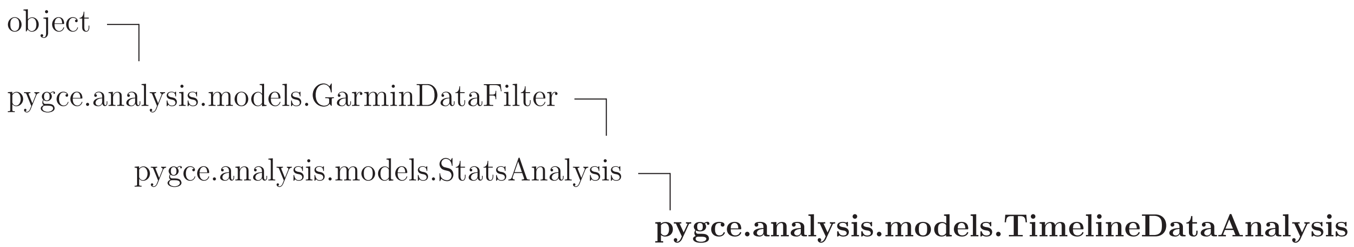
Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

4.2.2 Properties

Name	Description
Inherited from object	
__class__	

4.3 Class TimelineDataAnalysis



object

pygce.analysis.models.GarminDataFilter

pygce.analysis.models.StatsAnalysis

pygce.analysis.models.TimelineDataAnalysis

```
Machine-learn timeline data
```

4.3.1 Methods

__init__(*self*, *dataset_file*)

```
:param dataset_file: str
    Path to folder with data to analyse
```

Overrides: object.__init__

parse_csv(*self*)

```
:return: tuple [], [] of []
    Headers of csv file and data
```

Overrides: pygce.analysis.models.GarminDataFilter.parse_csv

show_correlation_matrix_of_data(*self*)

```
:return: void
    Shows correlation matrix of data of files in folder
```

predict_feature(*self*, *feature*)

```
:param feature: str
    Name of feature (column name) to predict
:return: TODO
    TODO
```

cluster_analyze(*self*)

```
:return: void
    Computes cluster analysis: see days based on differences.
    Each day is different from one another, there are days where you trained more, o
    The goal is to divide your days into categories (e.g highly-active, active ...)
```

show_bar_chart(*title*, *x_labels*, *y_values*, *y_label*)

```
:param title: str
    Title of chart
:param x_labels: [] of str
    Names for each variable
:param y_values: [] of float
    Values of x labels
:param y_label: str
    Label of y axis
:return: void
    Show bar chart
```

Inherited from pygce.analysis.models.StatsAnalysis(Section 4.2)

show_correlation_matrix()

Inherited from pygce.analysis.models.GarminDataFilter(Section 4.1)

convert_time_columns(), fix_floats()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

4.3.2 Properties

Name	Description
Inherited from object __class__	

4.3.3 Class Variables

Name	Description
HEADERS_TO_ANALYZE	**Value:** `["SUMMARY:kcal_count", "STEPS:distance", "SLEEP:light_sle...`
TIME_HEADERS_TO_CONVERT	**Value:** `["SLEEP:nap_time", "SLEEP:light_sleep_time", "SLEEP:awake...`

4.4 Class ActivitiesDataAnalysis

object

pygce.analysis.models.GarminDataFilter

pygce.analysis.models.StatsAnalysis

pygce.analysis.models.ActivitiesDataAnalysis

```
Machine-learn activities data
```

4.4.1 Methods

__init__(*self*, *dataset_file*)

```
:param dataset_file: str
    Path to folder with data to analyse
```

Overrides: object.__init__

parse_csv(*self*)

```
:return: tuple [], [] of []
    Headers of csv file and data
```

Overrides: pygce.analysis.models.GarminDataFilter.parse_csv

shows_correlation_matrix_of_data(*self*)

```
:return: void
    Shows correlation matrix of data of files in folder
```

Inherited from pygce.analysis.models.StatsAnalysis(Section 4.2)

show_correlation_matrix()

Inherited from pygce.analysis.models.GarminDataFilter(Section 4.1)

convert_time_columns(), fix_floats()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

4.4.2 Properties

Name	Description
Inherited from object __class__	

4.4.3 Class Variables

Name	Description
HEADERS_TO_ANALYZE	**Value:** `["Distance", "Time", "Avg Speed(Avg Pace)", "Max Speed(Be...`
TIME_HEADERS_TO_CONVERT	**Value:** `["Time", "Avg Speed(Avg Pace)", "Max Speed(Best Pace)"]`
HEADERS_WITH_MALFORMED_FLOATS	**Value:** `["Distance", "Training Effect"]`

5 Module pygce.cli

5.1 Functions

parse_yyyy_mm_dd(*d*)

```
:param d: str
    Date in the form yyyy-mm-dd to parse
:return: datetime
    Date parsed
```

create_args()

```
:return: ArgumentParser
    Parser that handles cmd arguments.
```

parse_args(*parser*)

```
:param parser: ArgumentParser
    Object that holds cmd arguments.
:return: tuple
    Values of arguments.
```

check_args(*user*, *password*, *chromedriver*, *days*, *format_out*, *path_out*)

```
:param user: str
    User to use
:param password: str
    Password to use
:param chromedriver: str
    Path to chromedriver to use
:param days: [] of datetime.date
    Days to save
:param format_out: str
    Format of output file (json, csv)
:param path_out: str
    File to use as output
:return: bool
    True iff args are correct
```

main()

5.2 Variables

Name	Description
AVAILABLE_OUTPUT_FORMATS	**Value:** `["json", "csv"]`

6 Package pygce.models

6.1 Modules

- **bot** *(Section 7, p. 15)*
- **garmin** *(Section 8, p. 18)*
  - **activities** *(Section 9, p. 19)*
  - **timeline** *(Section 10, p. 20)*
  - **utils** *(Section 11, p. 33)*

6.2 Variables

Name	Description
__package__	**Value:** `None`

7 Module pygce.models.bot

7.1 Class GarminConnectBot

object ─┐
pygce.models.bot.GarminConnectBot

```
Navigate through Garmin Connect app via a bot
```

7.1.1 Methods

__init__(*self*, *user_name*, *password*, *chromedriver_path*)

```
:param user_name: str
    Username (email) to login to Garmin Connect
:param password: str
    Password to login to Garmin Connect
:param chromedriver_path: str
    Path to Chrome driver to use as browser
```

Overrides: object.__init__

login(*self*)

```
:return: bool
    True iff correctly logged in
```

get_user_id(*self*)

```
:return: void
    Retrieves user unique id and token
```

go_to_dashboard(*self*)

```
:return: void
    Navigates to user homepage
```

go_to_day(*self*, *date_time*)

```
:param date_time: datetime
    Datetime object with date
:return: void
    Navigates to daily summary of given date
```

get_day(*self*, *date_time*)

```
:param date_time: datetime
    Datetime object with date
:return: GCDayTimline
    Data about day
```

get_days(*self*, *min_date_time*, *max_date_time*)

```
:param min_date_time: datetime
    Datetime object with date, this is the date when to start downloading data
:param max_date_time: datetime
    Datetime object with date, this is the date when to stop downloading data
:return: [] of GCDayTimline
    List of data about days
```

save_json_days(*self*, *min_date_time*, *max_date_time*, *output_file*)

```
:param min_date_time: datetime
    Datetime object with date, this is the date when to start downloading data
:param max_date_time: datetime
    Datetime object with date, this is the date when to stop downloading data
:param output_file: str
    Path where to save output to
:return: void
    Retrieves data about days in given range, then saves json dump
```

save_csv_days(*self*, *min_date_time*, *max_date_time*, *output_file*)

```
:param min_date_time: datetime
    Datetime object with date, this is the date when to start downloading data
:param max_date_time: datetime
    Datetime object with date, this is the date when to stop downloading data
:param output_file: str
    Path where to save output to
:return: void
    Retrieves data about days in given range, then saves csv dump
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

7.1.2 Properties

Name	Description
Inherited from object __class__	

7.1.3 Class Variables

Name	Description
USER_DASHBOARD	**Value:** `"https://connect.garmin.com/modern/"`
LOGIN_URL	**Value:** `"https://sso.garmin.com/sso/login?service=https%3A%2F%2Fc.`
LOGIN_BUTTON_ID	**Value:** `"login-btn-signin"`
USERNAME_FIELD_NAME	**Value:** `"username"`
PASSWORD_FIELD_NAME	**Value:** `"password"`
BROWSER_WAIT_TIMEOUT_SECONDS	**Value:** `5`

8 Package pygce.models.garmin

8.1 Modules

- **activities** *(Section 9, p. 19)*
- **timeline** *(Section 10, p. 20)*
- **utils** *(Section 11, p. 33)*

8.2 Variables

Name	Description
__package__	**Value:** `None`

9 Module pygce.models.garmin.activities

9.1 Variables

Name	Description
__package__	**Value:** `None`

10 Module pygce.models.garmin.timeline

10.1 Variables

Name	Description
__package__	**Value:** `'pygce.models.garmin'`

10.2 Class GCDaySection

object ─┐

pygce.models.garmin.timeline.GCDaySection

Known Subclasses: pygce.models.garmin.timeline.GCDayActivities, pygce.models.garmin.timeline.GCD
pygce.models.garmin.timeline.GCDaySleep, pygce.models.garmin.timeline.GCDaySteps, pygce.models.garm

```
Standard section in the Garmin Connect timeline of day.
```

10.2.1 Methods

__init__(*self*, *raw_html*, *tag=*`''`)

```


:param raw_html: str
    HTML source snippet with information about section
:param tag: str
    Unique str in order not to mistake this GCDaySection with another one
```

Overrides: object.__init__

parse(*self*)

```


:return: void
    Parses raw html source and tries to finds all information
```

to_dict(*self*)

```
:return: dict
    Dictionary with keys (obj fields) and values (obj values)
```

to_json(*self*)

```
:return: json object
    A json representation of this object
```

to_csv_dict(*self*)

```
:return: {}
    Like self.to_json() but with a unique str before each key to spot against differ
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

10.2.2 Properties

Name	Description
Inherited from object	
__class__	

10.3 Class GCDaySummary

object
pygce.models.garmin.timeline.GCDaySection
pygce.models.garmin.timeline.GCDaySummary

```
Standard activity in the Garmin Connect timeline of day.
Common features are likes, comment, kcal
```

10.3.1 Methods

__init__(*self*, *raw_html*)

```
:param raw_html: str
    HTML source snippet with information about section
```

Overrides: object.__init__

parse(*self*)

```
:return: void
    Parses raw html source and tries to finds all information
```

Overrides: pygce.models.garmin.timeline.GCDaySection.parse extit(inherited documentation)

parse_likes(*self*)

```
:return: void
    Finds likes count and stores value
```

parse_comment(*self*)

```
:return: void
    Finds comment value and stores value
```

parse_kcal_count(*self*)

```
:return: void
    Finds kcal value and stores value
```

to_dict(*self*)

```
:return: dict
    Dictionary with keys (obj fields) and values (obj values)
```

Overrides: pygce.models.garmin.timeline.GCDaySection.to_dict extit(inherited documentation)

Inherited from pygce.models.garmin.timeline.GCDaySection(Section 10.2)

to_csv_dict(), to_json()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

10.3.2 Properties

Name	Description
Inherited from object __class__	

10.4 Class GCDaySteps

object

pygce.models.garmin.timeline.GCDaySection

pygce.models.garmin.timeline.GCDaySteps

```
Standard activity in the Garmin Connect timeline of day.
Common features are total, goal, distance, avg daily
```

10.4.1 Methods

__init__(*self*, *raw_html*)

```
:param raw_html: str
    HTML source snippet with information about section
```

Overrides: object.__init__

parse(*self*)

```
:return: void
    Parses raw html source and tries to finds all information
```

Overrides: pygce.models.garmin.timeline.GCDaySection.parse extit(inherited documentation)

parse_steps_count(*self*)

```
:return: void
    Parses HTML source and finds goal and daily steps
```

parse_steps_stats(*self*)

```
:return: void
    Parses HTML source and finds daily distance and avg daily steps
```

to_dict(*self*)

```
:return: dict
    Dictionary with keys (obj fields) and values (obj values)
```

Overrides: pygce.models.garmin.timeline.GCDaySection.to_dict extit(inherited documentation)

Inherited from pygce.models.garmin.timeline.GCDaySection(Section 10.2)

to_csv_dict(), to_json()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

10.4.2 Properties

Name	Description
Inherited from object __class__	

10.5 Class GCDaySleep

object

pygce.models.garmin.timeline.GCDaySection

pygce.models.garmin.timeline.GCDaySleep

```
Standard activity in the Garmin Connect timeline of day.
Common features are total, deep total, light total, awake total
```

10.5.1 Methods

__init__(*self*, *raw_html*)

```
:param raw_html: str
    HTML source snippet with information about section
```

Overrides: object.__init__

parse(*self*)

```
:return: void
    Parses raw html source and tries to finds all information
```

Overrides: pygce.models.garmin.timeline.GCDaySection.parse extit(inherited documentation)

parse_sleep_totals(*self*)

```
:return: void
    Finds value of night/nap/total sleep times
```

parse_bed_time(*self*)

```
:return: void
    Finds hour start/end sleep
```

parse_sleep_times(*self*)

```
:return: void
    Finds deep/light/awake sleep times
```

to_dict(*self*)

```
:return: dict
    Dictionary with keys (obj fields) and values (obj values)
```

Overrides: pygce.models.garmin.timeline.GCDaySection.to_dict extit(inherited documentation)

Inherited from pygce.models.garmin.timeline.GCDaySection(Section 10.2)

to_csv_dict(), to_json()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

10.5.2 Properties

Name	Description
Inherited from object __class__	

10.6 Class GCDayActivities

object

pygce.models.garmin.timeline.GCDaySection

pygce.models.garmin.timeline.GCDayActivities

```
Standard activity in the Garmin Connect timeline of day.
Common features are kcal, time, distance, type, name, link
```

10.6.1 Methods

__init__(*self*, *raw_html*)

```
:param raw_html: str
    HTML source snippet with information about section
```

Overrides: object.__init__

parse(*self*)

```
:return: void
    Parses raw html source and tries to finds all information
```

Overrides: pygce.models.garmin.timeline.GCDaySection.parse extit(inherited documentation)

parse_activity(*raw_html*)

```
:param raw_html: str html code
    Raw HTML code of row of table containing activity to parse
:return: dict
    Dict with values of activity
```

to_dict(*self*)

```
:return: dict
    Dictionary with keys (obj fields) and values (obj values)
```

Overrides: pygce.models.garmin.timeline.GCDaySection.to_dict extit(inherited documentation)

to_json(*self*)

```
:return: json object
    A json representation of this object
```

Overrides: pygce.models.garmin.timeline.GCDaySection.to_json extit(inherited documentation)

to_csv_dict(*self*)

```
:return: {}
    Like super.to_csv_dict() but with totals instead
```

Overrides: pygce.models.garmin.timeline.GCDaySection.to_csv_dict

get_total_kcal(*self*)

```
:return: float
    Total kcal of all activities
```

get_total_duration(*self*)

```
:return: timedelta
    Total duration of all activities
```

get_total_distance(*self*)

```
:return: float
    Total distance of all activities
```

get_totals_dict(*self*)

```
:return: {}
    Self dict but with totals instead (total kcal, total distance ...)
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

10.6.2 Properties

Name	Description
Inherited from object __class__	

10.7 Class GCDayBreakdown

object

pygce.models.garmin.timeline.GCDaySection

pygce.models.garmin.timeline.GCDayBreakdown

```
Standard activity in the Garmin Connect timeline of day.
Common features are highly active %, active %, sedentary %, sleep %
```

10.7.1 Methods

__init__(*self*, *raw_html*)

```
:param raw_html: str
    HTML source snippet with information about section
```

Overrides: object.__init__

parse(*self*)

```
:return: void
    Parses raw html source and tries to finds all information
```

Overrides: pygce.models.garmin.timeline.GCDaySection.parse extit(inherited documentation)

to_dict(*self*)

```
:return: dict
    Dictionary with keys (obj fields) and values (obj values)
```

Overrides: pygce.models.garmin.timeline.GCDaySection.to_dict extit(inherited documentation)

Inherited from pygce.models.garmin.timeline.GCDaySection(Section 10.2)

to_csv_dict(), to_json()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

10.7.2 Properties

Name	Description
Inherited from object	
__class__	

10.8 Class GCDayTimeline

object

pygce.models.garmin.timeline.GCDayTimeline

```
Standard Garmin Connect timeline of day as in webpage.
Each standard day consists of different sections:
- summary (day, likes, comment, kcal)
- steps (total, goal, distance, avg daily)
- sleep (total, deep total, light total, awake total)
- activities (for each one: kcal, time, distance, type, name, link)
- breakdown (highly active %, active %, sedentary %, sleep %)
```

10.8.1 Methods

__init__(*self*, *date_time*, *summary_html*, *steps_section_html*, *sleep_section_html*, *activities_section_html*, *breakdown_section_html*)

```
:param date_time: datetime
    Datetime of day
:param summary_html: str
    HTML source snippet with information about the day
:param steps_section_html: str
    HTML source snippet with information about daily steps
:param sleep_section_html: str
    HTML source snippet with information about daily sleep
:param activities_section_html: str
    HTML source snippet with information about daily activities
:param breakdown_section_html: str
    HTML source snippet with information about daily breakdown
```

Overrides: object.__init__

parse(*self*)

```
:return: void
    Finds all sections to parse, then builds corresponding objects and parses everyt
```

__getattr__(*self*, *item*)

to_dict(*self*)

```
:return: dict
    Dictionary with keys (obj fields) and values (obj values)
```

to_csv_dict(*self*)

```
:return: {}
    Like self.to_dict() but with a set with keys and values NOT nested. Also for act
```

to_json(*self*)

```
:return: json object
    A json representation of this object
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

10.8.2 Properties

Name	Description
Inherited from object __class__	

11 Module pygce.models.garmin.utils

11.1 Functions

parse_num(*n*)

```
:param n: str
    Number to parse
:return: float
    Parses numbers written like 123,949.99
```

parse_hh_mm_ss(*h*)

```
:param h: str
    Hours, minutes and seconds in the form hh:mm:ss to parse
:return: datetime.time
    Time parsed
```

get_seconds(*s*)

```
:param s: str
    Datetime in the form %H:%M:%S
:return: int
    Seconds in time
```

parse_hh_mm(*h*)

```
:param h: str
    Hours and minutes in the form hh:mm to parse
:return: datetime.time
    Time parsed
```

11.2 Variables

Name	Description
GARMIN_CONNECT_URL	**Value:** ’https://connect.garmin.com’

continued on next page

Name	Description
GARMIN_CONNECT_ACTIVITIES_URL	**Value:** `'https://connect.garmin.com/modern/activities'`
__package__	**Value:** `'pygce.models.garmin'`

Index