Vulcan API Documentation

Release 2.3.0

Kacper Ziubryniewicz

Feb 16, 2023

Contents

CHAPTER 1

Installation

You can install `vulcan-api` using `pip`

```
$ pip install vulcan-api
```

or you can build it yourself

```
$ git clone https://github.com/kapi2289/vulcan-api.git
$ cd vulcan-api
$ pip install .
```

CHAPTER 2

Getting started

2.1 Core concepts and definitions

In order to use the API, it's important to understand some concepts and naming conventions in the API.

- `symbol` - sometimes referred to as "partition symbol". This is a textual grouping symbol representing a group of e-register instances: a town, a county or a part of them. The symbol is present in the e-register website URL:

  ```
  https://uonetplus.vulcan.net.pl/<symbol>
  ```

- `code` - or "school code" - a code representing a single school or few grouped (in an unit) school buildings. Often in the form of `001234`, sometimes also containing alphabet characters. Present in the URL:

  ```
  https://uonetplus.vulcan.net.pl/<symbol>/<code>
  ```

- *`Unit`* - a group of schools, sharing a similar name. May contain only one school.
- *`School`* - a part of a `unit`.
- *`Keystore`* - login data for an instance of the API. **Might be tied (registered) to multiple accounts.**
- *`Account`* - an account from a single `symbol`, containing one or more `students`, accessed using a corresponding `keystore`.
- *`Student`* - a person, school attendant.

2.2 Technical info

The Vulcan API is asynchronous (using `asyncio`) and works using coroutines. All the code presented in this documentation needs to be placed inside a coroutine block (except imports, obviously).

A sample coroutine block looks as follows:

```
import asyncio

async def main():
    # asynchronous code goes here

if __name__ == "__main__":
    loop = asyncio.get_event_loop()
    loop.run_until_complete(main())
```

2.2.1 Data fetching

All data getting methods are asynchronous.

There are three return types of those methods:

- object - applies to methods returning a single object (e.g. the currently selected student, the today's lucky number, the server date-time)
- list - applies to *get_students()*. The list is either read from the server or the in-memory cache.
- *AsyncIterator* - applies to all other data fetching methods. The returned iterator may be used like this:

```
grades = await client.data.get_grades()

# with a for loop
async for grade in grades:
    print(grade)

# convert to a list
grades = [grade async for grade in grades]
print(grades[0])
for grade in grades:
    print(grade)
```

Note: You cannot re-use the AsyncIterator (once iterated through). As it is asynchronous, you also cannot use the next() method on it.

2.2.2 Sessions

As all HTTP requests are also async, the API uses `aiohttp`'s sessions, which need to be opened and closed when needed.

Upon creation, the *Vulcan* object creates a session, which needs to be closed before the program terminates.

```
client = Vulcan(keystore, account)
# use the client here
await client.close()
```

It is also possible to use a context manager to handle session opening and closing automatically.

```
client = Vulcan(keystore, account)
async with client:
    # use the client here
```

Warning: Be aware that every `with` block creates and closes a new session. As per the `aiohttp` docs, it is recommended to group multiple requests to use with a single session, so it's best not to use a separate `with` block for every single request.

2.3 Keystore creation

The first step is to create a *Keystore*, which will be used to access any account to which it's registered:

```
from vulcan import Keystore

keystore = Keystore.create()
# or with an explicitly passed device model
keystore = Keystore.create(device_model="Vulcan API")
```

The keystore is now ready to be registered in exchange for an *Account*, but it's best to save it for later use:

```
with open("keystore.json", "w") as f:
    # use one of the options below:
    # write a formatted JSON representation
    f.write(keystore.as_json)
    # dump a dictionary as JSON to file (needs `json` import)
    json.dump(keystore.as_dict, f)
```

A once-saved keystore may be simply loaded back into an API-usable object:

```
with open("keystore.json") as f:
    # use one of the options below:
    # load from a file-like object
    keystore = Keystore.load(f)
    # load from a JSON string
    keystore = Keystore.load(f.read())
    # load from a dictionary (needs `json` import)
    keystore = Keystore.load(json.load(f))
```

The keystore is now ready for further usage.

2.4 Account registration

It is now necessary to register the previously created *Keystore* in the e-register, in order to get access to the *Account*'s data.

The Token, Symbol and PIN need to be obtained from the Vulcan e-register student/parent panel (in the "Mobile access/Dostęp mobilny" tab):

```
from vulcan import Account

account = Account.register(keystore, token, symbol, pin)
```

Just as for the keystore, it's recommended to save the account credentials for later usage:

```
with open("account.json", "w") as f:
    # use one of the options below:
    # write a formatted JSON representation
    f.write(account.as_json)
    # dump a dictionary as JSON to file (needs `json` import)
    json.dump(account.as_dict, f)
```

An account may be loaded back as follows:

```
with open("account.json") as f:
    # use one of the options below:
    # load from a file-like object
    account = Account.load(f)
    # load from a JSON string
    account = Account.load(f.read())
    # load from a dictionary (needs `json` import)
    account = Account.load(json.load(f))
```

You are now ready to use the API. The keystore and account registration is a one-time step.

2.5 Basic client usage

To create the API client:

```
from vulcan import Vulcan

client = Vulcan(keystore, account)
```

To select a student:

```
await client.select_student()  # select the first available student
print(client.student)  # print the selected student

students = await client.get_students()
client.student = students[1]  # select the second student
```

2.5.1 Simple data fetching

All data is fetched from the *VulcanData* class, available as `client.data` variable.

Note: Read the *VulcanData* docs to see all public data fetching methods.

```
lucky_number = await client.data.get_lucky_number()
print(lucky_number)
```

2.5.2 Data fetching - technical info

All data getting methods are asynchronous.

There are three return types of those methods:

- object - applies to methods returning a single object (e.g. the currently selected student, the today's lucky number, the server date-time)
- list - applies to *get_students()*. The list is either read from the server or the in-memory cache.
- *AsyncIterator* - applies to all other data fetching methods. The returned iterator may be used like this:

```
grades = await client.data.get_grades()

# with a for loop
async for grade in grades:
    print(grade)

# convert to a list
grades = [grade async for grade in grades]
print(grades[0])
for grade in grades:
    print(grade)
```

Note: You cannot re-use the AsyncIterator (once iterated through). As it is asynchronous, you also cannot use the next() method on it.

CHAPTER 3

Full API documentation

3.1 Client

class vulcan.**Vulcan**(*keystore*, *account*, *session=None*, *logging_level: int = None*)
Vulcan API client.

Contains methods for getting/setting the current student and for setting the logging level. All data is fetched from an instance of the *VulcanData*, accessible using the `data` variable.

Variables **data** (VulcanData) – the data client

get_students(*cached=True*) → List[vulcan.model._student.Student]
Gets students assigned to this account.

Parameters **cached** (*bool*) – whether to allow returning the cached list

Return type List[*Student*]

select_student()
Load a list of students associated with the account. Set the first available student as default for the API.

static set_logging_level(*logging_level: int*)
Set the API logging level.

Parameters **logging_level** (*int*) – logging level from *logging* module

student
Gets/sets the currently selected student.

Return type *Student*

class vulcan._data.**VulcanData**(*api: vulcan._api.Api*)
A data client for the API.

Contains methods for getting all data objects, some in form of a list, others as an object. All the methods are asynchronous. Additionally, the list getting methods return an *AsyncIterator* of the items.

The data client shall not be constructed outside of the main API class.

get_addressbook(***kwargs*) → Union[AsyncIterator[vulcan.data._addressbook.Addressbook], List[int]]
Yields the addressbook.

Return type Union[AsyncIterator[*Addressbook*], List[int]]

get_attendance(*last_sync: datetime.datetime = None*, *deleted=False*, *date_from=None*, *date_to=None*, ***kwargs*) → Union[AsyncIterator[vulcan.data._attendance.Attendance], List[int]]
Fetches attendance from the given date

Parameters

- **last_sync** (*datetime.datetime*) – date of the last sync, gets only the objects updated since this date
- **deleted** (*bool*) – whether to only get the deleted item IDs
- **date_from** (*datetime.date*) – Date, from which to fetch attendance, if not provided it's using the today date (Default value = None)
- **date_to** (*datetime.date*) – Date, to which to fetch attendance, if not provided it's using the *date_from* date (Default value = None)

Return type Union[AsyncIterator[*Attendance*], List[int]]

get_changed_lessons(*last_sync: datetime.datetime = None*, *deleted=False*, *date_from=None*, *date_to=None*, ***kwargs*) → Union[AsyncIterator[vulcan.data._lesson.ChangedLesson], List[int]]
Yields the student's changed lessons.

Parameters

- **last_sync** (*datetime.datetime*) – date of the last sync, gets only the objects updated since this date
- **deleted** (*bool*) – whether to only get the deleted item IDs
- **date_from** (*datetime.date*) – Date, from which to fetch lessons, if not provided it's using the today date (Default value = None)
- **date_to** (*datetime.date*) – Date, to which to fetch lessons, if not provided it's using the *date_from* date (Default value = None)

Return type Union[AsyncIterator[*ChangedLesson*], List[int]]

get_exams(*last_sync: datetime.datetime = None*, *deleted=False*, ***kwargs*) → Union[AsyncIterator[vulcan.data._grade.Grade], List[int]]
Yields the student's exams.

Parameters

- **last_sync** (*datetime.datetime*) – date of the last sync, gets only the objects updated since this date
- **deleted** (*bool*) – whether to only get the deleted item IDs

Return type Union[AsyncIterator[*Exam*], List[int]]

get_grades(*last_sync: datetime.datetime = None*, *deleted=False*, ***kwargs*) → Union[AsyncIterator[vulcan.data._grade.Grade], List[int]]
Yields the student's grades.

Parameters

- **last_sync** (*datetime.datetime*) – date of the last sync, gets only the objects updated since this date
- **deleted** (*bool*) – whether to only get the deleted item IDs

Return type Union[AsyncIterator[*Grade*], List[int]]

get_homework(*last_sync: datetime.datetime = None*, *deleted=False*, ***kwargs*) → Union[AsyncIterator[vulcan.data._homework.Homework], List[int]]

Yields the student's homework.

Parameters

- **last_sync** (*datetime.datetime*) – date of the last sync, gets only the objects updated since this date
- **deleted** (*bool*) – whether to only get the deleted item IDs

Return type Union[AsyncIterator[*Homework*], List[int]]

get_lessons(*last_sync: datetime.datetime = None*, *deleted=False*, *date_from=None*, *date_to=None*, ***kwargs*) → Union[AsyncIterator[vulcan.data._lesson.Lesson], List[int]]

Yields the student's lessons.

Parameters

- **last_sync** (*datetime.datetime*) – date of the last sync, gets only the objects updated since this date
- **deleted** (*bool*) – whether to only get the deleted item IDs
- **date_from** (*datetime.date*) – Date, from which to fetch lessons, if not provided it's using the today date (Default value = None)
- **date_to** (*datetime.date*) – Date, to which to fetch lessons, if not provided it's using the *date_from* date (Default value = None)

Return type Union[AsyncIterator[*Lesson*], List[int]]

get_lucky_number(*day: datetime.date = None*) → vulcan.data._lucky_number.LuckyNumber

Gets the lucky number for the specified date.

Parameters **day** (*datetime.date*) – date of the lucky number to get. Defaults to `None` (today).

Return type *LuckyNumber*

get_message_boxes(***kwargs*) → AsyncIterator[vulcan.data._messagebox.MessageBox]

Yields message boxes.

Return type Union[AsyncIterator[`MessageBox`]

get_messages(*message_box: str*, *last_sync: datetime.datetime = None*, *folder=1*, ***kwargs*) → Union[AsyncIterator[vulcan.data._message.Message], List[int]]

Yields messages received in the specified message box.

Parameters

- **message_box** (*str*) – the MessageBox's Global Key to get the messages from, can be obtained from get_message_boxes
- **last_sync** (*datetime.datetime*) – date of the last sync, gets only the objects updated since this date
- **folder** (*int*) – message folder: 1 - received; 2 - sent; 3 - deleted

Return type Union[AsyncIterator[*Message*], List[int]]

get_time() → vulcan.model._datetime.DateTime
Gets the current server time.

Return type *DateTime*

3.2 Core models

class vulcan.**Keystore**(*certificate*, *fingerprint*, *private_key*, *firebase_token*, *device_model*)
A keystore containing of:

- a PEM-encoded X509 certificate signed using SHA-256 with RSA algorithm
- SHA-1 fingerprint of the certificate, represented as lowercase hexadecimal characters
- a PEM-encoded PKCS#8 RSA 2048 private key

Additionally, to use with the Vulcan API the keystore contains:

- a Firebase Cloud Messaging token - to re-use for every request
- a device name string, also needed for API requests

Variables

- **certificate** (*str*) – a PEM-encoded certificate
- **fingerprint** (*str*) – the certificate's fingerprint
- **private_key** (*str*) – a PEM-encoded RSA 2048 private key
- **firebase_token** (*str*) – an FCM token
- **device_model** (*str*) – a device model string

class vulcan.**Account**(*login_id*, *user_login*, *user_name*, *rest_url*)
An account in the e-register.

Variables

- **login_id** (*int*) – the account's login ID
- **user_login** (*str*) – the account's login name (email/username)
- **user_name** (*str*) – probably the same as user_login
- **rest_url** (*str*) – the API base URL for the partition symbol

class vulcan.model.**Serializable**
A base class allowing to (de)serialize objects easily into appropriate class variables.

as_dict
Serialize the object as a dictionary.

Return type dict

as_json
Serialize the object as a JSON string.

Return type str

classmethod load(*data*) → T
Deserialize provided `data` into an instance of `cls`.

The `data` parameter may be:

- a JSON string
- a dictionary
- a handle to a file containing a JSON string

Parameters **data** – the data to deserialize

3.3 Common models

class vulcan.model.**Student**(*class_*, *symbol*, *symbol_code*, *pupil*, *unit*, *school*, *periods*)
A student object, along with his school, class and period information

Variables

- **class_** (*str*) – student class
- **symbol** (*str*) – the "partition" symbol - can be a town or county name
- **symbol_code** (*str*) – the school unit code - often a 6 digit number
- **pupil** (Pupil) – contains the student's IDs, names and email
- **unit** (Unit) – info about the school unit (e.g. several school buildings)
- **school** (School) – info about the school (a single building of the unit)
- **periods** (*List[Period]*) – a list of the student's school year periods

current_period
Gets the currently ongoing period of the student.

Return type *Period*

full_name
Gets the student's full name in "FirstName SecondName LastName" format or "FirstName LastName" format if there is no second name.

Return type str

classmethod get(*api*, ***kwargs*) → List[vulcan.model._student.Student]

Return type List[*Student*]

period_by_id(*period_id: int*) → vulcan.model._period.Period
Gets a period matching the given period ID.

Parameters **period_id** (*int*) – the period ID to look for

Return type *Period*

class vulcan.model.**DateTime**(*timestamp*, *date*, *time*)
A date-time object used for representing points in time.

Variables

- **timestamp** (*int*) – number of millis since the Unix epoch
- **date** (*datetime.date*) – a date object
- **time** (*datetime.time*) – a time object

date_time
Combine the date and time of this object.

Return type datetime.datetime

classmethod get(*api*, ***kwargs*) → vulcan.model._datetime.DateTime

Return type *DateTime*

class vulcan.model.**Period**(*id*, *level*, *number*, *current: bool*, *last: bool*, *start*, *end*)
A school year period.

Variables

- **id** (*int*) – the period ID
- **level** (*int*) – a grade/level number
- **number** (*int*) – number of the period in the school year
- **current** (*bool*) – whether the period is currently ongoing
- **last** (*bool*) – whether the period is last in the school year
- **start** (DateTime) – the period start datetime
- **end** (DateTime) – the period end datetime

class vulcan.model.**Pupil**(*id*, *login_id*, *first_name*, *last_name*, *gender*, *second_name=None*, *login_value=None*)
A class containing the student's data.

Variables

- **id** (*int*) – pupil's ID
- **login_id** (*int*) – pupil's account login ID
- **login_value** (*str*) – pupil's account login name (email/username)
- **first_name** (*str*) – student's first name
- **second_name** (*str*) – student's second name, optional
- **last_name** (*str*) – student's last name / surname
- **gender** (*Gender*) – student's gender

class vulcan.model.**School**(*id*, *name*, *short_name*, *address=None*)
A single school building.

Variables

- **id** (*int*) – school ID
- **name** (*str*) – school full name
- **short_name** (*str*) – school short name
- **address** (*str*) – school address (location)

class vulcan.model.**Subject**(*id*, *key*, *name*, *code*, *position*)
A school subject.

Variables

- **id** (*int*) – subject ID
- **key** (*str*) – subject's key (UUID)
- **name** (*str*) – subject's name
- **code** (*str*) – subject's code (e.g. short name or abbreviation)

- **position** (*int*) – unknown, yet

class vulcan.model.**Teacher**(*id*, *name*, *surname*, *display_name*)
A teacher or other school employee.

Variables

- **id** (*int*) – teacher ID
- **name** (*str*) – teacher's name
- **surname** (*str*) – teacher's surname
- **display_name** (*str*) – teacher's display name

class vulcan.model.**TeamClass**(*id*, *key*, *display_name*, *symbol*)
A school class.

Variables

- **id** (*int*) – class ID
- **key** (*str*) – class's key (UUID)
- **display_name** (*str*) – class's display name
- **symbol** (*str*) – class's symbol (e.g. a letter after the level, "C" in "6C")

class vulcan.model.**TeamVirtual**(*id*, *key*, *shortcut*, *name*, *part_type*)
A virtual team, i.e. a part of the school class. Often called a "distribution" of the class.

Variables

- **id** (*int*) – team ID
- **key** (*str*) – team's key (UUID)
- **shortcut** (*str*) – team's short name
- **name** (*str*) – team's name
- **part_type** (*str*) – type of the distribution

class vulcan.model.**TimeSlot**(*id*, *from_*, *to*, *displayed_time*, *position*)
Lesson time (start-end range)

Variables

- **id** (*int*) – lesson time ID
- **from_** (*datetime.time*) – lesson start time
- **to** (*datetime.time*) – lesson end time
- **displayed_time** (*str*) – lesson's displayed time
- **position** (*int*) – lesson position

class vulcan.model.**Unit**(*id*, *code*, *name*, *short_name*, *display_name*, *rest_url*, *address=None*)
A group of one or more schools.

Variables

- **id** (*int*) – unit ID
- **code** (*str*) – unit code (school code) - often 6 digits
- **name** (*str*) – unit full name
- **short_name** (*str*) – unit short name

- **display_name** (*str*) – unit display name
- **address** (*str*) – unit address (location)
- **rest_url** (*str*) – unit data's API base URL

3.4 Data models

class vulcan.data.**Addressbook**(*id*, *login_id*, *first_name*, *last_name*, *initials*, *roles*)
An address book.

Variables

- **id** (*str*) – recipient id
- **login_id** (*str*) – recipient login id
- **first_name** (*str*) – recipient's first name
- **last_name** (*str*) – recipient's last name
- **initials** (*str*) – recipient's initials
- **roles** (*list[Role]*) – recipient's role (eg. Teacher)

classmethod get(*api*, ***kwargs*) → Union[AsyncIterator[vulcan.data._addressbook.Addressbook], List[int]]

Return type Union[AsyncIterator[*Addressbook*], List[int]]

class vulcan.data.**Role**(*role_name*, *role_order*, *address_name*, *address_hash*, *first_name*, *last_name*, *initials*, *unit_symbol=None*, *constituent_unit_symbol=None*, *class_symbol=None*)
A role of addressee.

Variables

- **role_name** (*str*) – role name
- **role_order** (*int*) – role order
- **address_name** (*str*) – address name
- **address_hash** (*str*) – address hash
- **first_name** (*str*) – recipient's first name
- **last_name** (*str*) – recipient's last name
- **initials** (*str*) – recipient's initials
- **unit_symbol** (*str*) – recipient's unit_symbol
- **constituent_unit_symbol** (*str*) – recipient's constituent unit symbol
- **class_symbol** (*str*) – recipient's class symbol

class vulcan.data.**Attendance**(*lesson_id*, *id*, *lesson_number*, *global_key*, *lesson_class_id*, *lesson_class_global_key*, *calculate_presence: bool*, *replacement: bool*, *subject=None*, *topic=None*, *teacher=None*, *second_teacher=None*, *main_teacher=None*, *team_class=None*, *class_alias=None*, *date=None*, *time=None*, *date_modified=None*, *aux_presence_id=None*, *justification_status=None*, *presence_type=None*, *note=None*, *public_resources=None*, *remote_resources=None*, *group=None*, *visible=None*)

Attendance.

Variables

- **lesson_id** (*int*) – lesson ID
- **id** (*int*) – attendance ID
- **lesson_number** (*int*) – lesson number
- **global_key** (*str*) – attendance global key
- **lesson_class_id** (*int*) – lesson class ID
- **global_key** – lesson class global key
- **calculate_presence** (*bool*) – does it count for absences
- **replacement** (*bool*) – os it replaced
- **subject** (Subject) – subject of the lesson
- **topic** (*str*) – topic of the lesson
- **teacher** (Teacher) – teacher of the lesson
- **second_teacher** (Teacher) – second teacher of the lesson
- **main_teacher** (Teacher) – pupil main teacher
- **team_class** (TeamClass) – the class that had lesson
- **class_alias** (*str*) – class short name
- **date** (DateTime) – lesson's date
- **time** (TimeSlot) – lesson's time
- **date_modified** (DateTime) – attendance modification date, if not modified it is created date
- **id** – aux presence ID
- **justification_status** (*str*) – attendance justification status
- **presence_type** (PresenceType) – presence type
- **note** (*str*) – attendance note
- **public_resources** (*str*) – attendance public resources
- **remote_resources** (*str*) – attendance remote resources
- **group** (TeamVirtual) – group, that has the lesson
- **visible** (*bool*) – attendance visibility

classmethod get(*api*, *last_sync*, *deleted*, *date_from*, *date_to*, ***kwargs*) → Union[AsyncIterator[vulcan.data._attendance.Attendance], List[int]]

Return type Union[AsyncIterator[*Attendance*], List[int]]

class vulcan.data.**PresenceType**(*id*, *name*, *symbol*, *category_id*, *category_name*, *position*, *presence: bool*, *absence: bool*, *exemption: bool*, *late: bool*, *justified: bool*, *deleted: bool*)

Presence type

Variables

- **id** (*int*) – attendance ID

- **name** (*str*) – attendance name
- **symbol** (*str*) – attendance symbol
- **category_id** (*int*) – attendance category ID
- **category_name** (*str*) – attendance category name
- **position** (*int*) – attendance position
- **presence** (*bool*) – presence on lesson
- **absence** (*bool*) – absence on lesson
- **exemption** (*bool*) – exemption from lesson
- **late** (*bool*) – is late for lesson
- **justified** (*bool*) – justified absence
- **deleted** (*bool*) – whether the entry is deleted

class vulcan.data.**Exam**(*id*, *key*, *type*, *topic*, *date_created*, *date_modified*, *deadline*, *creator*, *subject*, *team_class=None*, *team_virtual=None*)

An exam or short quiz.

Variables

- **id** (*int*) – exam's ID
- **key** (*str*) – exam's key (UUID)
- **type** (*str*) – exam's type
- **topic** (*str*) – exam's topic
- **date_created** (DateTime) – exam's creation date
- **date_modified** (DateTime) – exam's modification date (may be the same as date_created if it was never modified)
- **deadline** (DateTime) – exam's date and time
- **creator** (Teacher) – the teacher who added the exam
- **subject** (Subject) – the exam's subject
- **team_class** (TeamClass) – the class taking the exam
- **team_virtual** (TeamVirtual) – the class distribution taking the exam, optional

classmethod get(*api*, *last_sync*, *deleted*, ***kwargs*) → Union[AsyncIterator[vulcan.data._exam.Exam], List[int]]

Return type Union[AsyncIterator[*Exam*], List[int]]

class vulcan.data.**Homework**(*id*, *key*, *homework_id*, *content*, *date_created*, *creator*, *subject*, *attachments*, *is_answer_required: vulcan.model._subject.Subject*, *deadline*, *answer_deadline=None*, *answer_date=None*)

A homework.

Variables

- **id** (*int*) – homework's external ID
- **key** (*str*) – homework's key (UUID)
- **homework_id** (*int*) – homework's internal ID
- **content** (*str*) – homework's content

- **date_created** (*DateTime*) – homework's creation date
- **creator** (*Teacher*) – the teacher who added the homework
- **subject** (*Subject*) – the homework's subject
- **attachments** (*List[Attachment]*) – attachments added to homework
- **is_answer_required** (*bool*) – Is an answer required
- **deadline** (*DateTime*) – homework's date and time
- **answer_deadline** (*DateTime*) – homework's answer deadline
- **answer_date** (*DateTime*) – homework's answer date and time

classmethod get(*api*, *last_sync*, *deleted*, ***kwargs*) → Union[AsyncIterator[vulcan.data._homework.Homework], List[int]]

Return type Union[AsyncIterator[Homework], List[int]]

class vulcan.data.**Lesson**(*id=None*, *date=None*, *time=None*, *room=None*, *teacher=None*, *second_teacher=None*, *subject=None*, *event=None*, *changes=None*, *team_class=None*, *pupil_alias=None*, *group=None*, *visible: bool = None*)

A lesson.

Variables

- **id** (*int*) – lesson's ID
- **date** (DateTime) – lesson's date
- **time** (TimeSlot) – lesson's time
- **room** (LessonRoom) – classroom, in which is the lesson
- **teacher** (Teacher) – teacher of the lesson
- **second_teacher** (Teacher) – second teacher of the lesson
- **subject** (Subject) – subject on the lesson
- **event** (*str*) – an event happening during this lesson
- **changes** (LessonChanges) – lesson changes
- **team_class** (TeamClass) – the class that has the lesson
- **pupil_alias** (*str*) – pupil alias
- **group** (TeamVirtual) – group, that has the lesson
- **visible** (*bool*) – lesson visibility (whether the timetable applies to the given student)

classmethod get(*api*, *last_sync*, *deleted*, *date_from*, *date_to*, ***kwargs*) → Union[AsyncIterator[vulcan.data._lesson.Lesson], List[int]]

Return type Union[AsyncIterator[*Lesson*], List[int]]

class vulcan.data.**ChangedLesson**(*id=None*, *unit_id=None*, *schedule_id=None*, *lesson_date=None*, *note=None*, *reason=None*, *time=None*, *room=None*, *teacher=None*, *second_teacher=None*, *subject=None*, *event=None*, *changes=None*, *change_date=None*, *team_class=None*, *group=None*)

Changed lesson.

Variables

- **id** (*int*) – changed lesson's ID
- **unit_id** (*int*) – unit ID
- **schedule_id** (*int*) – normal lesson's ID
- **lesson_date** (DateTime) – lesson's date
- **change_date** (DateTime) – change date
- **time** (TimeSlot) – lesson's time
- **note** (*str*) – change note
- **reason** (*str*) – change reason
- **room** (LessonRoom) – classroom, in which is the lesson
- **teacher** (Teacher) – teacher of the lesson
- **second_teacher** (Teacher) – second teacher of the lesson
- **subject** (Subject) – subject on the lesson
- **event** (*str*) – an event happening during this lesson
- **changes** (LessonChanges) – lesson changes
- **team_class** (TeamClass) – the class that has the lesson
- **group** (TeamVirtual) – group, that has the lesson

classmethod get(*api*, *last_sync*, *deleted*, *date_from*, *date_to*, ***kwargs*) → Union[AsyncIterator[vulcan.data._lesson.Lesson], List[int]]

Return type Union[AsyncIterator[ChangeLesson], List[int]]

class vulcan.data.**LessonChanges**(*id*, *type*, *separation: bool*)

Lesson changes

Variables

- **id** (*int*) – lesson change ID
- **type** (*int*) – lesson change type
- **code** (*bool*) – team separation

class vulcan.data.**LessonRoom**(*id*, *code*)

Lesson room

Variables

- **id** (*int*) – lesson room ID
- **code** (*str*) – classroom code

class vulcan.data.**Grade**(*id*, *key*, *pupil_id*, *content_raw*, *content*, *date_created*, *date_modified*, *teacher_created*, *teacher_modified*, *column*, *value=None*, *comment=None*, *numerator=None*, *denominator=None*)

A grade.

Variables

- **id** (*int*) – grade's ID
- **key** (*str*) – grade's key (UUID)
- **pupil_id** (*int*) – the related pupil's ID

- **content_raw** (*str*) – grade's content (with comment)
- **content** (*str*) – grade's content (without comment)
- **date_created** (DateTime) – grade's creation date
- **date_modified** (DateTime) – grade's modification date (may be the same as date_created if it was never modified)
- **teacher_created** (Teacher) – the teacher who added the grade
- **teacher_modified** (Teacher) – the teacher who modified the grade
- **column** (GradeColumn) – grade's column
- **value** (*float*) – grade's value, may be *None* if 0.0
- **comment** (*str*) – grade's comment, visible in parentheses in content_raw
- **numerator** (*float*) – for point grades: the numerator value
- **denominator** (*float*) – for point grades: the denominator value

classmethod get(*api*, *last_sync*, *deleted*, ***kwargs*) → Union[AsyncIterator[vulcan.data._grade.Grade], List[int]]

Return type Union[AsyncIterator[*Grade*], List[int]]

class vulcan.data.**GradeColumn**(*id*, *key*, *period_id*, *name*, *code*, *number*, *weight*, *subject*, *group=None*, *category=None*, *period=None*)

A grade column. Represents a topic which a student may get a grade from (e.g. a single exam, short test, homework).

Variables

- **id** (*int*) – grade column's ID
- **key** (*str*) – grade column's key (UUID)
- **period_id** (*int*) – ID of the period when the grade is given
- **name** (*str*) – grade column's name (description)
- **code** (*str*) – grade column's code (e.g. short name or abbreviation)
- **group** (*str*) – unknown, yet
- **number** (*int*) – unknown, yet
- **weight** (*int*) – weight of this column's grades
- **subject** (Subject) – the subject from which grades in this column are given
- **category** (GradeCategory) – category (base type) of grades in this column
- **period** (Period) – a resolved period of this grade

class vulcan.data.**GradeCategory**(*id*, *name*, *code*)

A base grade category. Represents a generic type, like an exam, a short test, a homework or other ("current") grades.

Variables

- **id** (*int*) – grade category's ID
- **name** (*str*) – grade category's name
- **code** (*str*) – grade category's code (e.g. short name or abbreviation)

class vulcan.data.**Message**(*id*, *global_key*, *thread_key*, *subject*, *content*, *sent_date*, *status*, *sender*, *receivers*, *attachments*, *read_date=None*)

A message.

Variables

- **id** (*str*) – Message id
- **global_key** (*str*) – Message Global Key
- **thread_key** (*str*) – Message thread key
- **subject** (*str*) – Subject of the message
- **content** (*str*) – Message content
- **sent_date** (*DateTime*) – Date with time when the message was sent
- **read_date** (*DateTime*) – Date with time when the message was read
- **status** (*int*) – Message status
- **sender** (*Address*) – Sender of the message
- **receivers** (*List[Address]*) – Receiver of the message
- **attachments** (*List[Attachment]*) – attachments added to message

classmethod get(*api*, *message_box*, *last_sync*, *folder*, ***kwargs*) → Union[AsyncIterator[vulcan.data._message.Message], List[int]]

Return type Union[AsyncIterator[*Message*], List[int]]

class vulcan.data.**Address**(*global_key*, *name*, *has_read=None*)

An address - "descriptor" used in the system containing the user's Global Key, his names and a information whether the user has read the message.

Variables

- **global_key** (*str*) – Global Key
- **name** (*str*) – address name
- **has_read** (*int*) – whether the user has read the message

class vulcan.data.**LuckyNumber**(*date*, *number*)

A lucky number for the specified date.

Variables

- **date** (*datetime.date*) – lucky number date
- **number** (*int*) – the lucky number

classmethod get(*api*, *day: _CountingAttr(counter=290*, *_default=NOTHING*, *repr=True*, *eq=True*, *order=True*, *hash=None*, *init=True*, *on_setattr=None*, *alias=None*, *metadata={'formatter': '%Y-%m-%d'*, *'key': 'Day'})*, ***kwargs*) → vulcan.data._lucky_number.LuckyNumber

Return type *LuckyNumber*

Index