feupy

Release 0.5.19

Jul 19, 2021

Contents:

CHAPTER 1

Student

class feupy.**Student**(*username: int*, *use_cache: bool = True*, *base_url: str = 'https://sigarra.up.pt/feup/en/'*)

This class represents a FEUP student as seen from their sigarra webpage.

Parameters

- **username** (*int*) – The username of the student, e.g. 201806185
- **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra
- **base_url** (str, optional) – The url of the faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

name

The name of the student

Type str

links

Urls from the student page (including *Student.personal_webpage*, if present)

Type tuple(str)

personal_webpage

Url of the student's personal page, if present. Otherwise it is set to None

Type str

username

The student's "pv_num_unico"

Type int

url

Url of the student's sigarra page

Type str

base_url
The url of the student's faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

Type str

courses
The courses this student is enrolled in

Type tuple(dict)

Each dictionary from the courses tuple has 3 keys

- "course" (a *Course* object or a string (if a link to a course wasn't available))
- "first academic year" (int): if your first year is 2019/2020, then "first academic year" will be 2019
- "institution" (string)

Example:

```
from feupy import Student

daniel = Student(201806185)

print(daniel.name)
# Daniel Filipe Amaro Monteiro

print(daniel.username)
# 201806185

print(daniel.courses)
# ({'course': Course(742, 2019), 'institution': 'Faculty of Engineering', 'first␣
↪academic year': 2018},)
```

classmethod from_a_tag(*bs4_tag: bs4.element.Tag*, *use_cache: bool = True*, *base_url: str = 'https://sigarra.up.pt/feup/en/'*)
Scrapes the student webpage from the given `bs4.tag` object and returns a *Student* object.

Parameters

- **bs4_tag** (`bs4.tag`) –
- **use_cache** (`bool`, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra
- **base_url** (`str`, optional) – The url of the faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

Returns A *Student* object

classmethod from_url(*url: str*, *use_cache: bool = True*, *base_url: str = 'https://sigarra.up.pt/feup/en/'*)
Scrapes the student webpage from the given url and returns a *Student* object.

Parameters

- **url** (*str*) – The url of the student's sigarra page
- **use_cache** (`bool`, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra
- **base_url** (`str`, optional) – The url of the faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

Returns A *Student* object

Example:

```
from feupy import Student

url = "https://sigarra.up.pt/feup/pt/fest_geral.cursos_list?pv_num_
↪unico=201806185"
daniel = Student.from_url(url)

print(daniel.name)
# Daniel Filipe Amaro Monteiro
```

full_info(*credentials: feupy._Credentials.Credentials*) → dict

Returns a dictionary with the information that one can get when it is logged in.

The dictionary has 7 keys:

- "courses" (list(dict))
- "email" (str)
- "links" (tuple(str))
- "name" (str)
- "personal_webpage" (str)
- "url" (str)
- "username" (int)

Parameters **credentials** (*Credentials*) – A *Credentials* object

Returns A dictionary

Example:

```
from feupy import Student, Credentials
from pprint import pprint

daniel = Student(201806185)

creds = Credentials()

pprint(daniel.full_info(creds))

# You will get something like this:
{'courses': [ # A list of dictionaries representing the courses' information
              {'course': Course(742, 2019), # A Course object. If a link to an␣
↪object isn't available, it's just a string
               'current year': 2,   # Could be None if a number isn't present␣
↪(or couldn't be parsed)
               'first academic year': 2018,
               'institution': 'Faculty of Engineering', # Best faculty
               'status': 'A Frequentar'}],
'email': 'up201806185@fe.up.pt',
'links': (),                        # personal_webpage is included in links
'name': 'Daniel Filipe Amaro Monteiro', # people tend to have a name
'personal_webpage': None,
'url': 'https://sigarra.up.pt/feup/en/fest_geral.cursos_list?pv_num_
↪unico=201806185',
'username': 201806185}
```

CHAPTER 2

Teacher

class feupy.**Teacher**(*p_codigo: int*, *use_cache: bool = True*, *base_url: str = 'https://sigarra.up.pt/feup/en/'*)

This class represents a FEUP teacher as seen from their sigarra webpage.

Note: Some attributes may be set to None, depending on whether or not that attribute was able to be parsed. For example: personal_webpage, email, and presentation are not always available in teacher pages.

Parameters

- **p_codigo** (*int*) – The id of the teacher
- **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra
- **base_url** (str, optional) – The url of the faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

p_codigo

The id of the teacher

Type int

name

The name of the teacher

Type str

acronym

The acronym of the teacher

Type str

status

The status of the teacher

Type str

links
Urls from the teacher page (including *Teacher.personal_webpage*, if present)

Type tuple(str)

personal_webpage
Url of the teacher's personal page

Type str

url
Url of the teacher's sigarra page

Type str

voip

Type int

email

Type str

rooms

Type str

category

Type str

career

Type str

profession

Type str

department

Type str

presentation
The presentation of this teacher

Type str

base_url
The url of the teacher's faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

Type str

Example:

```
from feupy import Teacher

jlopes = Teacher(230756)

print(jlopes.name)
# João António Correia Lopes

print(jlopes.acronym)
# JCL

print(jlopes.personal_webpage)
# http://www.fe.up.pt/~jlopes/
```

classmethod from_a_tag(*bs4_tag: bs4.element.Tag*, *use_cache: bool = True*, *base_url: str = 'https://sigarra.up.pt/feup/en/'*)

Scrapes the teacher webpage from the given `bs4.tag` object and returns a *Teacher* object.

Parameters

- **bs4_tag** (`bs4.tag`) –
- **use_cache** (`bool`, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra
- **base_url** (`str`, optional) – The url of the faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

Returns A *Teacher* object

classmethod from_url(*url: str*, *use_cache: bool = True*, *base_url: str = 'https://sigarra.up.pt/feup/en/'*)

Scrapes the teacher webpage from the given url and returns a *Teacher* object.

Parameters

- **url** (*str*) – The url of the teacher's sigarra page
- **use_cache** (`bool`, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra
- **base_url** (`str`, optional) – The url of the faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

Returns A *Teacher* object

Example:

```
from feupy import Teacher

url = "https://sigarra.up.pt/feup/en/func_geral.formview?p_codigo=230756"
jlopes = Teacher.from_url(url)

print(jlopes.name)
# João António Correia Lopes
```

picture() → PIL.Image.Image

Returns a picture of the teacher as a `PIL.Image.Image` object.

Returns A `PIL.Image.Image` object

CHAPTER 3

CurricularUnit

class feupy.**CurricularUnit**(*pv_ocorrencia_id: int*, *use_cache: bool = True*, *base_url: str = 'https://sigarra.up.pt/feup/en/'*, *try_recovery: bool = True*)

This class represents a FEUP curricular unit.

Parameters

- **pv_ocorrencia_id** (*int*) – The id of the curricular unit
- **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra
- **base_url** (str, optional) – The url of the faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

pv_ocorrencia_id

The id of the curricular unit

Type int

url

The url of the curricular unit page

Type str

name

The name of the curricular unit

Type str

code

Type str

acronym

The acronym of the curricular unit

Type str

academic_year

The academic year in which this curricular unit was taught

Type int

semester
The semester in which this curricular unit was taught (either '1', '2', or 'A')

Type str

has_moodle
Whether or not this curricular unit has a moodle page

Type bool

is_active
Whether or not this curricular unit is active

Type bool

webpage_url
The webpage of this curricular unit, if it exists. Otherwise it's set to None

Type str or None

number_of_students

Type int

curricular_years
usually 1-5

Type tuple(int)

ECTS_credits

Type float

regents

Type tuple(*Teacher*)

teachers

Type tuple(*Teacher*)

text
Basically a text dump starting from "Teaching language"

Type string

base_url
The url of the curricular unit's faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

Type str

Example:

```
from feupy import CurricularUnit
from pprint import pprint

mpcp = CurricularUnit(419989)

print(mpcp.name)
# Microprocessors and Personal Computers

print(mpcp.acronym)
# MPCP
```

(continues on next page)

(continued from previous page)

```
print(mpcp.semester)
# 2

for teacher in mpcp.teachers:
    print(teacher.name)
# João Paulo de Castro Canas Ferreira
# Bruno Miguel Carvalhido Lima
# António José Duarte Araújo
# João Paulo de Castro Canas Ferreira
# João Paulo Filipe de Sousa

pprint(vars(mpcp))
# You'll get something like this:
{'ECTS_credits': 6.0,
'academic_year': 2018,
'acronym': 'MPCP',
'code': 'EIC0016',
'curricular_years': (1,),
'has_moodle': True,
'is_active': True,
'name': 'Microprocessors and Personal Computers',
'number_of_students': 220,
'pv_ocorrencia_id': 419989,
'regents': (Teacher(210963),),
'semester': '2',
'teachers': (Teacher(210963),
             Teacher(547486),
             Teacher(211636),
             Teacher(210963),
             Teacher(210660)),
'text': 'Teaching language\n'
        'Portuguese\n'
        'Objectives\n'
        'BACKGROUND\n'
        'The PC-compatible desktop and mobile platforms are an everyday tool '
        'in modern societies. Their architecture reflects the current '
        'technological development, but also defines the limits of the '
        "computer's capabilities and performance. Variants of the ARM "
        'instruction set are used today n most mobile platforms (tablets, '
        'mobile phones)in use today. Both system architecture and ISA have a '
        'deep impact on the day-to-day practice of informatics engineers.\n'
        'SPECIFIC AIMS\n'
        'The ... etc',
'url': 'https://sigarra.up.pt/feup/en/ucurr_geral.ficha_uc_view?pv_ocorrencia_
↪id=419989',
'webpage_url': None}
```

all_timetables(*credentials: feupy._Credentials.Credentials*, *ignore_coherence: bool = False*) → dict

Parses all the timetables related to this curricular unit (see *timetable.parse_timetables* for further info).

Returns A dictionary which maps a tuple with two `datetime.date` objects, start and finish (the time span in which this timetable is valid), to a list of dictionaries (see *timetable.parse_timetable* for an example of such a list).

classes(*credentials: feupy._Credentials.Credentials*, *full_info=False*, *use_cache: bool = True*) → dict

Returns a dictionary which maps a class name to a list of students (the students of that class).

Parameters

- **credentials** (*Credentials*) – A *Credentials* object
- **full_info** (bool, optional) – Returns a list of tuples. Each tuple contains the student, the allocation date, whether or not it was allocated by the administration, and whether or not the student is enrolled
- **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra

Returns A dict

Example:

```
from feupy import CurricularUnit, Credentials
from pprint import pprint

mpcp = CurricularUnit(419989)

creds = Credentials()

pprint(mpcp.classes(creds))
# You'll get something like this:
{
    '1MIEIC01': [Student(201800000),
                 Student(201800001),
                 Student(201800002),
                 Student(201800003)],
    '1MIEIC02': [Student(201800004),
                 Student(201800005),
                 Student(201800006),
                 Student(201800007),
                 Student(201800008),
                 Student(201800009),
                 Student(201800010),
                 Student(201800011),
                 Student(201800012)],
    '...'     : [...]
}
```

contents(*credentials: feupy._Credentials.Credentials*) → dict

Returns a nested dictionary structure, where each dictionary represents a folder. Every dictionary maps a string (the folder or file name) to either a dictionary (a nested folder) or a tuple (a file or a link).

A tuple representing a file is made of 4 attributes:

0. file type, either "file" or "link" (str)
1. url (Note: you can download files by passing this url to *Credentials.download()*) (str)
2. info (str or None)
3. upload date (Note: links will have this attribute always set to today) (Datetime.Date)

Parameters **credentials** (*Credentials*) – A *Credentials* object

Returns A nested dictionary structure

Example:

```
from feupy import CurricularUnit, Credentials
from pprint import pprint

mpcp = CurricularUnit(419989)

creds = Credentials()

pprint(mpcp.contents(creds))
# You'll get something like this:
{'A folder': {'folderception': {'A file': ('file',
                                           'https://sigarra.up.pt/feup/pt/
↪conteudos_service.conteudos_cont?pct_id=012345&pv_cod=06ahastCa',
                                           None,
                                           datetime.date(2018, 10, 2)),
                               'Python': ('file',
                                          'https://sigarra.up.pt/feup/pt/
↪conteudos_service.conteudos_cont?pct_id=012346&pv_cod=06TtaSaPH7',
                                          'aaaaaa',
                                          datetime.date(2018, 11, 13)),
                               'Simulator': ('link',
                                             'https://github.com/hneemann/
↪Digital',
                                             'It\'s a simulator.',
                                             datetime.date(2019, 5, 25))},
              'jhbh': {'aa': ('link',
                              'https://salmanarif.bitbucket.io/visual/index.html
↪',
                              'aaaaaa',
                              datetime.date(2019, 8, 2)),
                       'bb': ('file',
                              'https://sigarra.up.pt/feup/pt/conteudos_service.
↪conteudos_cont?pct_id=012347&pv_cod=06WjyvGato',
                              'banana',
                              datetime.date(2018, 10, 12))}}}
```

exams (*use_cache: bool = True*)

Returns a list of the exams of this curricular unit.

Parameters **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra

Returns A list of dictionaries (see *exams.exams()* for more information about the dictionaries)

classmethod from_a_tag (*bs4_tag: bs4.element.Tag*, *use_cache: bool = True*, *base_url: str = 'https://sigarra.up.pt/feup/en/'*)

Scrapes the curricular unit webpage from the given bs4.tag object and returns a *CurricularUnit* object.

Parameters

- **bs4_tag** (bs4.tag) –
- **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra
- **base_url** (str, optional) – The url of the faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

Returns A *CurricularUnit* object

classmethod from_url(*url: str, use_cache: bool = True, base_url: str = 'https://sigarra.up.pt/feup/en/'*)

Scrapes the curricular webpage from the given url and returns a *CurricularUnit* object.

Parameters

- **url** (*str*) – The url of the curricular unit's sigarra page
- **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra
- **base_url** (str, optional) – The url of the faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

Returns A *CurricularUnit* object

Example:

```
from feupy import CurricularUnit

url = "https://sigarra.up.pt/feup/pt/ucurr_geral.ficha_uc_view?pv_ocorrencia_
↪id=419989"
mpcp = CurricularUnit.from_url(url)

print(mpcp.name)
# Microprocessors and Personal Computers
```

grades_distribution(*credentials: feupy._Credentials.Credentials*) → dict

Returns the distribution of the grades as a dict.

Parameters **credentials** (*Credentials*) – A *Credentials* object

Returns A dictionary that maps the grade to the number of students that got that grade.

Example:

```
from feupy import CurricularUnit, Credentials
from pprint import pprint

mpcp = CurricularUnit(419989)

creds = Credentials()

pprint(mpcp.grades_distribution(creds))
# You'll get something like this:
{'RFC': 9,
'RFE': 17,
'RFF': 18,
    5: 4,
    6: 5,
    7: 7,
    8: 6,
    9: 4,
    10: 12,
    11: 11,
    12: 13,
    13: 21,
    14: 4,
    15: 12,
    16: 7,
    17: 13,
```

(continues on next page)

(continued from previous page)

```
    18: 8,
    19: 7,
    20: 1}
```

other_occurrences(*use_cache: bool = True*) → tuple

Returns the occurrences of this curricular unit from other years as a tuple of *CurricularUnit* objects.

Parameters **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra

Returns A tuple of *CurricularUnit* objects

Example:

```
from feupy import CurricularUnit
from pprint import pprint

mpcp = CurricularUnit(419989)

pprint(mpcp.other_occurrences())
# You'll get something like this:
(CurricularUnit(436431),
CurricularUnit(419989),
CurricularUnit(399884),
CurricularUnit(384929),
CurricularUnit(368695),
CurricularUnit(350451),
CurricularUnit(333111),
CurricularUnit(272647),
CurricularUnit(272646),
CurricularUnit(272645),
CurricularUnit(272644),
CurricularUnit(272642),
CurricularUnit(272641),
CurricularUnit(272640),
CurricularUnit(272639))
```

results(*credentials: feupy._Credentials.Credentials*, *use_cache: bool = True*) → dict

Returns a dictionary which maps a string representing the exams season ('Época Normal (2ºS)'/'Época Recurso (2ºS)') to a list of tuples.

Each tuple in the list has two values:

1. student (*Student*)
2. grade (str or int)

Parameters

- **credentials** (*Credentials*) – A *Credentials* object
- **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra

Returns A dict

Example:

```
from feupy import CurricularUnit, Credentials
from pprint import pprint

mpcp = CurricularUnit(419989)

creds = Credentials()

pprint(mpcp.results(creds))
# You'll get something like this:
{'Época Normal (2ºS)': [(Student(201800001), 10),
                        (Student(201800002), 13),
                        (Student(201800003), 10),
                        (Student(201800004), 'RFE'),
                        (Student(201800005), 'RFF'),
                        (Student(201800006), 'RFF'),
                        ...],
'Época Recurso (2ºS)': [(Student(201800008), 11),
                        (Student(201800009), 7),
                        (Student(201800010), 8),
                        (Student(201800011), 8),
                        (Student(201800012), 'RFE'),
                        (Student(201800013), 13),
                        (Student(201800014), 5),
                        (Student(201800019), 'RFC'),
                        ...]}
```

statistics_history(*credentials: feupy._Credentials.Credentials*) → list

Returns a list of tuples each representing an academic year.

Contents of a tuple:

0. academic year (int)
1. registered students (int)
2. evaluated students (int)
3. approved students (int)
4. evaluated students average grade (float)
5. evaluated students standard deviation (float)
6. approved students average grade (float)
7. approved students standard deviation (float)

Parameters **credentials** (*Credentials*) – A *Credentials* object

Returns A list of tuples

Example:

```
from feupy import CurricularUnit, Credentials
from pprint import pprint

mpcp = CurricularUnit(419989)

creds = Credentials()
```

(continues on next page)

(continued from previous page)

```
pprint(mpcp.statistics_history(creds))

# You will get something like this:
[(2016, 208, 152, 120, 12.43, 3.75, 13.4,  2.5),
 (2017, 203, 149, 113, 12.08, 3.08, 12.27, 3.0),
 (2018, 203, 170, 133, 12.51, 3.76, 13.53, 2.76),
 ... ]
```

stats (*credentials: feupy._Credentials.Credentials*) → tuple

Returns a tuple with 3 ints:

0. number of registered students
1. number of evaluated students
2. number of approved students

Parameters **credentials** (*Credentials*) – A *Credentials* object

Returns A tuple with 3 ints

Example:

```
from feupy import CurricularUnit, Credentials

mpcp = CurricularUnit(419989)

creds = Credentials()

print(mpcp.stats(creds))
# You'll get something like this:
(220, 185, 162)
```

students (*credentials: feupy._Credentials.Credentials*, *use_cache: bool = True*) → list

Returns the students of this curricular unit as a list of tuples.

Each tuple has 4 elements:

0. student (*Student*)
1. status (str)
2. number of registrations (int)
3. type of student (str)

Parameters

- **credentials** (*Credentials*) – A *Credentials* object
- **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra

Returns A list of tuples

Example:

```
from feupy import CurricularUnit, Credentials
from pprint import pprint

mpcp = CurricularUnit(419989)

creds = Credentials()

pprint(mpcp.students(creds))
# You'll get something like this:
[
    ...,
    (Student(201812345), 'Ordinário', 1, 'Normal'),
    ...
]
```

timetable(*credentials: feupy._Credentials.Credentials*, *ignore_coherence: bool = False*) → list

Returns the curricular unit's current timetable as a list of dictionaries if possible, otherwise returns None. (see *timetable.parse_current_timetable()* for more info)

Returns A list of dicts

CHAPTER 4

Course

class feupy.**Course**(*pv_curso_id: int*, *pv_ano_lectivo: int = None*, *use_cache: bool = True*, *base_url: str = 'https://sigarra.up.pt/feup/en/'*, *try_recovery: bool = True*)

This class represents a course as seen from its sigarra webpage.

Parameters

- **pv_curso_id** (*int*) – The id of the course, for example MIEIC's id is 742
- **pv_ano_lectivo** (int, optional) – The year of this course. It defaults to the current year (i.e. 2019, at the time of writing)
- **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra
- **base_url** (str, optional) – The url of the faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

pv_curso_id
The id of the course

Type int

pv_ano_lectivo
The year of this course's page

Type int

url
Url of the course's sigarra page

Type str

name
The name the course

Type str

official_code
The official code

Type str

directors
The directors of this course

Type tuple(*Teacher*)

acronym
The acronym of this course

Type str

text
The text that can be found in the course's page

Type str or None

base_url
The url of the course's faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

Type str

involved_organic_units
All the faculties involved with this course

Type tuple(str)

Example:

```
from feupy import Course

mieic = Course(742)

print(mieic.name)
# Master in Informatics and Computing Engineering

print(mieic.acronym)
# MIEIC

for director in mieic.directors:
    print(director.name)
# João Carlos Pascoal Faria
# Maria Cristina de Carvalho Alves Ribeiro
```

classes(*credentials: feupy._Credentials.Credentials*) → dict
Returns a dictionary which maps the course's classes to their corresponding timetable links.

Parameters **credentials** (*Credentials*) – A *Credentials* object

Returns A dictionary

Example:

```
from feupy import Course, Credentials
from pprint import pprint

mieic = Course(742)

creds = Credentials()

pprint(mieic.classes(creds))
```

(continues on next page)

(continued from previous page)

```
# You will get something like this:
{'1MIEIC01':'https://sigarra.up.pt/feup/pt/hor_geral.turmas_view?pv_turma_
↪id=000001&pv_periodos=1&pv_ano_lectivo=2018',
'1MIEIC02': 'https://sigarra.up.pt/feup/pt/hor_geral.turmas_view?pv_turma_
↪id=000002&pv_periodos=1&pv_ano_lectivo=2018',
'1MIEIC03': 'https://sigarra.up.pt/feup/pt/hor_geral.turmas_view?pv_turma_
↪id=000003&pv_periodos=1&pv_ano_lectivo=2018',
'1MIEIC04': 'https://sigarra.up.pt/feup/pt/hor_geral.turmas_view?pv_turma_
↪id=000004&pv_periodos=1&pv_ano_lectivo=2018',
'1MIEIC05': 'https://sigarra.up.pt/feup/pt/hor_geral.turmas_view?pv_turma_
↪id=000005&pv_periodos=1&pv_ano_lectivo=2018',
'1MIEIC06': 'https://sigarra.up.pt/feup/pt/hor_geral.turmas_view?pv_turma_
↪id=000006&pv_periodos=1&pv_ano_lectivo=2018',
...
↪                                                    }
```

You can use the functions provided by *timetable* to get the timetables from the urls.

curricular_units (*use_cache: bool = True*) → list

Returns a list of *CurricularUnit* objects representing the curricular units of the course. Curricular units without a link are not included.

Parameters **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra

Returns A list of *CurricularUnit* objects

Example:

```
from feupy import Course
from pprint import pprint

mieic = Course(742)

pprint(mieic.curricular_units())

# You will get something like this:
[CurricularUnit(419981),
 CurricularUnit(419982),
 CurricularUnit(419983),
 CurricularUnit(419984),
 CurricularUnit(419985),
 CurricularUnit(419986),
 CurricularUnit(419987),
 CurricularUnit(419988),
 CurricularUnit(419989),
 CurricularUnit(419990),
 CurricularUnit(419991),
 CurricularUnit(419992),
 CurricularUnit(419993),
 CurricularUnit(419994),
 CurricularUnit(419995),
 CurricularUnit(419996),
 ...                   ]
```

exams (*use_cache: bool = True*) → list

Returns a list of the exams of this course.

Parameters **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra

Returns A list of dictionaries (see *exams.exams()* for more information about the dictionaries)

classmethod from_a_tag(*bs4_tag: bs4.element.Tag*, *use_cache: bool = True*, *base_url: str = 'https://sigarra.up.pt/feup/en/'*)

Scrapes the course webpage from the given bs4.tag object and returns a *Course* object.

Parameters

- **bs4_tag** (bs4.tag) –
- **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra
- **base_url** (str, optional) – The url of the faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

Returns A *Course* object

classmethod from_url(*url: str*, *use_cache: bool = True*, *base_url: str = 'https://sigarra.up.pt/feup/en/'*)

Scrapes the course webpage from the given url and returns a *Course* object.

Parameters

- **url** (*str*) – The url of the course's sigarra page
- **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra
- **base_url** (str, optional) – The url of the faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

Returns A *Course* object

Example:

```
from feupy import Course

url = "https://sigarra.up.pt/feup/pt/cur_geral.cur_view?pv_curso_id=742&pv_
↪ano_lectivo=2018"
mieic = Course.from_url(url)

print(mieic.name)
# Master in Informatics and Computing Engineering
```

syllabus(*use_cache: bool = True*) → list

Returns a list of tuples containing more information about the curricular units of the course.

Each tuple has 4 elements:

0. keyword (str)
1. course branch (either str or None, depending on whether or not the curricular unit belongs to a specific branch)
2. curricular unit (*CurricularUnit*)
3. is mandatory (bool)

Parameters **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra

Returns A list of tuples

Example:

```
from feupy import Course
from pprint import pprint

miem = Course(743)

pprint(miem.syllabus())

# You will get something like this:
[
    ('Automation', 'Automation', CurricularUnit(418771), False),
    ('Automation', 'Automation', CurricularUnit(418770), False),
    ('Automation', 'Automation', CurricularUnit(418737), True),
    ('Automation', 'Automation', CurricularUnit(418746), True),
    ('Automation', 'Automation', CurricularUnit(418747), False),
    ('Automation', 'Automation', CurricularUnit(418736), True),
    ('Automation', 'Thermal Energy', CurricularUnit(418773), False),
    ('Automation', 'Thermal Energy', CurricularUnit(418736), False),
    ('Automation', 'Production Management', CurricularUnit(418773), False),
    ('Automation', 'Production Management', CurricularUnit(418736), False),
    ('Automation', 'Production, Conception and Manufacturing',␣
↪CurricularUnit(418773), False),
    ('Automation', 'Structural Engineering and Machine Design',␣
↪CurricularUnit(418736), False),
    ('Automation', None, CurricularUnit(418722), True),
    ('Personal and Interpersonal Skills', 'Automation',␣
↪CurricularUnit(418787), False),
    ...
]
```

CHAPTER 5

Credentials

class feupy.**Credentials**(*username=None*, *password: str = None*, *base_url: str = 'https://sigarra.up.pt/feup/en/'*)

This class represents your login credentials. It also features a non-persistent cache that stores htmls of urls previously accessed.

Parameters

- **username** (int or str, optional) – Your username. You will be prompted for your username if you don't pass a username as an argument
- **password** (str, optional) – Your password. You will be prompted for your password if you don't pass a password as an argument
- **base_url** (str, optional) – The url of the faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

username
Your username (e.g. 201806185)

Type int or str

session
A requests.Session object that holds the login cookies

Type requests.Session

cache
A dictionary that maps a url string to an html string

Type dict

base_url
The url of your faculty (in english) (defaults to "https://sigarra.up.pt/feup/en/")

Type str

Example:

```
from feupy import Credentials

username = 201806185
password = "https://www.youtube.com/watch?v=dQw4w9WgXcQ"

creds = Credentials(username, password)

# You may instead enter your username and password at runtime
creds = Credentials()
# Username?
# :> up201806185
# Password for 201806185?
# :>
```

download(*url: str*, *folder_path: str*) → str

Downloads the file from the given url and saves it to the given folder path. If the path doesn't exist, it will be created automatically.

Parameters

- **url** (*str*) – The url of the file to be downloaded
- **folder_path** (*str*) – The path of the folder you want to download the file to

Returns The path of the file as an str

get_html(*url: str*, *params: dict = {}*) → str

Functionally equivalent to `requests.get(url, params).text`, with scripts and styles removed. If the result is already in cache, the method will just return the value from the cache instead of making a web request.

Parameters

- **url** (*str*) – The url of the html to be fetched
- **params** (dict, optional) – the query portion of the url, should you want to include a query

Returns A string which is the html from the requested page url

get_html_async(*urls*, *n_workers: int = 10*)

Credentials.get_html(), but async, give or take.

Takes a list (or any iterable) of urls and returns a corresponding generator of htmls. The htmls have their scripts and styles removed and are stored in cache.

Parameters

- **urls** (*iterable(str)*) – The urls to be accessed
- **n_workers** (int, optional) – The number of workers.

Returns An str generator

CHAPTER 6

User

class feupy.**User**(*pv_fest_id: int*, *credentials: feupy._Credentials.Credentials*)

This class represents the information that can be extracted from your personal webpage.

Parameters

- **pv_fest_id** (*int*) – Your pv_fest_id. See *User.get_pv_fest_ids()* for a way to get this value. If you are enrolled in only one course, then *User.from_credentials()* may be more convenient for you
- **credentials** (*Credentials*) – A *Credentials* object

pv_fest_id

A sort of course specific student identification number

Type int

course

The course related to this pv_fest_id

Type *Course*

credentials

A *Credentials* object (This is done to avoid having to pass the same *Credentials* object to every function)

Type *Credentials*

Example:

```
from feupy import Credentials, User

creds = Credentials()

me = User(123456, creds)
```

all_timetables() → dict

Parses all the timetables related to this user (see *timetable.parse_timetables* for further info).

Returns A dictionary which maps a tuple with two `datetime.date` objects, start and finish (the time span in which this timetable is valid), to a list of dictionaries (see *timetable.parse_timetable* for an example of such a list).

classes()
Returns the classes you are in as a list of tuples.

Each tuple has 2 elements:

0. A *CurricularUnit* object
1. The class name as a string.

Returns A list of tuples

courses_units() → list
Returns your grades as a list of tuples.

Each tuple has 2 elements:

0. **A *CurricularUnit* object which represents a curricular unit that** you either have had in a previous year or you are currently enrolled in
1. **Either an int which represents the grade you got at that curricular unit or None** if that grade is not available

Returns A list of tuples

Example:

```
from feupy import Credentials, User
from pprint import pprint


creds = Credentials()
me = User.from_credentials(creds)

pprint(me.courses_units())

# You will get something like this:
[(CurricularUnit(419981), 10),
 (CurricularUnit(419982), 11),
 (CurricularUnit(419983), 12),
 (CurricularUnit(419984), 13),
 (CurricularUnit(419985), 14),
 (CurricularUnit(420521), 15),
 (CurricularUnit(419986), None),
 (CurricularUnit(419987), None),
 (CurricularUnit(419988), None),
 (CurricularUnit(419989), None),
 (CurricularUnit(419990), None)]
```

classmethod from_credentials(*credentials: feupy._Credentials.Credentials*)
Returns a User object made from the first result from get_pv_fest_ids().

Usually, students are enrolled in only one course, which means that get_pv_fest_ids() tends to be a tuple with a single int.

Parameters **credentials** (*Credentials*) – A *Credentials* object

Returns A *User* object

Example:

```
from feupy import Credentials, User

creds = Credentials()
me = User.from_credentials(creds)
```

static get_pv_fest_ids(*credentials: feupy._Credentials.Credentials*) → tuple

Returns a tuple of ints, each representing a pv_fest_id.

Parameters **credentials** (*Credentials*) – A *Credentials* object

Returns A tuple of ints

timetable() → list

Returns the current user timetable as a list of dictionaries if possible, otherwise returns None. (see *timetable.parse_current_timetable()* for more info)

Returns A list of dicts

CHAPTER 7

cache

Caching utilities

feupy.cache.**cache**
A persistent dictionary-like object whose values are structured in the following way:

{
url0 : (timeout0, html0),
url1 : (timeout1, html1),
...
}

In which url is a string, timeout is an int or a float (which represents the "due by date" as seconds since epoch), and html is a string

Type shelve.DbfilenameShelf or None (Initially, see *load_cache()*)

feupy.cache.**load_cache**(*flag='c'*, *path=None*)
Loads the cache from disk and stores it in the variable *cache*. If *cache* is different than None, the function will do nothing.

Parameters

- **flag** (str, optional) – The flag parameter, see https://docs.python.org/3/library/dbm.html#dbm.open
- **path** (str or None, optional) – The path of the directory where the cache is stored. It defaults to this file's folder path

Note: Unless you intend to call *load_cache()* with non-default arguments, you don't have to call this function. The other functions in this module check whether or not the cache has been loaded and will load the cache for you.

Example:

```
from feupy import cache
cache.load_cache()
```

feupy.cache.**get_html**(*url*, *params={}*, *use_cache=True*)

More or less functionally equivalent to `requests.get(url, params).text`, with the added benefit of a persistent cache with customizable html treatment and timeouts, depending on the url. If the result is already in cache and is valid, the function will just return the value from the cache instead of making a web request.

Parameters

- **url** (*str*) – The url of the html to be fetched
- **params** (dict, optional) – the query portion of the url, should you want to include a query
- **use_cache** (bool, optional) – If this value is set to True, the cache will be checked for the url. If the url is not found in the cache keys or has timed out, the function will get the html from the web, remove scripts and styles from the html, store it in cache, and finally return the html. Otherwise, if it's set to False, the cache will not be checked

Returns A string which is the html from the requested page url

---

Note: The curricular units' pages, along with the students' and teachers' htmls, are modified to reduce their memory footprint.

Note: If you know that you are going to make a crapton of requests beforehand, you probably should call *get_html_async()* first to populate the cache.

---

feupy.cache.**reset**()

Eliminates all entries from the cache

feupy.cache.**remove_invalid_entries**(*urls=None*)

Removes all the cache entries in urls that have timed out.

Parameters **urls** (iterable(str) or None, optional) – The urls to be checked. If this argument is left untouched, all urls in the cache will be checked

feupy.cache.**get_html_async**(*urls*, *n_workers=10*, *use_cache=True*)

get_html(), but async, give or take.

Takes a list (or any iterable) of urls and returns a corresponding generator of htmls. The htmls have their scripts and styles removed and are stored in cache.

Parameters

- **urls** (*iterable(str)*) – The urls to be accessed
- **n_workers** (int, optional) – The number of workers.
- **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra

Returns An str generator

CHAPTER 8

exams

feupy.exams.**exams**(*url: str*, *use_cache: bool = True*, *base_url: str = 'https://sigarra.up.pt/feup/en/'*) → list

Returns a list of dictionaries.

Each dictionary represents an exam and has 6 keys:

key	value
"curricular unit"	(*feupy.CurricularUnit*) The subject being evaluated
"finish"	(datetime.datetime) The finish time of the exam
"observations"	(str)
"rooms"	(tuple(str)) The rooms in which the exam will take place
"season"	(str)
"start"	(datetime.datetime) The start time of the exam

Note: If a value is not available, it will be set to None.

Parameters

- **url** (*str*) – The url of the page with the exams to be parsed
- **use_cache** (bool, optional) – Attempts to use the cache if True, otherwise it will fetch from sigarra

Returns A list of dictionaries

Example:

```
from feupy.exams import exams
from pprint import pprint

mieic_exams_url = "https://sigarra.up.pt/feup/pt/exa_geral.mapa_de_exames?p_curso_
↪id=742"
```

(continues on next page)

(continued from previous page)

```
pprint(exams(mieic_exams_url))
# You'll get something like this:
[{'curricular unit': CurricularUnit(420016),
'finish': datetime.datetime(2019, 9, 4, 12, 0),
'observations': 'Exame em comum com Mecanica.Exame em comum com o da Época '
                'de Conclusão de Curso',
'rooms': ('B222',),
'season': 'Exames ao abrigo de estatutos especiais - Port.Est.Especiais 2ºS',
'start': datetime.datetime(2019, 9, 4, 9, 0)},

{'curricular unit': CurricularUnit(419990),
'finish': datetime.datetime(2019, 9, 6, 17, 0),
'observations': 'Sala de exame - Em principio 2 alunos ',
'rooms': ('B222',),
'season': 'Exames ao abrigo de estatutos especiais - Port.Est.Especiais 2ºS',
'start': datetime.datetime(2019, 9, 6, 14, 0)},

{'curricular unit': CurricularUnit(420021),
'finish': datetime.datetime(2019, 9, 18, 17, 30),
'observations': 'Sala de exame - possivelmente 3 alunos ',
'rooms': None,
'season': 'Exames ao abrigo de estatutos especiais - Port.Est.Especiais 2ºS',
'start': datetime.datetime(2019, 9, 18, 14, 30)},

{'curricular unit': CurricularUnit(438941),
'finish': datetime.datetime(2019, 9, 25, 13, 0),
'observations': None,
'rooms': ('B104', 'B208', 'B213'),
'season': 'Exames ao abrigo de estatutos especiais - Mini-testes (1ºS)',
'start': datetime.datetime(2019, 9, 25, 9, 0)},

{'curricular unit': CurricularUnit(438941),
'finish': datetime.datetime(2019, 9, 25, 17, 30),
'observations': None,
'rooms': ('B104', 'B213', 'B208', 'B207'),
'season': 'Exames ao abrigo de estatutos especiais - Mini-testes (1ºS)',
'start': datetime.datetime(2019, 9, 25, 13, 30)},

{'curricular unit': CurricularUnit(438941),
'finish': datetime.datetime(2019, 9, 26, 13, 0),
'observations': None,
'rooms': ('B104', 'B208', 'B213'),
'season': 'Exames ao abrigo de estatutos especiais - Mini-testes (1ºS)',
'start': datetime.datetime(2019, 9, 26, 9, 0)},

{'curricular unit': CurricularUnit(438941),
'finish': datetime.datetime(2019, 9, 26, 17, 30),
'observations': None,
'rooms': ('B104', 'B213', 'B208', 'B207'),
'season': 'Exames ao abrigo de estatutos especiais - Mini-testes (1ºS)',
'start': datetime.datetime(2019, 9, 26, 13, 30)}]
```

CHAPTER 9

timetable

feupy.timetable.**parse_current_timetable**(*credentials: feupy._Credentials.Credentials*, *url: str*, *ignore_coherence: bool = False*)

Attempts to return the related timetable that is valid today as a list of dictionaries. If no timetable is valid today, it returns None.

Parameters

- **credentials** (*feupy.Credentials*) – A *feupy.Credentials* object
- **url** (*str*) – The url of the timetable page
- **ignore_coherence** (bool, optional) – Whether or not this function should raise a CoherenceError exception if the information of a class can not be fetched

Returns A list of dicts (e.g. see *parse_timetable()*) or None

feupy.timetable.**parse_timetables**(*credentials: feupy._Credentials.Credentials*, *url: str*, *ignore_coherence: bool = False*) → dict

Returns the timetables related to this timetable (including) as a dictionary.

Parameters

- **credentials** (*feupy.Credentials*) – A *feupy.Credentials* object
- **url** (*str*) – The url of the timetable page
- **ignore_coherence** (bool, optional) – Whether or not this function should raise a CoherenceError exception if the information of a class can not be fetched

Returns A dictionary which maps a tuple with two datetime.date objects, start and finish (the time span in which this timetable was/is valid), to a list of dictionaries (e.g. see *parse_timetable()*).

Example:

```
from feupy import timetable, Credentials
from pprint import pprint
```

(continues on next page)

(continued from previous page)

```
timetable_url = "https://sigarra.up.pt/feup/pt/hor_geral.turmas_view?pv_turma_
↪id=207783&pv_periodos=1&pv_ano_lectivo=2017"
creds = Credentials()

pprint(timetable.parse_timetables(creds, timetable_url))
# You'll get something like this:
{(datetime.date(2017, 9, 24), datetime.date(2017, 10, 28)): [{'class type': 'TP',
                                                              'classes': ('1MIEIC01
↪',),
                                                              'curricular unit':␣
↪CurricularUnit(399800),
                                                              'finish': datetime.
↪time(11, 0),
                                                              'room': ('B232A',),
                                                              'start': datetime.
↪time(9, 0),
                                                              'teachers':␣
↪(Teacher(212345),),
                                                              'weekday': 'Monday'},
                                                             {'class type': 'T',
                                                              'classes': ('1MIEIC01
↪',
                                                                          '1MIEIC02
↪',
                                                                          '1MIEIC03
↪',
                                                                          '1MIEIC04
↪',
                                                                          '1MIEIC05
↪',
                                                                          '1MIEIC06
↪',
                                                                          '1MIEIC07
↪',
                                                                          '1MIEIC08
↪'),
                                                              'curricular unit':␣
↪CurricularUnit(399800),
                                                              'finish': datetime.
↪time(13, 0),
                                                              'room': ('B003',),
                                                              'start': datetime.
↪time(11, 0),
                                                              'teachers':␣
↪(Teacher(212345),),
                                                              'weekday': 'Monday'},
                                                             {'class type': 'T',
                                                              'classes': ('1MIEIC01
↪',
                                                                          '1MIEIC02
↪',
                                                                          '1MIEIC03
↪',
                                                                          '1MIEIC04
↪',
                                                                          '1MIEIC05
↪',
```

(continues on next page)

(continued from previous page)

```
                                                                    '1MIEIC06
↪',
                                                                    '1MIEIC07
↪',
                                                                    '1MIEIC08
↪'),
                                                                'curricular unit':␣
↪CurricularUnit(399800),
                                                                'finish': datetime.
↪time(10, 0),
                                                                'room': ('B001',),
                                                                'start': datetime.
↪time(8, 30),
                                                                'teachers':␣
↪(Teacher(212345),
                                                                           ␣
↪Teacher(212345)),
                                                                'weekday': 'Tuesday'},
                                                                {'class type': 'T',
                                                                'classes': ('1MIEIC01
↪',
                                                                            '1MIEIC02
↪',
                                                                            '1MIEIC03
↪',
                                                                            '1MIEIC04
↪',
                                                                            '1MIEIC05
↪',
                                                                            '1MIEIC06
↪',
                                                                            '1MIEIC07
↪',
                                                                            '1MIEIC08
↪'),
                                                                'curricular unit':␣
↪CurricularUnit(399800),
                                                                'finish': datetime.
↪time(11, 30),
                                                                'room': ('B001',),
                                                                'start': datetime.
↪time(10, 0),
                                                                'teachers':␣
↪(Teacher(212345),),
                                                                'weekday': 'Tuesday'},
                                                                {'class type': 'TP',
                                                                'classes': ('1MIEIC02
↪',),
                                                                'curricular unit':␣
↪CurricularUnit(399800),
                                                                'finish': datetime.
↪time(13, 30),
                                                                'room': ('B110',),
                                                                'start': datetime.
↪time(11, 30),
                                                                'teachers':␣
↪(Teacher(212345),),
```

(continues on next page)

(continued from previous page)

```
                                                                'weekday': 'Tuesday'},
                                                                {'class type': 'T',
                                                                'classes': ('1MIEIC01
↪',
                                                                            '1MIEIC02
↪',
                                                                            '1MIEIC03
↪',
                                                                            '1MIEIC04
↪',
                                                                            '1MIEIC05
↪',
                                                                            '1MIEIC06
↪',
                                                                            '1MIEIC07
↪',
                                                                            '1MIEIC08
↪'),
                                                                'curricular unit':␣
↪CurricularUnit(399800),
                                                                'finish': datetime.
↪time(10, 0),
                                                                'room': ('B001',),
                                                                'start': datetime.
↪time(8, 30),
                                                                'teachers':␣
↪(Teacher(212345),),
                                                                'weekday': 'Wednesday
↪'},
                                                                {'class type': 'T',
                                                                'classes': ('1MIEIC01
↪',
                                                                            '1MIEIC02
↪',
                                                                            '1MIEIC03
↪',
                                                                            '1MIEIC04
↪',
                                                                            '1MIEIC05
↪',
                                                                            '1MIEIC06
↪',
                                                                            '1MIEIC07
↪',
                                                                            '1MIEIC08
↪'),
                                                                'curricular unit':␣
↪CurricularUnit(399800),
                                                                'finish': datetime.
↪time(11, 30),
                                                                'room': ('B001',),
                                                                'start': datetime.
↪time(10, 0),
                                                                'teachers':␣
↪(Teacher(212345),
                                                                           ␣
↪Teacher(212345)),
```

(continues on next page)

(continued from previous page)

```
                                                                'weekday': 'Wednesday
↪'},
                                                                {'class type': 'TP',
                                                                'classes': ('1MIEIC03
↪',),
                                                                'curricular unit':␣
↪CurricularUnit(399800),
                                                                'finish': datetime.
↪time(13, 30),
                                                                'room': ('B205',),
                                                                'start': datetime.
↪time(11, 30),
                                                                'teachers':␣
↪(Teacher(212345),),
                                                                'weekday': 'Wednesday
↪'},
                                                                ... ],
(datetime.date(2017, 10, 29), datetime.date(2017, 11, 4)): [{'class type': 'TP',
                                                                'classes': ('1MIEIC04
↪',),
                                                                'curricular unit':␣
↪CurricularUnit(399800),
                                                                'finish': datetime.
↪time(11, 0),
                                                                'room': ('B232A',),
                                                                'start': datetime.
↪time(9, 0),
                                                                'teachers':␣
↪(Teacher(212345),),
                                                                'weekday': 'Monday'},
                                                                {'class type': 'T',
                                                                'classes': ('1MIEIC01
↪',
                                                                            '1MIEIC02
↪',
                                                                            '1MIEIC03
↪',
                                                                            '1MIEIC04
↪',
                                                                            '1MIEIC05
↪',
                                                                            '1MIEIC06
↪',
                                                                            '1MIEIC07
↪',
                                                                            '1MIEIC08
↪'),
                                                                'curricular unit':␣
↪CurricularUnit(399800),
                                                                'finish': datetime.
↪time(13, 0),
                                                                'room': ('B003',),
                                                                'start': datetime.
↪time(11, 0),
                                                                'teachers':␣
↪(Teacher(212345),),
                                                                'weekday': 'Monday'},
```

(continues on next page)

(continued from previous page)

```
                                                                {'class type': 'T',
                                                                'classes': ('1MIEIC01
↪',
                                                                            '1MIEIC02
↪',
                                                                            '1MIEIC03
↪',
                                                                            '1MIEIC04
↪',
                                                                            '1MIEIC05
↪',
                                                                            '1MIEIC06
↪',
                                                                            '1MIEIC07
↪',
                                                                            '1MIEIC08
↪'),
                                                                'curricular unit':␣
↪CurricularUnit(399800),
                                                                'finish': datetime.
↪time(10, 0),
                                                                'room': ('B001',),
                                                                'start': datetime.
↪time(8, 30),
                                                                'teachers':␣
↪(Teacher(212345),
                                                                           ␣
↪Teacher(212345)),
                                                                'weekday': 'Tuesday'},
                                                                {'class type': 'T',
                                                                'classes': ('1MIEIC01
↪',
                                                                            '1MIEIC02
↪',
                                                                            '1MIEIC03
↪',
                                                                            '1MIEIC04
↪',
                                                                            '1MIEIC05
↪',
                                                                            '1MIEIC06
↪',
                                                                            '1MIEIC07
↪',
                                                                            '1MIEIC08
↪'),
                                                                'curricular unit':␣
↪CurricularUnit(399800),
                                                                'finish': datetime.
↪time(11, 30),
                                                                'room': ('B001',),
                                                                'start': datetime.
↪time(10, 0),
                                                                'teachers':␣
↪(Teacher(212345),),
                                                                'weekday': 'Tuesday'},
                                                                {'class type': 'TP',
```

(continues on next page)

(continued from previous page)

```
                                                                'classes': ('1MIEIC06
↪',),
                                                                'curricular unit':␣
↪CurricularUnit(399800),
                                                                'finish': datetime.
↪time(13, 30),
                                                                'room': ('B110',),
                                                                'start': datetime.
↪time(11, 30),
                                                                'teachers':␣
↪(Teacher(212345),),
                                                                'weekday': 'Tuesday'},
                                                                ...]
}
```

feupy.timetable.**parse_timetable**(*credentials: feupy._Credentials.Credentials*, *url: str*, *ignore_coherence: bool = False*) → list

Parses the events (including overlaps) of the timetable with the given url as a list of dictionaries.

Each dictionary represents an timetable event (a class) and has 8 keys:

key	value
"class type"	(str) E.g. "T", "TP", etc.
"classes"	(tuple(str)) The classes that are being taught
"curricular unit"	(*feupy.CurricularUnit*) The subject being teached
"finish"	(`datetime.time`) The finish time of the event
"room"	(tuple(str)) The rooms in which the event will take place
"start"	(`datetime.time`) The start time of the event
"teachers"	(tuple(*feupy.Teacher*))
"weekday"	(str)

Parameters

- **credentials** (*feupy.Credentials*) – A *feupy.Credentials* object
- **url** (*str*) – The url of the timetable page
- **ignore_coherence** (`bool`, optional) – Whether or not this function should raise a `CoherenceError` exception if the information of a class can not be fetched

Returns A list of dicts

Example:

```
from feupy import timetable, Credentials
from pprint import pprint

timetable_url = "https://sigarra.up.pt/feup/pt/hor_geral.turmas_view?pv_turma_
↪id=207783&pv_periodos=1&pv_ano_lectivo=2017"
creds = Credentials()

pprint(timetable.parse_timetable(creds, timetable_url))
# You'll get something like this:
[{'class type': 'TP',
'classes': ('1MIEIC01',),
'curricular unit': CurricularUnit(399999),
```

(continues on next page)

(continued from previous page)

```
'finish': datetime.time(11, 0),
'room': ('B232A',),
'start': datetime.time(9, 0),
'teachers': (Teacher(212345),),
'weekday': 'Monday'},
{'class type': 'T',
'classes': ('1MIEIC01',
            '1MIEIC02',
            '1MIEIC03',
            '1MIEIC04',
            '1MIEIC05',
            '1MIEIC06',
            '1MIEIC07',
            '1MIEIC08'),
'curricular unit': CurricularUnit(399999),
'finish': datetime.time(13, 0),
'room': ('B003',),
'start': datetime.time(11, 0),
'teachers': (Teacher(212345),),
'weekday': 'Monday'},
{'class type': 'T',
'classes': ('1MIEIC01',
            '1MIEIC02',
            '1MIEIC03',
            '1MIEIC04',
            '1MIEIC05',
            '1MIEIC06',
            '1MIEIC07',
            '1MIEIC08'),
'curricular unit': CurricularUnit(399999),
'finish': datetime.time(10, 0),
'room': ('B001',),
'start': datetime.time(8, 30),
'teachers': (Teacher(212345), Teacher(212345)),
'weekday': 'Tuesday'},
{'class type': 'T',
'classes': ('1MIEIC01',
            '1MIEIC02',
            '1MIEIC03',
            '1MIEIC04',
            '1MIEIC05',
            '1MIEIC06',
            '1MIEIC07',
            '1MIEIC08'),
'curricular unit': CurricularUnit(399999),
'finish': datetime.time(11, 30),
'room': ('B001',),
'start': datetime.time(10, 0),
'teachers': (Teacher(212345),),
'weekday': 'Tuesday'},
{'class type': 'TP',
'classes': ('1MIEIC02',),
'curricular unit': CurricularUnit(399999),
'finish': datetime.time(13, 30),
'room': ('B110',),
'start': datetime.time(11, 30),
'teachers': (Teacher(212345),),
```

(continues on next page)

(continued from previous page)

```
'weekday': 'Tuesday'},
... ]
```

CHAPTER 10

Indices and tables

- genindex
- modindex
- search

Python Module Index

f

Index