

# PyGCE

## API Documentation

April 17, 2017

## Contents

<b>Contents</b>	<b>1</b>
<b>1 Package pygce</b>	<b>3</b>
1.1 Modules . . . . .	3
1.2 Variables . . . . .	3
<b>2 Package pygce.analysis</b>	<b>4</b>
2.1 Modules . . . . .	4
2.2 Variables . . . . .	4
<b>3 Module pygce.analysis.cli</b>	<b>5</b>
3.1 Functions . . . . .	5
<b>4 Module pygce.analysis.models</b>	<b>6</b>
4.1 Class GarminDataFilter . . . . .	6
4.1.1 Methods . . . . .	6
4.1.2 Properties . . . . .	7
4.2 Class StatsAnalysis . . . . .	7
4.2.1 Methods . . . . .	7
4.2.2 Properties . . . . .	8
4.3 Class MLAnalysis . . . . .	8
4.3.1 Methods . . . . .	8
4.3.2 Properties . . . . .	8
4.4 Class TimelineDataAnalysis . . . . .	9
4.4.1 Methods . . . . .	9
4.4.2 Properties . . . . .	10
4.4.3 Class Variables . . . . .	11
4.5 Class ActivitiesDataAnalysis . . . . .	11
4.5.1 Methods . . . . .	11
4.5.2 Properties . . . . .	12
4.5.3 Class Variables . . . . .	12
<b>5 Module pygce.cli</b>	<b>13</b>
5.1 Functions . . . . .	13
5.2 Variables . . . . .	14
<b>6 Package pygce.models</b>	<b>15</b>
6.1 Modules . . . . .	15

6.2	Variables . . . . .	15
<b>7</b>	<b>Module pygce.models.bot</b>	<b>16</b>
7.1	Class GarminConnectBot . . . . .	16
7.1.1	Methods . . . . .	16
7.1.2	Properties . . . . .	18
7.1.3	Class Variables . . . . .	18
<b>8</b>	<b>Package pygce.models.garmin</b>	<b>19</b>
8.1	Modules . . . . .	19
8.2	Variables . . . . .	19
<b>9</b>	<b>Module pygce.models.garmin.activities</b>	<b>20</b>
9.1	Variables . . . . .	20
<b>10</b>	<b>Module pygce.models.garmin.timeline</b>	<b>21</b>
10.1	Variables . . . . .	21
10.2	Class GCDaySection . . . . .	21
10.2.1	Methods . . . . .	21
10.2.2	Properties . . . . .	22
10.3	Class GCDaySummary . . . . .	22
10.3.1	Methods . . . . .	23
10.3.2	Properties . . . . .	24
10.4	Class GCDaySteps . . . . .	24
10.4.1	Methods . . . . .	24
10.4.2	Properties . . . . .	25
10.5	Class GCDaySleep . . . . .	26
10.5.1	Methods . . . . .	26
10.5.2	Properties . . . . .	27
10.6	Class GCDayActivities . . . . .	27
10.6.1	Methods . . . . .	28
10.6.2	Properties . . . . .	29
10.7	Class GCDayBreakdown . . . . .	30
10.7.1	Methods . . . . .	30
10.7.2	Properties . . . . .	31
10.8	Class GCDayTimeline . . . . .	31
10.8.1	Methods . . . . .	32
10.8.2	Properties . . . . .	33
<b>11</b>	<b>Module pygce.models.garmin.utils</b>	<b>34</b>
11.1	Functions . . . . .	34
11.2	Variables . . . . .	34
<b>Index</b>		<b>36</b>

# 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. 13*)
- **models** (*Section 6, p. 15*)
  - **bot** (*Section 7, p. 16*)
  - **garmin** (*Section 8, p. 19*)
    - \* **activities** (*Section 9, p. 20*)
    - \* **timeline** (*Section 10, p. 21*)
    - \* **utils** (*Section 11, p. 34*)

## 1.2 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

## 2 Package `pygce.analysis`

### 2.1 Modules

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

### 2.2 Variables

Name	Description
<code>--package--</code>	<b>Value:</b> 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)***

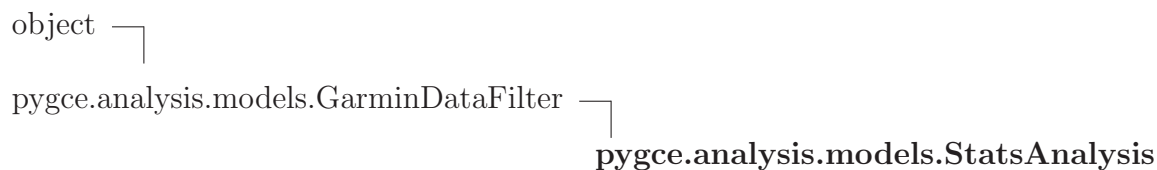
:param headers: [] of str  
 Column names of data  
:param headers\_to\_fix: [] of str  
 Column names of data to fix the float format  
:param data: [] of []  
 Raw data  
:return: [] of []  
 Input data but with fixed floats in columns

***Inherited from object***

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

**4.1.2 Properties**

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

**4.2 Class StatsAnalysis**

Computes correlation of data

**4.2.1 Methods**

<b><code>__init__(self, dataset_file)</code></b> <hr/> <p><code>:param dataset_file: str</code>          Path to folder with data to analyse</p> <p>Overrides: <code>object.__init__</code></p>
<b><code>show_correlation_matrix(self, title_image, headers_to_analyze)</code></b> <hr/> <p><code>:param title_image: str</code>          Title of output image</p> <p><code>:param headers_to_analyze: [] of str</code>          Compute correlation matrix of only these headers</p> <p><code>:return: void</code>          Shows correlation matrix of data of files in folder</p>

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

`convert_time_columns()`, `fix_floats()`, `parse_csv()`

### *Inherited from object*

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

### 4.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 4.3 Class *MLAnalysis*

object

`pygce.analysis.models.GarminDataFilter`

`pygce.analysis.models.MLAnalysis`

Carries out popular machine-learning tasks on Garmin data

### 4.3.1 Methods

<code>__init__(self, dataset_file)</code>
<pre>:param dataset_file: str     Path to folder with data to analyse Overrides: object.__init__</pre>

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

`convert_time_columns()`, `fix_floats()`, `parse_csv()`

### *Inherited from object*

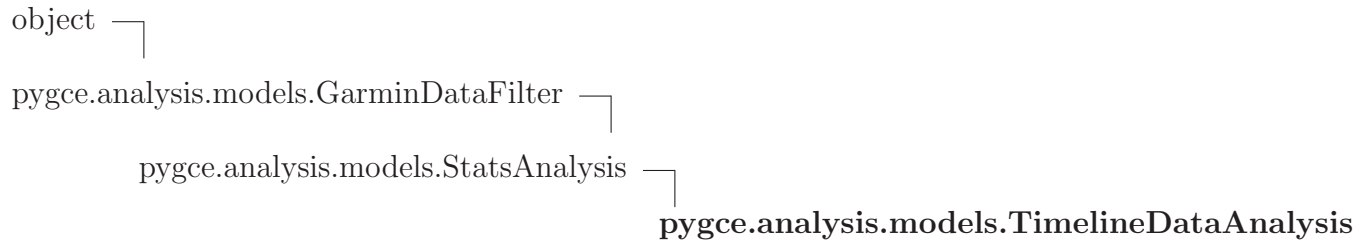
`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,  
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

### 4.3.2 Properties



Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

#### 4.4 Class *TimelineDataAnalysis*



Analyzes and provides insights into Garmin timeline 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`

**`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: void
    Predicts feature with linear regression
```

```
cluster_analyze(self, n_clusters=6)
```

```
:param n_clusters: int
    Number of clusters
:return: void
    Computes cluster analysis: see days based on differences.
    Each day is different from one another, there are days where you trained more, c
    The goal is to divide your days into categories (e.g highly-active, active ...)
    This way, the input matrix consists of multiple vectors with each one consisting
```

```
select_k_best(self, feature, k=5)
```

```
:param feature: str
    Name of feature (column name) to predict
:param k: int
    Number of features to select
:return: void
    Selects the best features to predict feature
```

**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__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

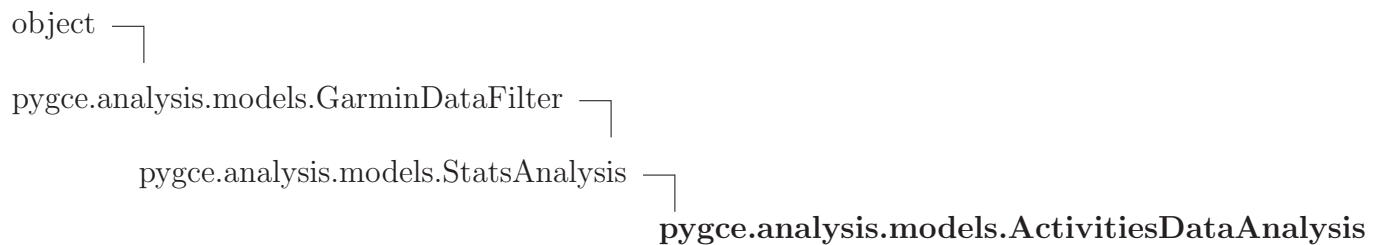
#### 4.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

#### 4.4.3 Class Variables

Name	Description
HEADERS_TO_ANALYZE	<b>Value:</b> ["SUMMARY:kcal_count", "STEPS:distance", "SLEEP:light_sle..."]
TIME_HEADERS_TO_CONVERT	<b>Value:</b> ["SLEEP:nap_time", "SLEEP:light_sleep_time", "SLEEP:awake..."]

### 4.5 Class ActivitiesDataAnalysis



Analyzes and provides insights into Garmin activities data

#### 4.5.1 Methods

<b><code>__init__(self, dataset_file)</code></b> <hr/> :param dataset_file: str Path to folder with data to analyse Overrides: <code>object.__init__</code>
<b><code>parse_csv(self)</code></b> <hr/> :return: tuple [], [] of [] Headers of csv file and data Overrides: <code>pygce.analysis.models.GarminDataFilter.parse_csv</code>

```
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__()
```

#### 4.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

#### 4.5.3 Class Variables

Name	Description
HEADERS_TO_ANALYZE	<b>Value:</b> ["Distance", "Time", "Avg Speed(Avg Pace)", "Max Speed(Be...
TIME_HEADERS_TO_CONVERT	<b>Value:</b> ["Time", "Avg Speed(Avg Pace)", "Max Speed(Best Pace)"]
HEADERS_WITH_MALFORMED_FLOATS	<b>Value:</b> ["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	<b>Value:</b> ["json", "csv"]

## 6 Package `pygce.models`

### 6.1 Modules

- **bot** (*Section 7, p. 16*)
- **garmin** (*Section 8, p. 19*)
  - **activities** (*Section 9, p. 20*)
  - **timeline** (*Section 10, p. 21*)
  - **utils** (*Section 11, p. 34*)

### 6.2 Variables

Name	Description
<code>--package--</code>	<b>Value:</b> 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: GCDayTimeline  
    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 GCDayTimeline  
    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__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

```

#### 7.1.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

#### 7.1.3 Class Variables

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

## 8 Package `pygce.models.garmin`

### 8.1 Modules

- **activities** (*Section 9, p. 20*)
- **timeline** (*Section 10, p. 21*)
- **utils** (*Section 11, p. 34*)

### 8.2 Variables

Name	Description
<code>--package--</code>	<b>Value:</b> None

## 9 Module `pygce.models.garmin.activities`

### 9.1 Variables

Name	Description
<code>--package--</code>	<b>Value:</b> None

## 10 Module `pygce.models.garmin.timeline`

### 10.1 Variables

Name	Description
<code>__package__</code>	Value: <code>'pygce.models.garmin'</code>

### 10.2 Class `GCDaySection`

object └─ `pygce.models.garmin.timeline.GCDaySection`

**Known Subclasses:** `pygce.models.garmin.timeline.GCDayActivities`, `pygce.models.garmin.timeline.GCDaySleep`, `pygce.models.garmin.timeline.GCDaySteps`, `pygce.models.garmin.timeline.GCDaySection`

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__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

#### 10.2.2 Properties

Name	Description
<i>Inherited from object</i>	
__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\_\_(), \_\_getattr\_\_(), \_\_hash\_\_(), \_\_new\_\_(), \_\_reduce\_\_(), \_\_reduce\_ex\_\_(),  
\_\_repr\_\_(), \_\_setattr\_\_(), \_\_sizeof\_\_(), \_\_str\_\_(), \_\_subclasshook\_\_()

### 10.3.2 Properties

Name	Description
<i>Inherited from object</i>	
__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

<b>__init__</b> ( <i>self</i> , <i>raw_html</i> )
: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()

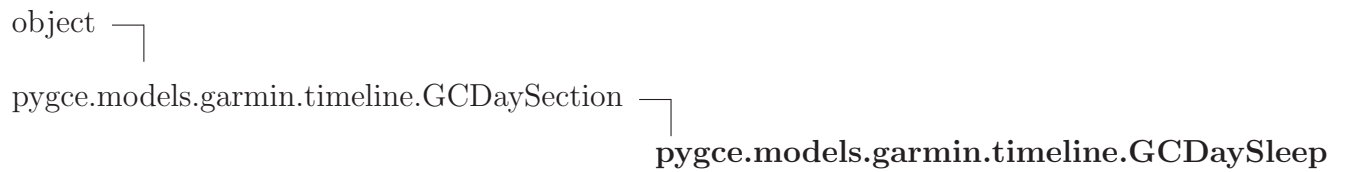
***Inherited from object***

\_\_delattr\_\_(), \_\_format\_\_(), \_\_getattr\_\_(), \_\_hash\_\_(), \_\_new\_\_(), \_\_reduce\_\_(), \_\_reduce\_ex\_\_(),  
\_\_repr\_\_(), \_\_setattr\_\_(), \_\_sizeof\_\_(), \_\_str\_\_(), \_\_subclasshook\_\_()

#### 10.4.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

## 10.5 Class 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__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

### 10.5.2 Properties

Name	Description
<i>Inherited from object</i>	
__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__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,  
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

### 10.6.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 10.7 Class *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__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
```

`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

### 10.7.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 10.8 Class `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
```

---



<b>to_json(<i>self</i>)</b>
<b>:return:</b> json object A json representation of this object

***Inherited from object***

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

**10.8.2 Properties**

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 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	<b>Value:</b> <code>'https://connect.garmin.com'</code>

*continued on next page*

Name	Description
GARMIN_CONNECT_ACTIVITIES_URL	<b>Value:</b> 'https://connect.garmin.com/modern/activities'
__package__	<b>Value:</b> 'pygce.models.garmin'

## Index

- pygce (*package*), 3
  - pygce.analysis (*package*), 4
    - pygce.analysis.cli (*module*), 5
    - pygce.analysis.models (*module*), 6–12
  - pygce.cli (*module*), 13–14
    - pygce.cli.check\_args (*function*), 13
    - pygce.cli.create\_args (*function*), 13
    - pygce.cli.main (*function*), 14
    - pygce.cli.parse\_args (*function*), 13
    - pygce.cli.parse\_yyyy\_mm\_dd (*function*), 13
  - pygce.models (*package*), 15
    - pygce.models.bot (*module*), 16–18
    - pygce.models.garmin (*package*), 19