

Введение

Сервер Mobile VyatSU реализует интерфейс API

API (application programming interface) — это посредник между разработчиком приложений и какой-либо средой, с которой это приложение должно взаимодействовать. API упрощает создание кода, поскольку предоставляет набор готовых классов, функций или структур для работы с имеющимися данными.

API Mobile VyatSU — это интерфейс, который позволяет получать информацию из базы данных с помощью http-запросов к специальному серверу. Вам не нужно знать в подробностях, как устроена база, из каких таблиц и полей каких типов она состоит — достаточно того, что API-запрос об этом «знает». Синтаксис запросов и тип возвращаемых ими данных строго определены на стороне самого сервиса.

Например, для получения данных о студенте с идентификатором 1 необходимо составить запрос такого вида:

```
http://127.0.0.1:8080/api/student.get?id=1
```

Запросы к API

Чтобы обратиться к методу API, Вам необходимо выполнить GET запрос такого вида:

```
http://127.0.0.1:8080/api/student.get?id=1
```

В данном запросе мы пытаемся получить информацию о студенте с идентификатором 1.

Разберем его по частям:

- http:// — протокол соединения
- 127.0.0.1:8080/api/ — адрес API сервиса
- student.get — метод сервиса
- ?id=1 — параметры запроса.

В общем виде все запросы выглядят так:

```
http://127.0.0.1:8080/api/<Method name>?<Parameters>
```

- Method Name (обязательно) — название метода API, к которому Вы хотите обратиться. Полный список методов доступен в разделе методов.
- Parameters (опционально) — входные параметры соответствующего метода API, последовательность пар name=value, разделенных амперсандом. Список параметров указан на странице с описанием метода.

В качестве ответа сервер возвращает JSON-объект с запрошенными данными (или сообщение об ошибке, если что-то пошло не так).

JSON — это формат записи данных в виде пар «имя свойства»: «значение»

В ответ на запрос

```
http://127.0.0.1:8080/api/<Method name>?<Parameters>
```

Сервер вернет такой ответ

```
{
  "id": 1,
  "firstName": "Даниил",
  "secondName": "Щесняк",
  "middleName": "Сергеевич",
  "groupID": 639,
  "groupShortName": "ИВТ6-32",
  "levelOfAccess": 0
}
```

Структура ответа каждого метода также строго задана, и при работе с API Вы заранее знаете, что в поле id придет число, а в поле firstName — строка. Такие правила оговариваются на страницах с описанием метода и соответствующих объектов, которые он возвращает в ответе.

Основные структуры

В данном разделе представленно описание основных структур:

- Student
- Teacher
- Group
- Schedule
- Announcement

Student

Данная структура описывает пользователя, который является студентом. Данная сущность может состоять в учебной группе и иметь определенные права доступа

Возможности студента:

- Получать расписание
- Читать и писать сообщения, прикрепленные к учебным парам
- Сменить учебную группу

Структура Student

Элемент	Тип	Значение
id	Число	Уникальный идентификатор студента
firstName	Строка	Имя студента
secondName	Строка	Фамилия студента
middleName	Строка	Отчество студента
groupId	Число	Уникальный идентификатор группы
groupShortName	Строка	Короткое имя группы
levelOfAccess	Число	Уровень прав студента

Каждый студент имеет различный уровень прав в группе.

Номер	Значение	Возможности
0	Не авторизованный	Нет никаких возможностей
1	Только чтение	Возможно только читать сообщения, прикрепленные к парам
2	Студент	Возможно писать и читать сообщения, повышать до 1-го уровня привелегий
3	Староста	Возможно писать и читать сообщения, повышать до 2-го уровня привелегий

Teacher

Данная структура описывает пользователя, который является преподавателем.

Возможности преподавателя:

- Получать расписание
- Читать и писать сообщения, прикрепленные к учебным парам
- Менять права студентов в учебных группах
- Добавлять учебные пары в расписаниях

Структура Teachers

Элемент	Тип	Значение
id	Число	Уникальный идентификатор преподавателя
firstName	Строка	Имя студента
secondName	Строка	Фамилия студента
middleName	Строка	Отчество студента
departmentID	Число	Уникальный идентификатор кафедры

Group

Данная структура описывает учебную группу. В учебной группе могут состоять только студенты

Структура Group

Название	Тип	Назначение
id	Число	Уникальный идентификатор группы
nameShort	Строка	Короткое имя группы
nameLong	Строка	Длинное имя группы
degree	Строка	Ученая степень группы (<i>Бакалавр, Специалитет, Магистратура, Аспирантура</i>)
formOfEducation	Строка	Форма обучения группы (<i>Очная, Заочная, Дистанционная</i>)

Schedule

Данная структура описывает учебное занятие. У учебного занятия может быть только один преподаватель, но несколько групп.

Структура Schedule

Название	Тип	Назначение
id	Число	Уникальный идентификатор занятия
teacherID	Число	Уникальный идентификатор преподавателя
firstName	Строка	Имя преподавателя
secondName	Строка	Фамилия преподавателя
middleName	Строка	Отчество преподавателя
subjectID	Число	Уникальный идентификатор предмета
subjectName	Строка	Название учебного предмета
nPair	Число	Порядковый номер учебной пары начиная от первого дня первой недели. недели
type	Число	Тип пары (<i>Лекция, Практика, Лабораторная работа</i>)
room	Строка	Номер учебного корпуса и номер аудитории
date	Строка	Дата проведения пары. Не обязательный параметр, который указывается только тогда, когда пара поставлена самим преподавателем
groups	Список	Список групп, у которых будет проходить заданная пара

Пары подразделяются на несколько типов:

Тип	Значение
0	Лекция
1	Практика
2	Лабораторная работа

Announcement

Данная структура описывает сообщение прикрепленное к учебной паре

Структура Announcement

Название	Тип	Назначение
id	Число	Уникальный идентификатор сообщения
scheduleID	Число	Уникальный идентификатор пары
firstName	Строка	Имя
secondName	Строка	Фамилия
middleName	Строка	Отчество
text	Строка	Сообщение
announceTime	Строка	Время написания сообщения
pairDate	Строка	День проведения учебной пары

Коды состояния

В зависимости от результатов обработки запроса будут возвращены различные коды состояния.

Таблица кодов состояния

Код состояния	Значение	Ошибки
200	Метод исполнился успешно	-
400	Неверный запрос	Неверный формат параметра Параметр не был передан
403	Доступ запрещен	Неверный логин или пароль Доступ запрещен
409	Конфликт	Пара уже существует Пользователь уже сущесвует

Ошибки

В данном разделе описываются ошибки, которые возникают во время вызова метода

Ошибка: Доступ запрещен

Данная ошибка случается при передаче неправильного ключа доступа пользователя, либо при попытке обратиться к методу, который недоступен для данного пользователя

Параметры

Название	Назначение	Значение
codeError	Код ошибки	5
errorName	Имя ошибки	Ошибка доступа
errorText	Текст ошибки	Доступ запрещен

Пример

Ответ сервера

```
{
  "errorCode": 5,
  "errorName": "Ошибка доступа",
  "errorText": "Доступ запрещен"
}
```

Ошибка: Ошибка аутентификации

Данная ошибка случается при передаче в метод `user.signIn` неправильного логина или пароля

Параметры

Название	Назначение	Значение
<code>codeError</code>	Код ошибки	4
<code>errorName</code>	Имя ошибки	Ошибка аутентификации
<code>errorText</code>	Текст ошибки	Неверно заданы параметры логин или пароль

Пример

Ответ сервера

```
{
  "errorCode": 4,
  "errorName": "Ошибка аутентификации",
  "errorText": "Неверно заданы параметры логин или пароль"
}
```

Ошибка: Неправильный формат

Данная ошибка сообщает о неправильном формате одного из переданных параметров

Параметры

Название	Назначение	Значение
codeError	Код ошибки	2
errorName	Имя ошибки	Ошибка формата
parameterName	Имя неправильного параметра	-
errorText	Текст ошибки	Ошибка. Неверный формат параметра

Пример

Ответ сервера

```
{
  "parameterName": "login",
  "errorCode": 2,
  "errorName": "Ошибка формата",
  "errorText": "Ошибка. Неверный формат параметра login"
}
```

Ошибка: Отсутствует параметр

Данная ошибка сообщает об отсутствии обязательного параметра при совершении запроса к методу

Параметры

Название	Назначение	Значение
codeError	Код ошибки	1
errorName	Имя ошибки	Параметр отсутствует
parameterName	Имя отсутствующего параметра	-
errorText	Текст ошибки	Отсутствует обязательный параметр

Пример

Ответ сервера

```
{
  "parameterName": "password",
  "errorCode": 1,
  "errorName": "Ошибка отсутствия параметра",
  "errorText": "Отсутствует обязательный параметр password"
}
```

Ошибка: Пара уже существует

Данная ошибка возникает при использовании метода [schedule.add](#), если в данное время у преподавателя есть учебная пара

Параметры

Название	Назначение	Значение
codeError	Код ошибки	6
errorName	Имя ошибки	Пара уже существует
errorText	Текст ошибки	Ошибка. У данного преподавателя в это время уже есть пара

Пример

Ответ сервера

```
{
  "errorCode": 6,
  "errorName": "Пара уже существует",
  "errorText": "Ошибка. У данного преподавателя в это время уже есть пара"
}
```

Ошибка: Пара отсутствует

Данная ошибка возникает при использовании метода [schedule.del](#), если в данное время у заданной учебной группы нет учебной пары

Параметры

Название	Назначение	Значение
codeError	Код ошибки	7
errorName	Имя ошибки	Пара не найдена
errorText	Текст ошибки	Ошибка. У данной группы в этой время нет пары

Пример

Ответ сервера

```
{
  "errorCode": 7,
  "errorName": "Пара не найдена",
  "errorText": "Ошибка. У данной группы в этой время нет пары"
}
```

Ошибка: Пользователь уже существует

Данная ошибка возникает при использовании метода `user.signup`, если данный пользователь уже существует в системе

Параметры

Название	Назначение	Значение
<code>codeError</code>	Код ошибки	3
<code>errorName</code>	Имя ошибки	Ошибка существующего пользователя
<code>errorText</code>	Текст ошибки	Ошибка. Данный пользователь уже существует

Пример

Ответ сервера

```
{
  "errorCode": 3,
  "errorName": "Ошибка существующего пользователя",
  "errorText": "Ошибка. Данный пользователь уже существует"
}
```


Методы

В данном разделе описываются методы, их параметры и возвращаемые значения

Метод: announcement.add

Данный метод позволяет прикрепить сообщение к учебной паре

- Сообщения может прикрепить только преподаватель или студент, который имеет уровень привелегий выше 1
- Длина сообщения не может быть более 255 символов

Параметры

Название	Назначение
token	Уникальный ключ пользователя Строка
id	Уникальный идентификатор учебной пары Положительное число
text	Текст сообщения Строка
day	День проведения пары Число
month	Месяц проведения пары Число
year	Год проведения пары Число

Ответ запроса

В качестве ответа возвращается код статуса 200, который сообщает, что метод выполнен успешно, иначе возвращается JSON объект ошибки

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
token	niy4vkrlif2mmpbpdlixx3q1
id	4017
text	Последний день сдачи пояснительных записок
day	29
month	12
year	2016

Ответ сервера

```
statusCode: 200
```

Метод: announcement.del

Данный метод позволяет удалить сообщение пользователя

- Сообщение может удалить пользователь, который его создал
- Возможно удалить сообщение студента, если его удаляет преподаватель или студент с более высоким уровнем доступа

Параметры

Название	Назначение
token	Уникальный ключ пользователя Строка
id	Уникальный идентификатор сообщения Положительное число

Ответ запроса

В качестве ответа возвращается код статуса 200, который сообщает, что метод выполнен успешно, иначе возвращается JSON объект ошибки

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
token	niy4vkrlif2mmpbpdlixx3q1
id	1

Ответ сервера

```
statusCode: 200
```

Метод: announcement.get

Данный метод позволяет получить последние сообщения, прикрепленные к учебной паре.

- Сообщения может получить только преподаватель или студент, который имеет уровень привелегий выше 0

Параметры

Название	Назначение
token	Уникальный ключ пользователя Строка
count	Количество последних сообщений, которые нужно получить

Ответ запроса

В качестве ответа возвращается json список из объектов с заданными полями:

Название	Назначение
id	Уникальный идентификатор сообщения Положительное число
scheduleID	Уникальный идентификатор пары Положительное число
firstName	Имя Строка
secondName	Фамилия Строка
middleName	Отчество Строка
text	Сообщение Строка
announceTime	Время написания сообщения Строка
pairDate	День проведения учебной пары Строка

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
token	niy4vkrlif2mmpbpdlixx3q1
count	1

Ответ сервера

```
[
  {
    "id": 1,
```

```
"scheduleID": 4017,  
"firstName": "Геннадий",  
"secondName": "Чистяков",  
"middleName": "Андреевич",  
"text": "Последний день сдачи пояснительных записок",  
"announceTime": "2016-12-17 15:36:25"  
"pairDate": "2016-12-29"  
}  
]
```

Метод: deparment.changeSecretKey

Данный метод позволяет изменить **любому** преподавателю секретный ключ своей кафедры.
Данный метод доступен только для авторизованных преподавателей

Параметры

Название	Назначение
token	Уникальный ключ доступа пользователя Строка
secretKey	Новый секретный ключ для регистрации преподаватлей на данной кафедре. Ключ не более 50 символов Строка

Ответ запроса

В качестве ответа возвращается успешный код статуса (200) или ошибка, если во время выполнения метода произошла ошибка.

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
token	niy4vkrlif2mmpbpdlixx3q1
secretKey	VerySecretKey!

Ответ сервера

statuscode: 200

Метод: group.changeAccess

Данный метод позволяет изменить права студента внутри группы.
Права студента может изменять **любой** преподаватель. Права студента так же может изменить и студент, если он состоит в этой же группе и его права выше.
Подпробнее о правах доступа можно прочитатть в [данном разделе](#)

Параметры

Название	Назначение
studentID	Положительное число, которое характеризует уникальный идентификатор студента. Положительное число
access	Число от 0 до 3, которое обозначает уровень прав студента. Положительное число
token	Уникальный ключ пользователя Строка

Ответ запроса

В качестве ответа код статуса 200, если метод успешно выполнен и ошибка, если в процессе выполнения метода произошли исключения.

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
id	1
access	3
token	niy4vkrlif2mmpbpdlix3q1

Ответ сервера

```
statusCode: 200
```

Метод: group.get

Данный метод позволяет получить информации о заданной группе.
Возвращает информацию о группе с заданным **id**

Параметры

Название	Назначение
id	Положительное число, которое характеризует уникальный идентификатор учебной группы. Целое число

Ответ запроса

В качестве ответа возвращается json объект с заданными полями:

Название	Назначение
id	Уникальный идентификатор группы Положительное число
nameShort	Короткое имя группы Строка
nameLong	Длинное имя группы Строка
degree	Ученая степень группы (<i>Бакалавр, Специалитет, Магистратура, Аспирантура</i>) Строка
formOfEducation	Форма обучения группы (<i>Очная, Заочная, Дистанционная</i>) Строка

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
id	639

Ответ сервера

```
{
  "id": 639,
  "nameShort": "ИВТ6-32",
  "nameLong": "ИВТ6-3302-02-00",
  "degree": "Бакалавр",
  "formOfEducation": "Очная"
}
```


Метод: group.getMembers

Данный метод возвращает список студентов, которые обучаются в заданной учебной группы.
Возвращает информацию о студентах обучающихся в группе с заданным **id**

Параметры

Название	Назначение
id	Положительное число, которое характеризует уникальный идентификатор группы. Положительное число

Ответ запроса

В качестве ответа возвращается json список состоящий из объектов с заданными полями:

Название	Назначение
id	Уникальный идентификатор студента Положительное число
firstName	Имя пользователя Строка
secondName	Фамилия Строка
middleName	Отчество Строка
groupID	Уникальный идентификатор учебной группы Положительное число
groupShortName	Короткое имя группы Строка
levelOfAccess	Права доступа в группе. Подробнее о правах доступа можно прочитать в данном разделе Положительное число

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
id	639

Ответ сервера

```
[
  {
    "id": 1,
    "firstName": "Даниил",
    "secondName": "Щесняк",
    "middleName": "Сергеевич",
    "groupID": 639,
    "groupShortName": "ИВТ6-32",
    "levelOfAccess": 0
  },
  {
    "id": 2,
    "firstName": "Никита",
```

```
"secondName": "Братухин",
"middleName": "",
"groupID": 639,
"groupShortName": "ИВТ6-32",
"levelOfAccess": 0
},
{
  "id": 3,
  "firstName": "Алексей",
  "secondName": "Губин",
  "middleName": "",
  "groupID": 639,
  "groupShortName": "ИВТ6-32",
  "levelOfAccess": 0
}
]
```

Метод: group.search

Данный метод возвращает список групп, которые соответствуют поисковому запросу.
Возвращает информацию о группах, которые имеют короткое имя как в параметре **name**

Параметры

Название	Назначение
name	Поисковый запрос. Строка

Ответ запроса

В качестве ответа возвращается json список из объектов с заданными полями:

Название	Назначение
id	Уникальный идентификатор группы Положительное число
nameShort	Короткое имя группы Строка
nameLong	Длинное имя группы Строка
degree	Ученая степень группы (<i>Бакалавр, Специалитет, Магистратура, Аспирантура</i>) Строка
formOfEducation	Форма обучения группы (<i>Очная, Заочная, Дистанционная</i>) Строка

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
name	ивтб-3

Ответ сервера

```
[
  {
    "id": 638,
    "nameShort": "ИВТб-31",
    "nameLong": "ИВТб-3301-01-00",
    "degree": "Бакалавр",
    "formOfEducation": "Очная"
  },
  {
    "id": 639,
    "nameShort": "ИВТб-32",
    "nameLong": "ИВТб-3302-02-00",
    "degree": "Бакалавр",
    "formOfEducation": "Очная"
  }
]
```

Метод: schedule.add

Данный метод позволяет преподавателю добавлять учебные пары в расписание.

- Добавленные пары добавляются только для преподавателя, который вызвал данный метод, иначе говоря, добавить пару для другого преподавателя **нельзя**
- Преподаватель может добавлять **любой** предмет в расписание
- Преподаватель может добавлять **любую** учебную группу в расписание

Параметры

Название	Назначение
token	Уникальный ключ пользователя Строка
groupID	Положительное число, которое характеризует уникальный идентификатор учебной группы. Положительное число
subjectID	Положительное число, которое характеризует уникальный идентификатор дисциплины. Положительное число
n	Номер учебной пары по порядку от 0 до 6. Положительное число
type	Параметр, который определяет тип пары: 0 - Лекция 1 - Практика 2 - лабораторная работа Целое число
room	Номер кабинета и учебного корпуса Строка
day	День Число
month	Месяц Число
year	Год Число

Ответ запроса

В качестве ответа возвращается объект пары в формате json

Название	Назначение
id	Уникальный идентификатор занятия Положительное число
teacherID	Уникальный идентификатор преподавателя Положительное число
firstName	Имя преподавателя Строка
secondName	Фамилия преподавателя Строка
middleName	Отчество преподавателя Строка
subjectID	Уникальный идентификатор предмета Положительное число

subjectName	Название учебного предмета Строка
nPair	Порядковый номер учебной пары начиная от первого дня первой недели. Пример. 0 - первая пара в понедельник первой недели Положительное число
type	Тип пары (Лекция, Практика, Лабораторная работа) Строка
room	Номер учебного корпуса и номер аудитории Строка
date	Дата проведения пары. Не обязательный параметр, который указывается только тогда, когда пара поставлена самим преподавателем Строка
groups	Список групп, у которых будет проходить заданная пара Список

Элемент списка **groups** имеет следующий формат:

Название	Назначение
id	Уникальный идентификатор группы Положительное число
nameShort	Короткое имя группы Строка
nameLong	Длинное имя группы Строка
degree	Ученая степень группы (Бакалавр, Специалист, Магистратура, Аспирантура) Строка
formOfEducation	Форма обучения группы (Очная, Заочная, Дистанционная) Строка

Ошибки

Данный метод возвращает стандартные ошибки.
Стандартные ошибки определены [в данном разделе](#).
Так же есть специальные ошибки:

- Данная пара уже существует

Пример

Параметры запроса

Название	Назначение
token	niy4vkrlif2mmpbpdlix3q1
groupID	587
subjectID	77
n	3
type	0
room	1-128
day	17

month	12
year	2016

Ответ сервера

```
{
  "id": 4017,
  "teacherID": 468,
  "firstName": "Геннадий",
  "secondName": "Чистяков",
  "middleName": "Андреевич",
  "subjectID": 77,
  "subjectName": "Тайм-менеджмент",
  "nPair": 80,
  "type": "Лекция",
  "room": "1-128",
  "groups": [
    {
      "id": 587,
      "nameShort": "ИВТ6-32",
      "nameLong": "ИВТ6-3302-02-00",
      "degree": "Бакалавр",
      "formOfEducation": "Очная"
    }
  ],
  "date": "2016-12-17"
}
```

Метод: schedule.addGroup

Данный метод позволяет преподавателю добавлять учебным группам пары.

- Пары могут добавляться группам, у которых в данное время стоит другая пара
- Возможно добавлять только те пары, которые ведет сам преподаватель

Параметры

Название	Назначение
token	Уникальный ключ пользователя Строка
id	Положительное число, которое характеризует уникальный идентификатор пары. Положительное число
groupID	Положительное число, которое характеризует уникальный идентификатор учебной группы. Положительное число

Ответ запроса

В качестве ответа возвращается код статуса 200, если во время выполнения метода не произошли ошибки, иначе возвращается ошибка в формате JSON

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Название	Назначение
token	niy4vkrlif2mmpbpdlixx3q1
id	4017
groupID	586

Ответ сервера

```
statusCode: 200
```

Метод: schedule.del

Данный метод позволяет преподавателю убирать учебные пары из расписания.

- Пары могут быть убраны преподавателем только из своего расписания
- Возможно убрать только пары поставленные самим преподавателем

Параметры

Название	Назначение
token	Уникальный ключ пользователя Строка
id	Положительное число, которое характеризует уникальный идентификатор пары.

Ответ запроса

В качестве ответа возвращается код статуса 200, если во время выполнения метода не произошли ошибки, иначе возвращается ошибка в формате JSON

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Так же есть специальные ошибки:

- [Данная пара не существует](#)

Пример

Параметры запроса

Название	Назначение
token	niy4vkrlif2mmpbpdlixx3q1
id	4017

Ответ сервера

statuscode: 200

Метод: schedule.delGroup

Данный метод позволяет преподавателю убирать учебным группам пары.

- Возможно убирать только те пары, которые ведет сам преподаватель

Параметры

Название	Назначение
token	Уникальный ключ пользователя Строка
id	Положительное число, которое характеризует уникальный идентификатор пары. Положительное число
groupID	Положительное число, которое характеризует уникальный идентификатор учебной группы. Положительное число

Ответ запроса

В качестве ответа возвращается код статуса 200, если во время выполнения метода не произошли ошибки, иначе возвращается ошибка в формате JSON

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Название	Назначение
token	niy4vkrlif2mmpbpdlixx3q1
id	4017
groupID	586

Ответ сервера

```
statusCode: 200
```

Метод: schedule.get

Возвращает расписание заданной группы или преподавателя.

Параметры

Название	Назначение
id	Положительное число, которое характеризует уникальный идентификатор учебной группы, либо преподавателя. Целое число
type	Параметр, который определяет чье расписание будет получено: 0 - учебной группы 1 - преподавателя

Ответ запроса

В качестве ответа возвращается json список из объектов с заданными полями:

Название	Назначение
id	Уникальный идентификатор занятия Положительное число
teacherID	Уникальный идентификатор преподавателя Положительное число
firstName	Имя преподавателя Строка
secondName	Фамилия преподавателя Строка
middleName	Отчество преподавателя Строка
subjectID	Уникальный идентификатор предмета Положительное число
subjectName	Название учебного предмета Строка
nPair	Порядковый номер учебной пары начиная от первого дня первой недели. Пример. 0 - первая пара в понедельник первой недели Положительное число
type	Тип пары (<i>Лекция, Практика, Лабораторная работа</i>) Число
room	Номер учебного корпуса и номер аудитории Строка
date	Дата проведения пары. Не обязательный параметр, который указывается только тогда, когда пара поставлена самим преподавателем Строка
groups	Список групп, у которых будет проходить заданная пара Список

Элемент списка **groups** имеет следующий формат:

Название	Назначение
id	Уникальный идентификатор группы Положительное число

nameShort	Короткое имя группы Строка
nameLong	Длинное имя группы Строка
degree	Ученая степень группы (Бакалавр, Специалитет, Магистратура, Аспирантура) Строка
formOfEducation	Форма обучения группы (Очная, Заочная, Дистанционная) Строка

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
id	156
type	1

Ответ сервера

```
[
  ...
  {
    "id": 4017,
    "teacherID": 156,
    "firstName": "Мария",
    "secondName": "Долженкова",
    "middleName": "Львовна",
    "subjectID": 664,
    "subjectName": "Технологии программирования",
    "nPair": 31,
    "type": "0",
    "room": "1-242",
    "groups": [
      {
        "id": 585,
        "nameShort": "ИВТ6-31",
        "nameLong": "ИВТ6-3301-01-00",
        "degree": "Бакалавр",
        "formOfEducation": "Очная"
      },
      {
        "id": 587,
        "nameShort": "ИВТ6-32",
        "nameLong": "ИВТ6-3302-02-00",
        "degree": "Бакалавр",
        "formOfEducation": "Очная"
      }
    ]
  },
  ...
]
```

Метод: student.changeGroup

Данный метод позволяет студенту изменить свою учебную группу.

Параметры

Название	Назначение
id	Положительное число, которое характеризует уникальный идентификатор учебной группы. Положительное число
token	Уникальный ключ пользователя Строка

Ответ запроса

В качестве ответа код статуса 200, если метод успешно выполнен и ошибка, если в процессе выполнения метода произошли исключения.

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
id	443
token	niy4vkrlif2mmpbpdlixx3q1

Ответ сервера

```
statusCode: 200
```

Метод: student.get

Данный метод обеспечивает получение данных о заданном студенте.

Возвращает информацию о студенте с заданным **id**

Параметры

Название	Назначение
id	Положительное число, которое характеризует уникальный идентификатор студента. Целое число

Ответ запроса

В качестве ответа возвращается json объект с заданными полями:

Название	Назначение
id	Уникальный идентификатор студента Положительное число
firstName	Имя пользователя Строка
secondName	Фамилия Строка
middleName	Отчество Строка
groupID	Уникальный идентификатор учебной группы Положительное число
groupShortName	Короткое имя группы Строка
levelOfAccess	Права доступа в группе. Подробнее о правах доступа можно прочитать в данном разделе Положительное число

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
id	1

Ответ сервера

```
{
  "id": 1,
  "firstName": "Даниил",
  "secondName": "Щесняк",
  "middleName": "Сергеевич",
  "groupID": 639,
  "groupShortName": "ИБТ6-32",
  "levelOfAccess": 0
}
```

Метод: student.search

Данный метод позволяет найти студента по его фамилии, имени или отчеству.
Возвращает информацию о студенте с заданным полем **name**

Параметры

Название	Назначение
name	Часть имени, фамилии или отчества студента. Строка

Ответ запроса

В качестве ответа возвращается json список состоящий из объектов с заданными полями:

Название	Назначение
id	Уникальный идентификатор студента Положительное число
firstName	Имя пользователя Строка
secondName	Фамилия Строка
middleName	Отчество Строка
groupID	Уникальный идентификатор учебной группы Положительное число
groupShortName	Короткое имя группы Строка
levelOfAccess	Права доступа в группе. Подробнее о правах доступа можно прочитать в данном разделе Положительное число

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
name	Дан

Ответ сервера

```
[
  {
    "id": 1,
    "firstName": "Даниил",
    "secondName": "Щесняк",
    "middleName": "Сергеевич",
    "groupID": 639,
    "groupShortName": "ИВТ6-32",
    "levelOfAccess": 0
  }
]
```

Метод: teacher.get

Данный метод обеспечивает получение данных о заданном преподавателе.

Возвращает информацию о преподавателе с заданным **id**

Параметры

Название	Назначение
id	Положительное число, которое характеризует уникальный идентификатор преподавателя. Положительное число

Ответ запроса

В качестве ответа возвращается json объект с заданными полями:

Название	Назначение
id	Уникальный идентификатор преподавателя Положительное число
firstName	Имя пользователя Строка
secondName	Фамилия Строка
middleName	Отчество Строка
departmentID	уникальный идентификатор кафедры Положительное число

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
id	1

Ответ сервера

```
{
  "id": 1,
  "firstName": "Александр",
  "secondName": "Абатуров",
  "middleName": "Иванович",
  "DepartmentID": 74
}
```

Метод: teacher.getSubjects

Данный метод позволяет получить список дисциплин, которые преподает заданный преподаватель

Параметры

Название	Назначение
id	Положительное число, которое характеризует уникальный идентификатор преподавателя. Положительное число

Ответ запроса

В качестве ответа возвращается json список объекты которого имеют заданные поля:

Название	Назначение
id	Уникальный идентификатор учебной дисциплины Положительное число
subjectName	Название учебной дисциплины Строка

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
id	648

###Ответ сервера

```
[
  {
    "id": 77,
    "subjectName": "Тайм-менеджмент"
  },
  {
    "id": 1239,
    "subjectName": "Программирование"
  },
  {
    "id": 2086,
    "subjectName": "Разработка программных систем"
  },
  {
    "id": 2091,
    "subjectName": "Технология разработки программного обеспечения"
  }
]
```


Метод: teacher.search

Данный метод позволяет найти преподавателя по его фамилии, имени или отчеству.
Возвращает информацию о преподавателе с заданным полем **name**

Параметры

Название	Назначение
name	Часть имени, фамилии или отчества преподавателя. Строка

Ответ запроса

В качестве ответа возвращается json список состоящий из объектов с заданными полями:

Название	Назначение
id	Уникальный идентификатор преподавателя Положительное число
firstName	Имя пользователя Строка
secondName	Фамилия Строка
middleName	Отчество Строка
departmentID	уникальный идентификатор кафедры Положительное число

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
name	Чар

Ответ сервера

```
[
  {
    "id": 631,
    "firstName": "Антон",
    "secondName": "Чарушин",
    "middleName": "Николаевич",
    "DepartmentID": 26
  },
  {
    "id": 632,
    "firstName": "Николай",
    "secondName": "Чарушин",
    "middleName": "Александрович",
    "DepartmentID": 26
  }
]
```

Метод: user.changeName

Данный метод позволяет изменить личную информацию пользователя. Таковую как имя, фамилию или отчество

Параметры

Название	Назначение
token	Уникальный ключ пользователя. Строка
firstName	Имя пользователя Строка
secondName	Фамилия пользователя Строка
middleName	Отчество пользователя Строка

Даже если пользователь не изменил часть своего ФИО, то необходимо обязательно отправлять все поля

Ответ запроса

В качестве ответа возвращается код 200 в случае успешного выполнения метода и ошибки, если они возникли в процессе выполнения метода

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
token	n1y4vkr1if2mmpbpdlix3q1
firstName	Даниил
secondName	Щесняк
middleName	Сергеевич

Ответ сервера

statusCode: 200

Метод: user.signin

Данный метод обеспечивает смену пароля пользователя

Параметры

Название	Назначение
old_password	Старый пароль. Строка
new_password	Новый пароль. Строка
token	Уникальный ключ доступа пользователя. Строка

Ответ запроса

В качестве ответа в коде статуса возвращается ответ от сервера.
Значение кодов статуса представлены [в соответствующем разделе](#)

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
old_password	pass
new_password	password
token	8qv19jlmaubcho8j50r1r3th

Ответ сервера

statusCode: 200

Метод: user.logout

Данный метод обеспечивает корректный выход пользователя из его ученой записи.

Параметры

Название	Назначение
token	Уникальный ключ пользователя. Строка

Ответ запроса

В качестве ответа в коде статуса возвращается ответ от сервера.
Значение кодов статуса представлены [тут](#)

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Параметры запроса

Параметры	Значение параметра
token	8qv19jlmaubcho8j50r1r3th

Ответ сервера

statusCode: 200

Метод: user.signin

Данный метод обеспечивает авторизацию пользователя.
В результате работы метода будет возвращен **token**, который необходим для совершения операций от имени пользователя

Параметры

Название	Назначение
login	Имя пользователя. Строка
password	Пароль пользователя. Строка

Ответ запроса

В качестве ответа возвращается json объект с заданными полями:

Название	Назначение
id	Идентификатор студента\преподавателя. Целое число
type	Тип ученой записи: 0 - Студент 1 - Преподаватель Положительное число
token	Уникальный ключ доступа пользователя. Строка

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Так же есть специальные ошибки:

- [Неверные данные для авторизации](#)

Пример

Параметры запроса

Параметры	Значение параметра
login	user
password	pass

Ответ сервера

```
{
  "id": 648,
  "type": 1,
  "token": "niy4vkrlif2mmpbpdlix3q1"
}
```

Метод: user.signup

Данный метод осуществляет регистрацию пользователя

Параметры

Название	Назначение
login	Логин пользователя. Строка
password	Пароль пользователя. Строка
type	Тип регистрируемой ученой записи: студент(0) или преподаватель(1). Число [0-1]

Параметры преподавателя

Название	Назначение
id	Уникальный идентификатор преподавателя, с которым связывается ученая запись. Положительное число
secretKey	Секретный пароль для проверки принадлежности преподавателя к кафедре. Строка

Параметры студента

Название	Назначение
GroupID	Уникальный идентификатор группы. Положительное число
firstName	Имя студента. Строка
secondName	Фамилия студента. Строка
middleName	Отчество студента. Строка

Ответ запроса

В качестве ответа в коде статуса возвращается ответ от сервера.
Значение кодов статуса представлены в [соответствующем разделе](#)

Ошибки

Данный метод возвращает [стандартные ошибки](#).

Пример

Данный пример иллюстрирует регистрацию преподавателя

Параметры запроса

Параметры	Значение параметра
login	user
password	pass

type	1
id	1
secretKey	SecretKey

Ответ сервера

statusCode: 200