

ПРОГРАММНЫЙ МОДУЛЬ ДЛЯ ИНТЕГРАЦИИ СТОРОННИХ ПРИЛОЖЕНИЙ С
СИСТЕМОЙ ОРИОКС

РУКОВОДСТВО ПРОГРАММИСТА

АННОТАЦИЯ

В данном документе приведено руководство программиста по использованию ПМ ИСП. Руководство состоит из пяти разделов.

В разделе “Назначение и условия применения программы” указаны назначение и функции, выполняемые программой, условия, необходимые для выполнения программы (объем оперативной памяти, требования к составу и параметрам периферийных устройств, требования к программному обеспечению и т.п.).

В разделе “Характеристики программы” приведено описание основных характеристик и особенностей программы (временные характеристики, режим работы, средства контроля правильности выполнения и самовосстанавливаемости программы и т.п.).

В разделе “Обращение к программе” приведено описание процедур вызова программы (способы передачи управления и параметров данных и др.).

В разделе “Входные и выходные данные” приведено описание организации используемой входной и выходной информации и, при необходимости, ее кодирования.

В разделе “Сообщения” указаны тексты сообщений, выдаваемых программисту или оператору в ходе выполнения программы, описание их содержания и действия, которые необходимо предпринять по этим сообщениям.

Оформление программного документа “Руководство программиста” произведено по требованиям ЕСПД (ГОСТ 19.101-77, ГОСТ 19.103-77, ГОСТ 19.104-78, ГОСТ 19.105-78, ГОСТ 19.106-78, ГОСТ 19.504-79).

СОДЕРЖАНИЕ

Аннотация	2
1 Назначение и условия применения программы	4
1.1 Требования к информационной и программной совместимости	4
2 Характеристики программы	5
3 Обращение к программе	6
3.1 Получение токена	6
3.2 Получение раздела “Контакты”	6
3.3 Получение информации о текущей неделе	7
3.4 Получение списка групп	8
3.5 Получение расписания конкретной группы	9
3.6 Получение информации о студенте	10
3.7 Получение информации о дисциплинах	11
4 Входные и выходные данные	13
5 Сообщения	14

1 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ

Назначение программы: качественное улучшение разработки приложений для системы ОРИОКС, предоставление удобного и быстрого способа получения данных студентов для разработчиков.

Состав выполняемых функций:

- создание токенов аутентификации;
- аннулирование токенов аутентификации;
- предоставление учебного расписания групп;
- предоставление информации из раздела «Контакты»;
- предоставление информации из раздела «FAQ»;
- предоставление информации из раздела «Новости»;
- работа с разделом «Обучение», а именно:
 - информация об академический задолженностях;
 - информация о дисциплинах;
 - ресурсы дисциплин;
 - уведомления;
 - информация о сданных справках по болезни;
 - просмотр и загрузка личных файлов студента;
 - просмотр и создания заявок на справку;
 - просмотр и создание заявок на помощь.

1.1 Требования к информационной и программной совместимости

Для работы ПМ ИСП требуется развёрнутое веб-приложение ОРИОКС с версией не ниже, чем 3.1. ОРИОКС уже предоставляет все необходимые программные и информационные зависимости.

2 ХАРАКТЕРИСТИКИ ПРОГРАММЫ

Программный модуль рассчитан на круглосуточную работу в течение 365 дней в году. Средства контроля и мониторинга встроены в фреймворк Yii2. В случае ошибки они автоматически выполняют следующие действия:

- 1) Программный модуль автоматически перезапускается в случае не критической ошибки.
- 2) Посылается сообщение администратору об ошибке в письме на электронную почту.
- 3) Ошибка записывается в журнал ошибок.

При критической ошибке для восстановления работы программного модуля обратитесь к системному администратору.

3 ОБРАЩЕНИЕ К ПРОГРАММЕ

Получение данных происходит при помощи HTTP запросов к точкам входа в API.

3.1 Получение токена

Чтобы получить доступ к данным студента необходимо сначала получить токен. Для получения токена используется логин и пароль от ОРИОКС, один студент может получить до восьми токенов.

Запрос выглядит следующим образом:

```
1 GET /api/v1 HTTP/1.1
2 Accept: application/json
3 Authorization: Basic <encoded_auth>
4 User-Agent: <app>/<app_version> <os>
```

В запрос входит:

<encoded_auth> — разделённые двоеточием логин:пароль, закодированные в Base64.

<app> — имя приложения, использующего токен.

<app_version> — версия приложения, использующего токен.

<os> — операционная система, на которой запущено приложение (желательно с указанием версии).

Ответ выглядит следующим образом:

```
6 HTTP/1.1 200 OK
7 {
8   "token": "String"
9 }
```

Ответ содержит:

token — токен представляет из себя случайную строку длиной в 32 символа (состоит из цифр и латинских букв).

3.2 Получение раздела “Контакты”

Запрос: GET api/v1/contacts

Формат ответа:

```
3 HTTP/1.1 200 OK
4 [
5   {
6     "classroom": "String",
7     "department": "String",
8     "email": "String",
9     "inner_tel": "String",
10    "tel": "String"
11  }
12 ]
```

Описание полей:

classroom — номер аудитории кафедры.

department — кафедра.

email — электронная почта кафедры.

inner_tel — внутренний телефон кафедры.

tel — телефон кафедры.

Пример ответа:

```
14 HTTP/1.1 200 OK
15 [
16   {
17     "classroom": "3241",
18     "department": "ВМ-1",
19     "email": "hm1@mieee.ru",
20     "inner_tel": "29-38",
21     "tel": "(499) 720-87-38"
22   }
23 ]
```

3.3 Получение информации о текущей неделе

Запрос: GET api/v1/schedule

Формат ответа:

```
3 HTTP/1.1 200 OK
4 {
5   "type": "Integer",
6   "week": "Integer"
7 }
```

Описание полей:

type — тип недели. Используется для получение расписания группы:

- 1: 1 числитель
- 2: 1 знаменатель
- 3: 2 числитель
- 4: 2 знаменатель

week — номер недели.

Пример ответа:

```

9 HTTP/1.1 200 OK
10 {
11   "type": 3,
12   "week": 7,
13 }
```

3.4 Получение списка групп

Запрос: GET api/v1/schedule/groups

Формат ответа:

```

3 HTTP/1.1 200 OK
4 [
5   {
6     "id": "Integer",
7     "name": "String"
8   }
9 ]
```

Описание полей:

id — идентификатор группы.

name — имя учебной группы.

Пример ответа:


```
11 HTTP/1.1 200 OK
12 [
13   {
14     "name": "МП-11",
15     "id": 73
16   },
17   {
18     "name": "МП-12",
19     "id": 90
20   }
21 ]
```

3.5 Получение расписания конкретной группы

Запрос: GET api/v1/schedule/groups/<id>/types/<type>

Значение параметров:

<id> — идентификатор группы.

<type> — тип недели.

Формат ответа:

```
3 HTTP/1.1 200 OK
4 [
5   {
6     "day": "Integer",
7     "classes": [
8       {
9         "classroom": "String",
10        "name": "String",
11        "number": "Integer",
12        "teacher": "String"
13      }
14    ]
15  }
16 ]
17
```

Описание полей:

day — номер дня недели.

classroom — номер аудитории.

name — название предмета.

number — номер пары.

teacher — ФИО преподавателя.

Пример ответа:

```
19 [
20   {
21     "day": 1,
22     "classes": [
23       {
24         "classroom": "4212a",
25         "name": "Нейронные сети",
26         "number": 3,
27         "teacher": "Рычагов Михаил Николаевич"
28       },
29       {
30         "classroom": "1204",
31         "name": "Управление программными проектами",
32         "number": 4,
33         "teacher": "Слюсарь Валентин Викторович"
34       }
35     ]
36   }
37 ]
```

3.6 Получение информации о студенте

Запрос: GET api/v1/student

Формат ответа:

```
3 HTTP/1.1 200 OK
4 {
5   "course": "Integer",
6   "full_name": "String",
7   "group": "String",
8   "kaf": "String",
9   "np": "String",
10  "semester": "Integer",
11  "up": "String"
12 }
```

Описание полей:

course — курс, на котором сейчас находится студент.

full_name — ФИО студента.

group — учебная группа.

kaf — кафедра студента.

np — направление подготовки.

semester — текущий семестр (первый или второй).

up — профиль обучения.

Пример ответа:

```
14 HTTP/1.1 200 OK
15 {
16   "course": 1,
17   "kaf": "ИПОВС",
18   "full_name": "Иванов Иван Иванович",
19   "group": "МП-15",
20   "semester": 2,
21   "np": "Программная инженерия",
22   "up": "Программные технологии распределенной обработки информации"
23 }
```

3.7 Получение информации о дисциплинах

Запрос: GET api/v1/student/dis?semester=<semester>&year=<year>

Значение параметров:

<semester> — семестр, дисциплины за который хотим получить.

<year> — год, к которому относится семестр.

Если параметры не указаны, будут получены дисциплины за текущий семестр.

Формат ответа:

```
3 HTTP/1.1 200 OK
4 [
5   {
6     "ball": "Float",
7     "dateExam": "String",
8     "formControl": "String",
9     "id": "Integer",
10    "kaf": "String",
11    "maxBall": "Float",
12    "name": "String",
13    "semester": "Integer",
14    "teachers": [ "String" ]
15  }
16 ]
```

Описание полей:

ball — текущий балл по дисциплине. dateExam — дата экзамена в формате YYYY-MM-DD

ISO 8601. `formControl` — форма контроля. Возможные значения: Зачёт, Дифференцированный зачёт, Экзамен, Курсовая работа, Курсовой проект, Защита ВКР.

`id` — уникальный идентификатор дисциплины.

`kaf` — кафедра, преподающая предмет.

`maxBall` — максимальный балл по дисциплине.

`name` — название дисциплины.

`semester` — семестр, в котором преподают дисциплину.

`teachers` — ФИО преподавател(я/ей).

Пример ответа:

```
18 HTTP/1.1 200 OK
19 [
20   {
21     "ball": 100.0,
22     "dateExam": "2018-06-25",
23     "formControl": "Зачёт",
24     "id": 25,
25     "kaf": "ИГД",
26     "maxBall": 55.0,
27     "name": "3D моделирование",
28     "semester": 6,
29     "teachers": [ "Соколова Татьяна Юрьевна" ]
30   }
31 ]
```

4 ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Входные данные:

- токен аутентификации или логин/пароль для получения токена;
- запрашиваемый ресурс (аутентификация, дисциплины и т.д.);
- имя и версия приложения + операционная система клиента;
- HTTP-заголовок Accept;
- тип HTTP-запроса (GET, POST);
- тело запроса (для создания новой сущности), если запрос имеет тип POST.

Выходные данные:

- код ответа сервера;
- описание ошибки, если запрос некорректный;
- запрашиваемые данные (если запрос типа GET).

Формат запроса:

```
1 GET /api/v1/example HTTP/1.1
2 Accept: application/json
3 Authorization: Bearer <token>
4 User-Agent: <app>/<app-version> <os>
```

Значение параметров:

<token> — токен доступа.

<app> — имя приложения, использующего токен.

<app_version> — версия приложения, использующего токен.

<os> — операционная система, на которой запущено приложение (желательно с указанием версии).

5 СООБЩЕНИЯ

В ходе работы ПМ ИСП программисту выдаются сообщения. Здесь приводится описание их содержания и действия, которые необходимо принять по этим сообщениям.

Ошибка “400 Bad Request”

Возможные описания:

- “Отсутствует обязательный заголовок {}”;
- “Недопустимый формат заголовка {}”;
- “Отсутствует обязательный ключ {}”;
- “Недопустимый формат ключа {}”.

Где {} - заголовок или ключ, вызвавший ошибку.

Для устранения ошибки нужно добавить недостающий заголовок или ключ, или исправить формат уже существующего.

Ошибка “401 Unauthorized: Несуществующий или аннулированный токен.”

Возникает, если передан несуществующий или аннулированный токен в заголовке Authorization. Для устранения ошибки нужно передать верный токен.

Ошибка “403 Forbidden: Один студент может получить не более восьми токенов.”

Возникает при попытке получить более восьми токенов одним пользователем. Чтобы устранить ошибку, пользователь должен удалить, которые не используются, освободив этим место для новых токенов.

Ошибка “404 Not Found: Отсутствует ресурс по данному URI.”

Возникает при обращении по несуществующему URL. Для устранения поверьте, что введен правильный URL ресурса. Если URL верный, следует сообщить об ошибке в поддержку API ОРИОКС.

Ошибка “405 Method Not Allowed”

Описание имеет формат: “Метод {} запрещён для данного ресурса”.

Где {} — один из методов: POST, PATCH, PUT, DELETE.

Для устранения ошибки следует использовать правильный метод доступа.

Ошибка “410 Gone: Данная версия API устарела.”

Возникает, если происходит обращение к старой версии API, которая более не поддерживается и была удалена. Для устранения ошибки следует использовать новую версию API.