

## **ПРОЕКТ СЕРВИСА УПРОЩЕНИЯ РАБОТЫ С ИС «Модеус»**

**Аннотация.** В статье представлен процесс разработки сервиса для упрощения работы с ИС «Модеус» путем синхронизации его календаря со сторонними сервисами, более подробно рассматривается синхронизация с Google Calendar.

**Ключевые слова:** Модеус, Управление учебным процессом, Гугл Календарь, Веб-сервис, Цифровая технология, синхронизация.

### **ВВЕДЕНИЕ**

В нескольких университетах России используется информационная система «Модеус». Она предназначена для составления и показа расписания как для преподавательского состава, так и для студентов, учета успеваемости и посещаемости. Можно скачать расписание на неделю в формате iCal [1]. Однако, у этой системы есть несколько недостатков, которые можно решить созданием стороннего сервиса для работы с ним. Среди главных проблем можно выделить: отсутствие удобной синхронизации календаря Модеуса с личным или рабочим календарем в стороннем сервисе, уведомления приходят только на университетскую электронную почту. Каждый раз для получения доступа необходима авторизация в системе, push-уведомления или уведомления на личную почту не приходят.

Требуется разработать дополнительный сервис, который улучшит работу с системой «Модеус». Сервис должен синхронизировать календарь занятий и личный календарь, такой как Google Calendar, присылать уведомления при изменении расписания занятий, сократить время доступа к расписанию занятий.

## 1. ВЫБОР КАЛЕНДАРЯ

В качестве синхронизируемого календаря был выбран Google Calendar по нескольким причинам: он есть во всех Android-смартфонах, и его можно без проблем поставить на iOS. Google Calendar умеет отправлять push-уведомления в браузеры для напоминания о событиях, в данном случае событиями являются занятия в университете. Google имеет очень удобный и понятный API [2], с которым просто работать. Также Google Calendar тесно интегрирован с остальными сервисами Google. Полезным для нас будет Google Maps, который будет показывать, где будет проходить занятие.

## 2. РАЗРАБОТКА СЕРВИСА

### ***2.1 Определение принципа работы сервиса авторизации Modeus***

Для определения принципа работы сервиса авторизации использовались Chrome Dev Tools и Fiddler. При первичном заходе на главный URL Modeus – utmn.modeus.org идет перенаправление на сервис аутентификации fs.modeus.org. В свою очередь, fs.modeus.org делает запрос на auth.modeus.org на генерацию id сессии, который потом используется на fs.modeus.org для авторизации. Если напрямую зайти на auth.modeus.org, то мы попадем на админ-панель WSO2 Identity Server – средства для управления доступом к различным веб-приложениям с открытым исходным кодом.

На WSO2 Identity Server есть документация[3], в которой указано, что у Identity Server есть API для авторизации. Но, при попытке обратиться к API auth.modeus.org на клиенте возникает ошибка CORS – cross-origin resource sharing [4]. Получается, что на API auth.modeus.org можно попасть только с домена modeus.org, и написать запрос для работы с API вне этого домена невозможно, и нужно использовать другие методы для работы с Modeus.

### ***2.2 Обход ограничения доступа с других доменов***

Для обхода ограничения CORS было решено имитировать работу пользователя в браузере с помощью программных средств. Было решено для этого использовать библиотеку Puppeteer [5] для Node.js, главным языком для разработки всего сервиса стал также Node.js [6].

Puppeteer имитирует работу человека в браузере: можно заполнять поля, нажимать на кнопки, ждать появления элементов на странице, на это все у Puppeteer есть соответствующие методы.

В начале работы с Puppeteer было обнаружено, что Modeus защищен от подобного рода манипуляций с помощью невидимого div'a, сообщаящим об отсутствии поддержки браузером JavaScript, однако, добавив несколько настроек в Puppeteer данная проблема была решена, и можно было приступить к получению расписания на неделю в формате iCal, эта функция предусмотрена самим Модеусом.

Работа программы имитации работа браузера: изначальный URL – utmn.modeus.org, при заходе на utmn.modeus идет JS-редирект на auth.modeus, где получается токен для логина, потом идет опять JS-редирект на fs.utm – страницу входа. Ждем рендера нужных элементов формы авторизации в Modeus, метод ожидания Puppeteer принимает одним из аргументов CSS-селектор, при нахождении которого выполнение кода продолжается. Вводим логин и пароль, отправляем форму. Ждем загрузки календаря, и ищем CSS-селектором div, при нажатии на который скачивается файл расписания. Вызываем метод нажатия на найденный div и получаем файл расписания в формате iCal.

Из-за того, что пока не написан клиент сервиса, учетные данные Modeus передаются через терминал. У Modeus нет особенностей авторизации, и для входа хватит только логина и пароля.

## **2.2 Обработка iCal-файла.**

После скачивания получаем iCal-файл, который представляет собой массив объектов со следующей структурой и примерными значениями:

- UID: '4094ae0b-5ca5-4706-ab6d-71aab3b10101',
- DateStart: 20210521T140000
- DateEnd: 20210521T153000
- Summary: Информационные технологии в прикладных системах / Классификация документов (Naive Bayes Classifier) / ИТвприкладныхсистемах-ЛБ-22ИСИТ189-1

- Location: УЛК-05 / 306 (УЛК-05)
- Description: Лабораторное занятие / Учебно-исследовательская работа
- Преподаватель: Бидуля Юлия Владимировна + посмотреть в полной версии + URL.

Данный файл считывался модулем fs Node.js как обычный текст, затем с помощью библиотеки ical2json [7] преобразуем iCal в JavaScript-объект. На выходе получаем массив объектов. Каждый объект нужно обработать, в основном нужно было поменять названия нескольких полей, чтобы это правильно работало с Google Calendar API. Поля Summary, Description, Преподаватель, и номер аудитории из Location были объединены. В поле преподавателя осталось только ФИО преподавателя. Из полей DateStart и DateEnd были созданы JS-даты, для API Google Calendar помимо даты и времени нужен часовой пояс, который был вшит в поле DateStart и DateEnd. Была написана функция, которая принимает название корпуса, и возвращает адрес, к примеру: getAddress('УЛК-05 / 306 (УЛК-05)') вернет 'Перекопская ул., 15АТюмень, Тюменская обл., 625003'. Это нужно для Google Maps. Событие может содержать место, и в таком формате при нажатии на адрес откроется Google Maps и покажется место.

### ***2.3. Перенос данных в Google Calendar***

Для работы с Google Calendar API нужно сначала получить ключ API и выдать права ключу на календарь в Google Cloud Platform [8]. Этот ключ, и токены для работы с ним зашиваются в googleauth из Node.js-модуля googleapis [9].

Потом вызываем метод добавления события Google Calendar из модуля googleapis, куда передается JS-объект события.

Так как для данного сервиса пока не написан клиент, к ключу API был привязан Google-аккаунт через сервис Google OAuth 2.0 Playground [10]. Когда будет написан фронт, авторизация будет обычной через Google Account, с запросом на использование календаря пользователя.

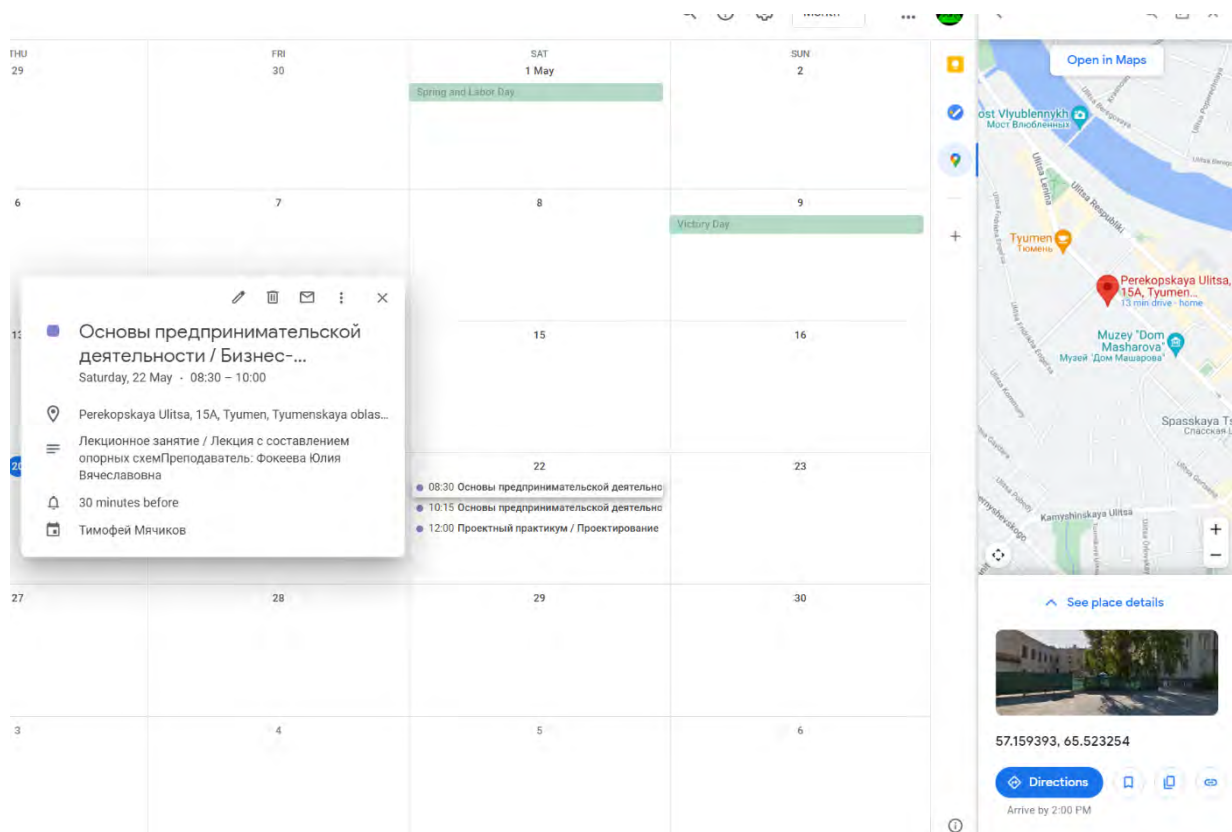


Рис. 1. Занятия показываются в Google Calendar.

#### 2.4. Уведомления об изменении расписания

Для мониторинга изменения расписания вычисляется контрольное значение каждого занятия, у каждого занятия есть ID. Контрольное значение складывается из строк:

$$V = DateStart + DateEnd + Location,$$

где DateStart – начало события, DateEnd – конец события, Location – место.

PostgreSQL [11] был выбран в качестве базы данных проекта. В БД есть таблица следующего вида: ID урока: контрольное значение. С определенным интервалом запускается перерасчет контрольного значения. Оно поменяется, если поменялось хотя бы одно значение, из которого оно складывается. Если оно поменялось, то в браузер идет push-уведомление [12], идет поиск события в Google Calendar и изменение данных, в БД заносится обновленное значение контрольного значения.

## ЗАКЛЮЧЕНИЕ

В процессе работы над проектом поставленные задачи были выполнены: авторизация в Google и Modeus, получение расписания и его перенос в Google Calendar, приходят push-уведомления в браузер при изменении расписания.

Планируется создать веб-клиент для удобной работы с сервисом, с нормальной авторизацией в Google-аккаунт и аккаунт Modeus. Также нужно сделать возможной работу не только с Google Calendar, но и другими календарями, в первую очередь Apple iCloud Calendar.

## СПИСОК ЛИТЕРАТУРЫ

1. Страница с информацией об ИС «Модеус». URL: <https://modeus.custis.ru/>
2. Google Calendar API. URL: <https://developers.google.com/calendar>.
3. WSO2 Identity Server Documentation. URL: <https://is.docs.wso2.com/en/latest>.
4. Энциклопедия Mozilla Developer Network: статья “Cross-origin resource sharing”. URL: <https://developer.mozilla.org/ru/docs/Web/HTTP/CORS>
5. Документация Puppeteer. URL: <https://pptr.dev/>
6. Документация Node.js. URL: <https://nodejs.org/en/docs/>
7. Документация iCal2json. URL: <https://www.npmjs.com/package/ical2json>
8. Google Cloud. URL: <https://cloud.google.com/>
9. Node.js Google Calendar Node.js Quickstart. URL: <https://developers.google.com/calendar/quickstart/nodejs>
10. OAuth 2.0 Playground. URL: <https://developers.google.com/oauthplayground/>
11. Документация PostgreSQL. URL: <https://www.postgresql.org/docs/>
12. Документация Google Chrome Developers: Notifications URL: <https://developer.chrome.com/docs/extensions/reference/notifications/>