МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

**(«ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

ДОКУМЕНТАЦИЯ ПРОГРАММИСТА

по дисциплине «Технологии программирования»

Выполнил студент группы ИВТ-32 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Щесняк Д. С./

Выполнил студент группы ИВТ-32 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Ямашев Р. И./

Проверил доцент кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Долженкова М. Л./

Киров 2017

# 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": "ИВТб-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.

<div class="page-break"></div>

# 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

<div class="page-break"></div>

## 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

<div class="page-break"></div>

## 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 (\*Бакалавр\*, \*Специалитет\*, \*Магистратура\*, \*Аспирантура\*)

| \*\*formOfEducation\*\* | Строка | Форма обучения группы (\*Очная\*, \*Заочная\*, \*Дистанционная\*)

<div class="page-break"></div>

## 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 идентфикатор предмета

| \*\*subjectName\*\* | Строка | Название учебного предмета

| \*\*nPair\*\* | Число | Порядковый номер учебной пары начиная от первого дня первой недели. недели

| \*\*type\*\* | Число |Тип пары (\*Лекция\*, \*Практика\*, \*Лабораторная работа\*)

| \*\*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|

<div class="page-break"></div>

## 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

<div class="page-break"></div>

# 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<br> Параметр не был передан

\*\*403\*\* | Доступ запрещен | Неверный логин или пароль<br> Доступ запрещен

\*\*409\*\* | Конфликт | Пара уже существует <br> Пользователь уже сущесвует

<div class="page-break"></div>

# Ошибки

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

<div class="page-break"></div>

# Ошибка: Доступ запрещен <a name="AccessDenidedError"></a>

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"

}

```

<div class="page-break"></div>

# Ошибка: Ошибка аутентификации <a name="AuthError"></a>

Данная ошибка случается при передаче в метод [user.signin](#user.signin) неправильного логина или пароля

## Параметры

| Название | Назначение | Значение

| :------------- | :------------- | :------------- |

| \*\*codeError\*\* | Код ошибки | 4

| \*\*errorName\*\* | Имя ошибки | Ошибка аутентификации

| \*\*errorText\*\* | Текст ошибки | Неверно заданы параметры логин или пароль

## Пример

### Ответ сервера

```

{

"errorCode": 4,

"errorName": "Ошибка аутентификации",

"errorText": "Неверно заданы параметры логин или пароль"

}

```

<div class="page-break"></div>

# Ошибка: Неправильный формат <a name="BadParameterFormatError"></a>

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"

}

```

<div class="page-break"></div>

# Ошибка: Отсутсвует параметр <a name="MissingParameterError"></a>

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

## Параметры

| Название | Назначение | Значение

| :------------- | :------------- | :------------- |

| \*\*codeError\*\* | Код ошибки | 1

| \*\*errorName\*\* | Имя ошибки | Параметр отсуствует |

| \*\*parameterName\*\* | Имя отсуствующего параметра | -

| \*\*errorText\*\* | Текст ошибки | Отсуствует обязательный параметр

## Пример

### Ответ сервера

```

{

"parameterName": "password",

"errorCode": 1,

"errorName": "Ошибка отсуствия параметра",

"errorText": "Отсуствует обязательный параметр password"

}

```

<div class="page-break"></div>

# Ошибка: Пара уже существует <a name="PairAlreadyExists"></a>

Данная ошибка возникает при использовании метода [schedule.add](#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"

}

```

<div class="page-break"></div>

# Ошибка: Пара отсуствует <a name="PairAlreadyExists"></a>

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

## Параметры

| Название | Назначение | Значение

| :------------- | :------------- | :------------- |

| \*\*codeError\*\* | Код ошибки | 7

| \*\*errorName\*\* | Имя ошибки | Пара не найдена

| \*\*errorText\*\* | Текст ошибки | Ошибка. Заданная пара не сущесвует

## Пример

### Ответ сервера

```

{

"errorCode": 7,

"errorName": "Пара не найдена",

"errorText": "Ошибка. Заданная пара не сущесвует"

}

```

<div class="page-break"></div>

# Ошибка: Пользователь уже существует <a name="UserAlreadyExistsError"></a>

Данная ошибка возникает при использовании метода [user.signup](#user.signup), если данный пользователь уже существует в системе

## Параметры

| Название | Назначение | Значение

| :------------- | :------------- | :------------- |

| \*\*codeError\*\* | Код ошибки | 3

| \*\*errorName\*\* | Имя ошибки | Ошибка существующего пользователя

| \*\*errorText\*\* | Текст ошибки | Ошибка. Данный пользователь уже сущесвует

## Пример

### Ответ сервера

```

{

"errorCode": 3,

"errorName": "Ошибка существующего пользователя",

"errorText": "Ошибка. Данный пользователь уже сущесвует"

}

```

<div class="page-break"></div>

# Методы

В данном разделе описываются методы, их параметры и возвращаемые значения

<div class="page-break"></div>

# Метод: announcement.add<a name="announcement.add"></a>

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 <br>\*\*Строка\*\*

| \*\*id\*\* | Уникальный идентификатор учебной пары<br>\*\*Положительное число\*\*

| \*\*text\*\* | Текст сообщения <br> \*\*Строка\*\*

| \*\*day\*\* | День проведения пары <br> \*\*Число\*\*

| \*\*month\*\* | Месяц проведения пары <br> \*\*Число\*\*

| \*\*year\*\* | Год проведения пары <br> \*\*Число\*\*

## Ответ запроса

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

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## 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

```

<div class="page-break"></div>

# Метод: announcement.del<a name="announcement.del"></a>

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 <br>\*\*Строка\*\*

| \*\*id\*\* | Уникальный идентификатор сообщения<br>\*\*Положительное число\*\*

## Ответ запроса

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

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## Пример

### Параметры запроса

| Параметры | Значение параметра |

| :------------- | :------------- |

| \*\*token\*\* | niy4vkrlif2mmpbpdlixx3q1

| \*\*id\*\* | 1

### Ответ сервера

```

statuscode: 200

```

<div class="page-break"></div>

# Метод: announcement.get<a name="announcement.get"></a>

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 <br>\*\*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<br>\*\*Положительное число\*\*

| \*\*scheduleID\*\* | Уникальный идентификатор пары<br>\*\*Положительное число\*\*

| \*\*firstName\*\* | Имя<br>\*\*Строка\*\*

| \*\*secondName\*\* | Фамилия<br>\*\*Строка\*\*

| \*\*middleName\*\* | Отчество<br>\*\*Строка\*\*

| \*\*text\*\* | Сообщение<br>\*\*Строка\*\*

| \*\*announceTime\*\* | Время написания сообщения<br>\*\*Строка\*\*

| \*\*pairDate\*\* | День проведения учебной пары<br>\*\*Строка\*\*

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## 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"

}

]

```

<div class="page-break"></div>

# Метод: deparment.changeSecretKey<a name="deparment.changeSecretKey"></a>

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 <br>\*\*Строка\*\*

| \*\*secretKey\*\* | Новый секретный ключ для регистрации преподаватлей на данной кафедре. Ключ не более 50 символов <br> \*\*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](#errors).<br>

## Пример

### Параметры запроса

| Параметры | Значение параметра |

| :------------- | :------------- |

| \*\*token\*\* | niy4vkrlif2mmpbpdlixx3q1 |

| \*\*secretKey\*\* | VerySecretKey!

### Ответ сервера

```

statuscode: 200

```

<div class="page-break"></div>

# Метод: group.changeAccess<a name="group.changeAccess"></a>

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 прочитать в [разделе о структуре студентов](#levelOfAccess)

## Параметры

| Название | Назначение |

| :------------- | :------------- |

| \*\*studentID\*\* | Положительное число, которое характеризует уникальный идентификатор студента. <br>\*\*Положительное число\*\*

| \*\*access\*\* | Число от 0 до 3, которое обозначает уровень прав студента. <br> \*\*Положительное число\*\*

| \*\*token\*\* | Уникальный ключ пользователя <br>\*\*Строка\*\*

## Ответ запроса

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

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## Пример

### Параметры запроса

| Параметры | Значение параметра |

| :------------- | :------------- |

| \*\*id\*\* | 1 |

| \*\*access\*\* | 3

| \*\*token\*\* | niy4vkrlif2mmpbpdlixx3q1

### Ответ сервера

```

statuscode: 200

```

<div class="page-break"></div>

# Метод: group.get<a name="group.get"></a>

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. <br>\*\*Целое число\*\*

## Ответ запроса

В качестве ответа возвращается json объект с заданными полями:

| Название | Назначение |

| :------------- | :------------- |

|\*\*id\*\* | Уникальный идентификатор группы<br>\*\*Положительное число\*\*

|\*\*nameShort\*\* | Короткое имя группы<br>\*\*Строка\*\*

| \*\*nameLong\*\* | Длинное имя группы<br>\*\*Строка\*\*

| \*\*degree\*\* | Ученая степень группы (\*Бакалавр\*, \*Специалитет\*, \*Магистратура\*, \*Аспирантура\*)<br>\*\*Строка\*\*

| \*\*formOfEducation\*\* | Форма обучения группы (\*Очная\*, \*Заочная\*, \*Дистанционная\*)<br>\*\*Строка\*\*

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## Пример

### Параметры запроса

| Параметры | Значение параметра |

| :------------- | :------------- |

| \*\*id\*\* | 639 |

### Ответ сервера

```

{

"id": 639,

"nameShort": "ИВТб-32",

"nameLong": "ИВТб-3302-02-00",

"degree": "Бакалавр",

"formOfEducation": "Очная"

}

```

<div class="page-break"></div>

# Метод: group.getMembers<a name="group.getMembers"></a>

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. <br>\*\*Положительное число\*\*

## Ответ запроса

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

| Название | Назначение |

| :------------- | :------------- |

| \*\*id\*\* | Уникальный идентификатор студента<br>\*\*Положительное число\*\*

| \*\*firstName\*\* | Имя пользователя<br>\*\*Строка\*\*

| \*\*secondName\*\* | Фамилия<br>\*\*Строка\*\*

| \*\*middleName\*\* | Отчество<br>\*\*Строка\*\*

| \*\*groupID\*\* | Уникальный идентификатор учебной группы<br>\*\*Положительное число\*\*

| \*\*groupShortName\*\* | Короткое имя группы<br>\*\*Строка\*\*

| \*\*levelOfAccess\*\* | Права доступа в группе. Подробнее о правах доступа можно прочитать в [данном разделе](#levelOfAccess)<br>\*\*Положительное число\*\*

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## Пример

### Параметры запроса

| Параметры | Значение параметра |

| :------------- | :------------- |

| \*\*id\*\* | 639 |

### Ответ сервера

```

[

{

"id": 1,

"firstName": "Даниил",

"secondName": "Щесняк",

"middleName": "Сергеевич",

"groupID": 639,

"groupShortName": "ИВТб-32",

"levelOfAccess": 0

},

{

"id": 2,

"firstName": "Никита",

"secondName": "Братухин",

"middleName": "",

"groupID": 639,

"groupShortName": "ИВТб-32",

"levelOfAccess": 0

},

{

"id": 3,

"firstName": "Алексей",

"secondName": "Губин",

"middleName": "",

"groupID": 639,

"groupShortName": "ИВТб-32",

"levelOfAccess": 0

}

]

```

<div class="page-break"></div>

# Метод: group.search<a name="group.search"></a>

Данный метод возвращает список групп, которые соотвествуют поисковому запросу.

Возвращает информацию о группах, которые имеют короткое имя как в параметре \*\*name\*\*

## Параметры

| Название | Назначение |

| :------------- | :------------- |

| \*\*name\*\* | Поисковый запрос. <br>\*\*Строка\*\*

## Ответ запроса

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

| Название | Назначение |

| :------------- | :------------- |

| \*\*id\*\* | Уникальный идентификатор группы<br>\*\*Положительное число\*\*

| \*\*nameShort\*\* | Короткое имя группы<br>\*\*Строка\*\*

| \*\*nameLong\*\* | Длинное имя группы<br>\*\*Строка\*\*

| \*\*degree\*\* | Ученая степень группы (\*Бакалавр\*, \*Специалитет\*, \*Магистратура\*, \*Аспирантура\*)<br>\*\*Строка\*\*

| \*\*formOfEducation\*\* | Форма обучения группы (\*Очная\*, \*Заочная\*, \*Дистанционная\*)<br>\*\*Строка\*\*

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## 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"

}

]

```

<div class="page-break"></div>

# Метод: schedule.add<a name="schedule.add"></a>

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 <br> \*\*Строка\*\*

| \*\*groupID\*\* | Положительное число, которое характеризует уникальный идентификатор учебной группы. <br>\*\*Положительное число\*\*

| \*\*subjectID\*\* | Положительное число, которое характеризует уникальный идентификатор дисциплины. <br>\*\*Положительное число\*\*

| \*\*n\*\* | Номер учебной пары по порядку от 0 до 6. <br>\*\*Положительное число\*\*

| \*\*type\*\* | Параметр, который определяет тип пары: <br> 0 - Лекция <br> 1 - Практика <br> 2 - лабораторная работа <br> \*\*Целое число\*\*

| \*\*room\*\* | Номер кабинета и учебного корпуса <br> \*\*Строка\*\*

| \*\*day\*\* | День <br> \*\*Число\*\*

| \*\*month\*\* | Месяц <br> \*\*Число\*\*

| \*\*year\*\* | Год <br> \*\*Число\*\*

## Ответ запроса

В качестве ответа возвращается обект пары в формате json

| Название | Назначение |

| :------------- | :------------- |

| \*\*id\*\* | Уникальный идентификатор занятия<br>\*\*Положительное число\*\*

| \*\*teacherID\*\* | Уникальный идентификатор преподавателя<br>\*\*Положительное число\*\*

| \*\*firstName\*\* | Имя преподавателя<br>\*\*Строка\*\*

| \*\*secondName\*\* | Фамилия преподавателя<br>\*\*Строка\*\*

| \*\*middleName\*\* | Отчество преподавателя<br>\*\*Строка\*\*

| \*\*subjectID\*\* | Уникальный идентфикатор предмета<br>\*\*Положительное число\*\*

| \*\*subjectName\*\* | Название учебного предмета<br>\*\*Строка\*\*

| \*\*nPair\*\* | Порядковый номер учебной пары начиная от первого дня первой недели.<br> Пример. 0 - первая пара в понедельник первой недели<br>\*\*Положительное число\*\*

| \*\*type\*\* | Тип пары (\*Лекция\*, \*Практика\*, \*Лабораторная работа\*)<br>\*\*Строка\*\*

| \*\*room\*\* | Номер учебного корпуса и номер аудитории<br>\*\*Строка\*\*

| \*\*date\*\* | Дата проведения пары. Не обязательный параметр, который указывается только тогда, когда пара поставлена самим преподавателем<br>\*\*Строка\*\*

| \*\*groups\*\* | Список групп, у которых будет проходить заданная пара<br>\*\*Список\*\*

Элемент списка \*\*groups\*\* имеет следующий формат:

| Название | Назначение |

| :------------- | :------------- |

| \*\*id\*\* | Уникальный идентификатор группы<br>\*\*Положительное число\*\*

| \*\*nameShort\*\* | Короткое имя группы<br>\*\*Строка\*\*

| \*\*nameLong\*\* | Длинное имя группы<br>\*\*Строка\*\*

| \*\*degree\*\* | Ученая степень группы (\*Бакалавр\*, \*Специалитет\*, \*Магистратура\*, \*Аспирантура\*)<br>\*\*Строка\*\*

| \*\*formOfEducation\*\* | Форма обучения группы (\*Очная\*, \*Заочная\*, \*Дистанционная\*)<br>\*\*Строка\*\*

## Ошибки

Данный метод возвращает стандартные ошибки.

Стандартные ошибки определены [в данном разделе](#errors).

Так же есть специальные ошибки:

- [Данная пара уже существует](#PairAlreadyExists)

## Пример

### Параметры запроса

| Название | Назначение |

| :------------- | :------------- |

| \*\*token\*\* | niy4vkrlif2mmpbpdlixx3q1

| \*\*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": "ИВТб-32",

"nameLong": "ИВТб-3302-02-00",

"degree": "Бакалавр",

"formOfEducation": "Очная"

}

],

"date": "2016-12-17"

}

```

<div class="page-break"></div>

# Метод: schedule.addGroup<a name="schedule.addGroup"></a>

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 <br> \*\*Строка\*\*

| \*\*id\*\* | Положительное число, которое характеризует уникальный идентификатор пары. <br> \*\*Положительное число\*\*

| \*\*groupID\*\* | Положительное число, которое характеризует уникальный идентификатор учебной группы. <br> \*\*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](#errors).<br>

## Пример

### Параметры запроса

| Название | Назначение |

| :------------- | :------------- |

| \*\*token\*\* | niy4vkrlif2mmpbpdlixx3q1

| \*\*id\*\* | 4017

| \*\*groupID\*\* | 586

### Ответ сервера

```

statuscode: 200

```

<div class="page-break"></div>

# Метод: schedule.del<a name="schedule.del"></a>

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 <br> \*\*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](#errors).<br>

Так же есть специальные ошибки:

- [Данная пара не существует](#PairDoesntExists)

## Пример

### Параметры запроса

| Название | Назначение |

| :------------- | :------------- |

| \*\*token\*\* | niy4vkrlif2mmpbpdlixx3q1

| \*\*id\*\* | 4017

### Ответ сервера

```

statuscode: 200

```

<div class="page-break"></div>

# Метод: schedule.delGroup<a name="schedule.delGroup"></a>

Данный метод позволяет преподавателю убирать учебным группам пары.

- Возможно убирать только те пары, которые ведет сам преподаватель

## Параметры

| Название | Назначение |

| :------------- | :------------- |

| \*\*token\*\* | Уникальный ключ пользователя <br> \*\*Строка\*\*

| \*\*id\*\* | Положительное число, которое характеризует уникальный идентификатор пары. <br> \*\*Положительное число\*\*

| \*\*groupID\*\* | Положительное число, которое характеризует уникальный идентификатор учебной группы. <br> \*\*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](#errors).<br>

## Пример

### Параметры запроса

| Название | Назначение |

| :------------- | :------------- |

| \*\*token\*\* | niy4vkrlif2mmpbpdlixx3q1

| \*\*id\*\* | 4017

| \*\*groupID\*\* | 586

### Ответ сервера

```

statuscode: 200

```

<div class="page-break"></div>

# Метод: schedule.get<a name="schedule.get"></a>

Возвращает расписание заданной группы или преподавателя.

## Параметры

| Название | Назначение |

| :------------- | :------------- |

| \*\*id\*\* | Положительное число, которое характеризует уникальный идентификатор учебной группы, либо преподавателя. <br>\*\*Целое число\*\*

| \*\*type\*\* | Параметр, который определяет чье расписание будет получено: <br> 0 - учебной группы <br> 1 - преподавателя

## Ответ запроса

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

| Название | Назначение |

| :------------- | :------------- |

| \*\*id\*\* | Уникальный идентификатор занятия<br>\*\*Положительное число\*\*

| \*\*teacherID\*\* | Уникальный идентификатор преподавателя<br>\*\*Положительное число\*\*

| \*\*firstName\*\* | Имя преподавателя<br>\*\*Строка\*\*

| \*\*secondName\*\* | Фамилия преподавателя<br>\*\*Строка\*\*

| \*\*middleName\*\* | Отчество преподавателя<br>\*\*Строка\*\*

| \*\*subjectID\*\* | Уникальный идентфикатор предмета<br>\*\*Положительное число\*\*

| \*\*subjectName\*\* | Название учебного предмета<br>\*\*Строка\*\*

| \*\*nPair\*\* | Порядковый номер учебной пары начиная от первого дня первой недели.<br> Пример. 0 - первая пара в понедельник первой недели<br>\*\*Положительное число\*\*

| \*\*type\*\* | Тип пары (\*Лекция\*, \*Практика\*, \*Лабораторная работа\*)<br>\*\*Число\*\*

| \*\*room\*\* | Номер учебного корпуса и номер аудитории<br>\*\*Строка\*\*

| \*\*date\*\* | Дата проведения пары. Не обязательный параметр, который указывается только тогда, когда пара поставлена самим преподавателем<br>\*\*Строка\*\*

| \*\*groups\*\* | Список групп, у которых будет проходить заданная пара<br>\*\*Список\*\*

Элемент списка \*\*groups\*\* имеет следующий формат:

| Название | Назначение |

| :------------- | :------------- |

| \*\*id\*\* | Уникальный идентификатор группы<br>\*\*Положительное число\*\*

| \*\*nameShort\*\* | Короткое имя группы<br>\*\*Строка\*\*

| \*\*nameLong\*\* | Длинное имя группы<br>\*\*Строка\*\*

| \*\*degree\*\* | Ученая степень группы (\*Бакалавр\*, \*Специалитет\*, \*Магистратура\*, \*Аспирантура\*)<br>\*\*Строка\*\*

| \*\*formOfEducation\*\* | Форма обучения группы (\*Очная\*, \*Заочная\*, \*Дистанционная\*)<br>\*\*Строка\*\*

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## 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": "ИВТб-31",

"nameLong": "ИВТб-3301-01-00",

"degree": "Бакалавр",

"formOfEducation": "Очная"

},

{

"id": 587,

"nameShort": "ИВТб-32",

"nameLong": "ИВТб-3302-02-00",

"degree": "Бакалавр",

"formOfEducation": "Очная"

}

]

},

...

]

```

<div class="page-break"></div>

# Метод: student.changeGroup<a name="student.changeGroup"></a>

Данный метод позволяет студенту изменить свою учебную группу.

## Параметры

| Название | Назначение |

| :------------- | :------------- |

| \*\*id\*\* | Положительное число, которое характеризует уникальный идентификатор учебной группы. <br>\*\*Положительное число\*\*

| \*\*token\*\* | Уникальный ключ пользователя<br>\*\*Строка\*\*

## Ответ запроса

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

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## Пример

### Параметры запроса

| Параметры | Значение параметра |

| :------------- | :------------- |

| \*\*id\*\* | 443 |

| \*\*token\*\* | niy4vkrlif2mmpbpdlixx3q1

### Ответ сервера

```

statuscode: 200

```

<div class="page-break"></div>

# Метод: student.get<a name="student.get"></a>

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. <br>\*\*Целое число\*\*

## Ответ запроса

В качестве ответа возвращается json объект с заданными полями:

| Название | Назначение |

| :------------- | :------------- |

| \*\*id\*\* | Уникальный идентификатор студента<br>\*\*Положительное число\*\*

| \*\*firstName\*\* | Имя пользователя<br>\*\*Строка\*\*

| \*\*secondName\*\* | Фамилия<br>\*\*Строка\*\*

| \*\*middleName\*\* | Отчество<br>\*\*Строка\*\*

| \*\*groupID\*\* | Уникальный идентификатор учебной группы<br>\*\*Положительное число\*\*

| \*\*groupShortName\*\* | Короткое имя группы<br>\*\*Строка\*\*

| \*\*levelOfAccess\*\* | Права доступа в группе. Подробнее о правах доступа можно прочитать в [данном разделе](#levelOfAccess)<br>\*\*Положительное число\*\*

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## Пример

### Параметры запроса

| Параметры | Значение параметра |

| :------------- | :------------- |

| \*\*id\*\* | 1 |

### Ответ сервера

```

{

"id": 1,

"firstName": "Даниил",

"secondName": "Щесняк",

"middleName": "Сергеевич",

"groupID": 639,

"groupShortName": "ИВТб-32",

"levelOfAccess": 0

}

```

<div class="page-break"></div>

# Метод: student.search<a name="student.search"></a>

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. <br>\*\*Строка\*\*

## Ответ запроса

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

| Название | Назначение |

| :------------- | :------------- |

| \*\*id\*\* | Уникальный идентификатор студента<br>\*\*Положительное число\*\*

| \*\*firstName\*\* | Имя пользователя<br>\*\*Строка\*\*

| \*\*secondName\*\* | Фамилия<br>\*\*Строка\*\*

| \*\*middleName\*\* | Отчество<br>\*\*Строка\*\*

| \*\*groupID\*\* | Уникальный идентификатор учебной группы<br>\*\*Положительное число\*\*

| \*\*groupShortName\*\* | Короткое имя группы<br>\*\*Строка\*\*

| \*\*levelOfAccess\*\* | Права доступа в группе. Подробнее о правах доступа можно прочитать в [данном разделе](#levelOfAccess)<br>\*\*Положительное число\*\*

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## Пример

### Параметры запроса

| Параметры | Значение параметра |

| :------------- | :------------- |

| \*\*name\*\* | Дан |

### Ответ сервера

```

[

{

"id": 1,

"firstName": "Даниил",

"secondName": "Щесняк",

"middleName": "Сергеевич",

"groupID": 639,

"groupShortName": "ИВТб-32",

"levelOfAccess": 0

}

]

```

<div class="page-break"></div>

# Метод: teacher.get<a name="teacher.get"></a>

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. <br>\*\*Положительное число\*\*

## Ответ запроса

В качестве ответа возвращается json объект с заданными полями:

| Название | Назначение |

| :------------- | :------------- |

| \*\*id\*\* | Уникальный идентификатор преподавателя<br> \*\*Положительное число\*\*

| \*\*firstName\*\* | Имя пользователя<br>\*\*Строка\*\*

| \*\*secondName\*\* | Фамилия<br>\*\*Строка\*\*

| \*\*middleName\*\* | Отчество<br>\*\*Строка\*\*

| \*\*departmentID\*\* | уникальный идентификатор кафедры<br> \*\*Положительное число\*\*

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## Пример

### Параметры запроса

| Параметры | Значение параметра |

| :------------- | :------------- |

| \*\*id\*\* | 1 |

### Ответ сервера

```

{

"id": 1,

"firstName": "Александр",

"secondName": "Абатуров",

"middleName": "Иванович",

"DepartamentID": 74

}

```

<div class="page-break"></div>

# Метод: teacher.getSubjects<a name="teacher.getSubjects"></a>

Данный метод позволяет получить список дисциплин, которые преподает заданный преподаватель

## Параметры

| Название | Назначение |

| :------------- | :------------- |

| \*\*id\*\* | Положительное число, которое характеризует уникальный идентификатор преподавателя. <br>\*\*Положительное число\*\*

## Ответ запроса

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

| Название | Назначение |

| :------------- | :------------- |

| \*\*id\*\* | Уникальный идентификатор учебной дисциплины<br>\*\*Положительное число\*\*

| \*\*subjectName\*\* | Название учебной дисциплины<br>\*\*Строка\*\*

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## 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"

}

]

```

<div class="page-break"></div>

# Метод: teacher.search<a name="teacher.search"></a>

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. <br>\*\*Строка\*\*

## Ответ запроса

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

| Название | Назначение |

| :------------- | :------------- |

| \*\*id\*\* | Уникальный идентификатор преподавателя<br> \*\*Положительное число\*\*

| \*\*firstName\*\* | Имя пользователя<br>\*\*Строка\*\*

| \*\*secondName\*\* | Фамилия<br>\*\*Строка\*\*

| \*\*middleName\*\* | Отчество<br>\*\*Строка\*\*

| \*\*departmentID\*\* | уникальный идентификатор кафедры<br> \*\*Положительное число\*\*

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## Пример

### Параметры запроса

| Параметры | Значение параметра |

| :------------- | :------------- |

| \*\*name\*\* | Чар |

### Ответ сервера

```

[

{

"id": 631,

"firstName": "Антон",

"secondName": "Чарушин",

"middleName": "Николаевич",

"DepartamentID": 26

},

{

"id": 632,

"firstName": "Николай",

"secondName": "Чарушин",

"middleName": "Александрович",

"DepartamentID": 26

}

]

```

<div class="page-break"></div>

# Метод: user.changeName <a name="user.changeName"></a>

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

## Параметры

| Название | Назначение |

| :------------- | :------------- |

| \*\*token\*\* | Уникальный ключ пользователя. <br>\*\*Строка\*\*

| \*\*firstName\*\* | Имя пользователя<br>\*\*Строка\*\*

| \*\*secondName\*\* | Фамилия пользователя<br>\*\*Строка\*\*

| \*\*middleName\*\* | Отчество пользователя<br>\*\*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](#errors).<br>

## Пример

### Параметры запроса

| Параметры | Значение параметра |

| :------------- | :------------- |

| \*\*token\*\* | niy4vkrlif2mmpbpdlixx3q1

| \*\*firstName\*\* | Даниил

| \*\*secondName\*\* | Щесняк

| \*\*middleName\*\* | Сергеевич

### Ответ сервера

```

statuscode: 200

```

<div class="page-break"></div>

# Метод: user.signin <a name="user.changePassword"></a>

Данный метод обеспечивает cмену пароля пользователя

## Параметры

| Название | Назначение |

| :------------- | :------------- |

| \*\*old\_password\*\* | Старый пароль. \*\*Строка\*\*

| \*\*new\_password\*\*| Новый пароль.<br>\*\*Строка\*\*

| \*\*token\*\* | Уникальный ключ доступа пользователя.<br>\*\*Строка\*\*

## Ответ запроса

В качестве ответа в коде статуса возвращается ответ от сервера.

Значение кодов статуса представлены [в соотвествующем разделе](#statusCode)

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

# Пример

## Параметры запроса

| Параметры | Значение параметра |

| :------------- | :------------- |

| \*\*old\_password\*\* | pass |

| \*\*new\_password\*\* | password

| \*\*token\*\* | 8qv19jlmaubcho8j50r1r3th

## Ответ сервера

```

statusCode: 200

```

<div class="page-break"></div>

# Метод: user.logout <a name="user.logout"></a>

Данный метод обеспечивает корректный выход пользователя из его ученой записи.

## Параметры

| Название | Назначение |

| :------------- | :------------- |

| \*\*token\*\* | Уникальный ключ пользователя. <br>\*\*Строка\*\*

## Ответ запроса

В качестве ответа в коде статуса возвращается ответ от сервера.

Значение кодов статуса представлены [тут](#statusCode)

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## Пример

### Параметры запроса

| Параметры | Значение параметра |

| :------------- | :------------- |

| \*\*token\*\* | 8qv19jlmaubcho8j50r1r3th |

### Ответ сервера

```

statusCode: 200

```

<div class="page-break"></div>

# Метод: user.signin <a name="user.signin"></a>

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. <br>\*\*Строка\*\*

| \*\*password\*\*| Пароль пользователя. <br>\*\*Строка\*\* |

## Ответ запроса

В качестве ответа возвращается json объект с заданными полями:

| Название | Назначение |

| :------------- | :------------- |

| \*\*id\*\* | Идентификатор студента\преподавателя. <br> \*\*Целое число\*\*

| \*\*type\*\* | Тип ученой записи: <br> 0 - Студент<br> 1 - Преподаватель<br> \*\*Положительное число\*\*

| \*\*token\*\*| Уникальный ключ доступа пользователя.<br> \*\*Строка\*\*|

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

Так же есть специальные ошибки:

- [Неверные данные для авторизации](#AuthError)

## Пример

### Параметры запроса

| Параметры | Значение параметра |

| :------------- | :------------- |

| \*\*login\*\* | user |

| \*\*password\*\* | pass

### Ответ сервера

```

{

"id": 648,

"type": 1,

"token": "niy4vkrlif2mmpbpdlixx3q1"

}

```

<div class="page-break"></div>

# Метод: user.signup<a name="user.signup"></a>

Данный метод осуществляет регистрацию пользователя

## Параметры

| Название | Назначение |

| :------------- | :------------- |

| \*\*login\*\* | Логин пользователя. <br>\*\*Строка\*\*

| \*\*password\*\* | Пароль пользователя. <br>\*\*Строка\*\*

| \*\*type\*\* | Тип регистрируемой ученой записи: \*студент(0)\* или \*преподаватель(1)\*. <br>\*\*Число [0-1]\*\*

### Параметры преподавателя

| Название | Назначение |

| :--------| :----------|

| \*\*id\*\* | Уникальный идентификатор преподавателя, с которым связывается ученая запись. <br>\*\*Положительное число\*\*

| \*\*secretKey\*\* | Секретный пароль для проверки принадлежности преподавателя к кафедре. <br>\*\*Строка\*\*

### Параметры студента

| Название | Назначение |

| :------------- | :------------- |

| \*\*GroupID\*\* | Уникальный идентификатор группы. <br>\*\*Положительное число\*\*

| \*\*firstName\*\* | Имя студента. <br>\*\*Строка\*\*

| \*\*secondName\*\* | Фамилия студента. <br>\*\*Строка\*\*

| \*\*middleName\*\* | Отчество студента. <br>\*\*Строка\*\*

## Ответ запроса

В качестве ответа в коде статуса возвращается ответ от сервера.

Значение кодов статуса представлены [в соотвествующем разделе](#statusCode)

## Ошибки

Данный метод возвращает [стандартные ошибки](#errors).<br>

## 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

```

<div class="page-break"></div>