



С.О. Кашкинов, Д.В. Еленев

ПРОБЛЕМА УЧЕТА ПОСЕЩАЕМОСТИ ЗАНЯТИЙ В BIGBLUEBUTTON

(Самарский университет)

В течение 2020 года в связи с распространением новой коронавирусной инфекции дистанционный формат проведения занятий стал занимать существенную долю в образовательном процессе университетов. Применение дистанционных образовательных технологий позволяет сократить издержки на проведение аудиторных занятий и позволяют обучающимся получать знания и полезные навыки находясь на удалении от учебного заведения. Множество курсов, учебных программ в университетах и колледжах, различные конференции зачастую проводятся в режиме онлайн с помощью различных платформ электронного обучения.

Для организации взаимодействия студентов и преподавателей необходимо использовать различные программные продукты и системы для организации видеоконференций. Одной из самых популярных программ такого рода является BigBlueButton — открытое программное обеспечение для проведения веб-конференций [1]. Система разработана в первую очередь для дистанционного обучения и нашла широкое применение в Самарском национальном исследовательском университете имени академика С.П. Королева (Самарский университет) и других вузах.

Проведение занятий с использованием BigBlueButton имеет несколько технических ограничений. К наиболее существенным относятся ограничение на количество одновременно работающих комнат (сессий, или проводимых занятий) и на количество пользователей в одной комнате (максимально до 100 пользователей). Для проведения большего количества одновременных сессий работы в Самарском университете применяется автоматический балансировщик нагрузки, распределяющий создаваемые сессии по пулу имеющихся отдельных виртуальных серверов. Для входа в систему BigBlueButton используется единая система аутентификации пользователей электронной информационно-образовательной среды Самарского университета, построенная с использованием технологии Active Directory на основе кадрового учета.

При проведении занятий преподаватель отмечает студентов, присутствующих на занятии, для чего используется функционал личных кабинетов работников и обучающихся – корпоративного web-приложения Самарского университета [2]. Однако учет посещаемости с проставлением отметок в электронном журнале учета посещаемости и успеваемости (рисунок 1) производится в ручном режиме. В случаях, когда занятия проводятся для малых групп с аудиторией не более 10 человек, то это не составляет труда, но при работе с аудиторией в несколько десятков человек, например, при потоковых лекциях, процесс учета посещаемости не только становится трудоемким процессом, но и еще отнимает время проведения занятий.

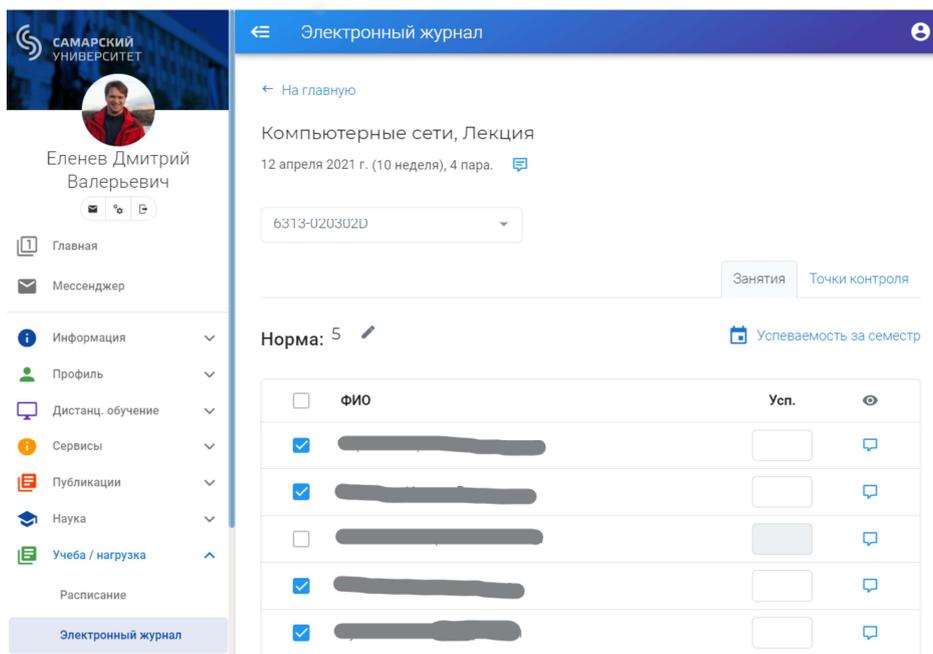


Рисунок 1 – Учет посещаемости в ЛК преподавателя

В связи с этим была поставлена задача автоматизации учета посещаемости путем разработки дополнительных модулей, обеспечивающих автоматическую фиксацию входа и присутствия пользователей в заранее определенных комнатах. Адрес комнаты считается заранее известным и предоставленным преподавателем.

Система автоматического учета посещаемости ориентирована на решение следующих задач:

1. Получение расписания групп из базы данных личных кабинетов.
2. Оборачивание полученных данных в классы Java.
3. Анализ лог-файла и оборачивание необходимых данных в Java классы.
4. Сопоставление данных расписания и лог-файла.
5. Анализ полученной информации и сохранение учета посещаемости в БД.

Данные задачи стали отражением анализа существующих методов взаимодействия с BigBlueButton.

Существуют несколько методов получения информации из сервера BBV:

1. Работа с API BBV. BigBlueButton предоставляет API для взаимодействия со своим сервером. Форма отправки запроса показана на рисунке 2.

```
https://доменное_имя_сервера_bbb/bigbluebutton/api/метод_api?параметр=1&checksum=123
```

Рисунок 2 – Шаблон запроса на сервер BBV

Checksum представляет из себя контрольную сумму полученную методом SHA1 из строки, в которую входят:



- имя вызова (метод api),
- строка запроса (параметра),
- секрет (общий секрет безопасности, который можно получить из вашего сервера ВВВ), из всей этой строки получается.

API ВВВ больше подходит для интеграции BigBlueButton в другое приложение, так как там есть методы предоставления записей, получение конфигурации и присоединения к митингу, а не получения статистики по посещаемости комнат обучающимися.

2. Анализ лог-файла bbb-web.log. Данные в лог-файле можно разделить на абстрактные блоки. Каждый блок – это отдельное действие происходящее на сервере. Рассмотрим на примере присоединение нового участника конференции (рисунок 2). Каждая новая строка начинается с указания времени в java типе DateTime. Первой строкой идет название вызванного метода API. После этого идет проверка переданной checksum и checksum посчитанной на сервере. Если checksum одинаковая, то указываются данные сессии и привязка определенных данных к подключающемуся пользователю.

По данному лог-файлу можно проследить, когда создается отдельная сессия, какое у него название и кто к нему подключается, поэтому данный способ оказался приоритетным для решения задачи учета посещаемости в режиме онлайн на платформе ВВВ.

```
2021-04-26T01:47:46.390-05:00 DEBUG o.b.web.controllers.ApiController - ApiController#join
2021-04-26T01:47:46.391-05:00 INFO o.b.api.ParamsProcessorUtil - CHECKSUM=c4d8d3f379c8ffcf4070bf0ac92abfcd44770ca0 length=40
2021-04-26T01:47:46.392-05:00 INFO o.b.api.ParamsProcessorUtil - CHECKSUM=c4d8d3f379c8ffcf4070bf0ac92abfcd44770ca0 length=40
2021-04-26T01:47:46.392-05:00 INFO o.b.web.controllers.ApiController - Retrieving meeting cf82cc67a2f800f70208dc50cd269f5c028ec9fa
2021-04-26T01:47:46.393-05:00 INFO o.b.web.controllers.ApiController - Session sessionToken for User 6006197 [null]
2021-04-26T01:47:46.394-05:00 INFO o.b.web.controllers.ApiController - Session user-token for User 6006197 [null]
2021-04-26T01:47:46.396-05:00 INFO o.b.web.controllers.ApiController - --analytics-- data={"extMeetingid":"random-4052213","logCode":"join_api","name":"User
6006197","sessionToken":"qeou1ovfm48zpsrb","description":"Handle JOIN API.", "meetingid":"cf82cc67a2f800f70208dc50cd269f5c028ec9fa-1619419481777","userid":"w_ppwhjtx21nv7"}
2021-04-26T01:47:46.397-05:00 INFO o.b.web.controllers.ApiController - Redirecting to http://10.0.2.15/html5client/join?sessionToken=qeou1ovfm48zpsrb
```

Рисунок 2 – Пример блока присоединения нового участника

На основе выбранного способа можно продемонстрировать пошаговый словесный алгоритм работы данной программы:

1. Приложение разворачивается на сервере с ВВВ.
2. Из базы данных достаются расписание по группам на сегодняшний день вместе с списком студентов этих групп.
3. Для большего удобства анализа полученные данные помещаются в Java классы: название группы, список <студент: имя>, список <занятия: название, время, url ссылка на серверную комнату>.
4. Запрашивается информация из лог-файла bbb-web.log.
5. Необходимая информация извлекается из этого файла: время события, тип события (создание митинга, добавление пользователя к митингу, удаление к митингу), основные данные по произошедшему событию.
6. Полученные данные анализируются, каждой сессии определяется свой Java объект вида: идентификатор сессии, название, url комнаты создания (у каждого преподавателя есть своя комната, от которой создается сессия), спи-



сок присутствующих пользователей (студентов), время начала, время окончания.

7. Сопоставляем данные из расписания и списки студентов группы с теми, что получились при анализе. Сопоставление происходит по URL комнаты и времени начала занятия. Все студенты, не присутствующие на занятиях, удаляются из списков.

8. Получившиеся списки студентов отправляются в БД расписания.

Литература

1. API BigBlueButton [Электронный ресурс]. URL: <https://docs.bigbluebutton.org/dev/api.html> (дата обращения: 15.04.2020).

2. Еленев Д.В., Линник А.О. Личные кабинеты научно-педагогических работников университета // Известия Самарского научного центра Российской академии наук. 2015. Т. 17. № 2-5. С. 964-969.

Д.Д. Князева, Д.А. Попова-Коварцева

ПРОГРАММНЫЙ ТРЕНАЖЕР «ОПЕРАЦИИ РЕЛЯЦИОННОЙ АЛГЕБРЫ»

(Самарский университет)

Информация и данные в современном обществе рассматриваются как жизненно важные, ключевые ресурсы. Базы данных составляют основу информационных банковских систем, систем управления качеством, систем бронирования и многих других. Отсюда логично вытекает потребность в кадрах, имеющих навыки создания, отладки, модернизации, обслуживания баз данных.

Реляционная модель баз данных получила широкое распространение поскольку характеризуется простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных. Несмотря на кажущуюся простоту, очень часто обучающиеся испытывают трудности при освоении практикума по применению операций реляционной алгебры над данными. Поэтому разработка тренажера, который позволит студентам отрабатывать знания на практике без необходимости ожидания проверки выполненного задания преподавателем, является актуальной задачей.

Реляционная алгебра задает набор операторов для выполнения операций над реляционными отношениями. Операции реляционной алгебры можно разделить на две группы: *базовые теоретико-множественные* и *специальные реляционные*.

1. Первая группа операций включает в себя операторы, представляющие собой традиционные операции над множествами, а именно:

- объединение отношений;
- пересечение отношений;
- разность (вычитание) отношений;