

процессов. Лучшие результаты дает решение в виде разложения по степеням t и анализ численного решения. Из численного решения следует, что $\frac{\rho_{33}}{\rho_{22}} = 0.27$. Другими словами,

резонансное двухфотонное поглощение может усиливать поглощение на переходе ${}^3H_6 \rightarrow {}^3F_4$ до 27 % процентов. Этот вывод хорошо согласуется с экспериментальными данными.

Таким образом, противоречие между наблюдаемой интенсивностью поглощения на переходе ${}^3H_6 \rightarrow {}^3F_4$ и временем жизни 3F_4 можно устранить при учете дополнительного или сверх поглощения, обусловленного резонансными двухфотонными процессами.

УДК 004.4

РАЗРАБОТКА REST-СЕРВИСА ДЕКАНАТА УНИВЕРСИТЕТА

Кузнецов А.А., д.т.н., проф., Казаков В.Е., к.т.н., доц., Осинковский Ю.В., маг.

*Витебский государственный технологический университет,
г. Витебск, Республика Беларусь*

Реферат. В статье рассматривается разработка REST-сервиса как серверной части веб-приложения деканата университета.

Ключевые слова: деканат, веб-приложение, REST, API, JSON, Spring.

Деканат – возглавляемый деканом организационный центр по управлению работой факультета. Деканат выполняет функции координации и административного обеспечения учебного процесса, ведения делопроизводства. Здесь составляется расписание занятий. Деканат контролирует работу преподавателей и студентов на предмет её соответствия учебному плану, осуществляет общее руководство научной работой студентов.

В перечень его основных задач входит:

- контроль над деятельностью преподавателей (выполнение учебного плана, ведение документации и пр.);
- оценка успеваемости студентов;
- совместная работа с приемной комиссией в части приема и зачисления абитуриентов;
- организация учебной деятельности на факультете (утверждение учебных планов, индивидуальной программы обучения, утверждение расписания занятий и пр.);
- работа со студентами: решение организационных и общих вопросов, консультирование учащихся по отдельным и личным вопросам, предоставление справок и иных документов по месту требования;
- оформление приказов и распоряжений (о зачислении, отчислении, переводе и пр.).

Использование веб-сервиса для автоматизации работы деканата существенно сократит время работы основных бизнес-процессов, уменьшит количество невынужденных ошибок.

Для реализации веб-сервиса необходима клиентская часть и серверная часть.

Клиентская часть веб-приложения – это часть веб-приложения, которая работает в браузере пользователя в виде html-страницы, отображает основной контент и может взаимодействовать с серверной частью используя различные протоколы.

Серверная часть веб-приложения – это сервер, который предоставляет клиентской части API [1] для взаимодействия с клиентом.

В качестве архитектуры веб-приложения выбрана REST [2]. Следовательно, все взаимодействие будет происходить через HTTP-протокол. А вся информация будет передаваться в JSON-формате.

Основные API, которые необходимо реализовать, – это получение всех пропусков студента, в том числе отработанных и нет, получение оценок студента по соответствующей дисциплине за конкретный семестр, получение долгов студента по дисциплине.

Исходя из поставленных задач, в качестве парадигмы разработки сервиса будет использована многослойная архитектура.

Многослойная архитектура является одной из архитектурных парадигм разработки ПО, при которой функциональные области приложения разделяются на группы.

Всю функциональность приложения, так или иначе, можно разделить на группы, в

зависимости от того, какая задача выполняется этой функциональностью, что приложение посредством неё делает (общается с пользователем или осуществляет валидацию данных, или манипулирует данными из базы, или описывает сущности, или описывает логику и т. д.).

Для реализации многослойной архитектуры будет использован Spring Framework [3]. В качестве слоев приложения будут использованы следующие слои:

- Repository – слой взаимодействия с базой данных;
- Service – слой реализации бизнес логики;
- Controller – слой, в котором предоставляются API для взаимодействия с клиентом.

Архитектура сервиса представлена на рисунке 1.

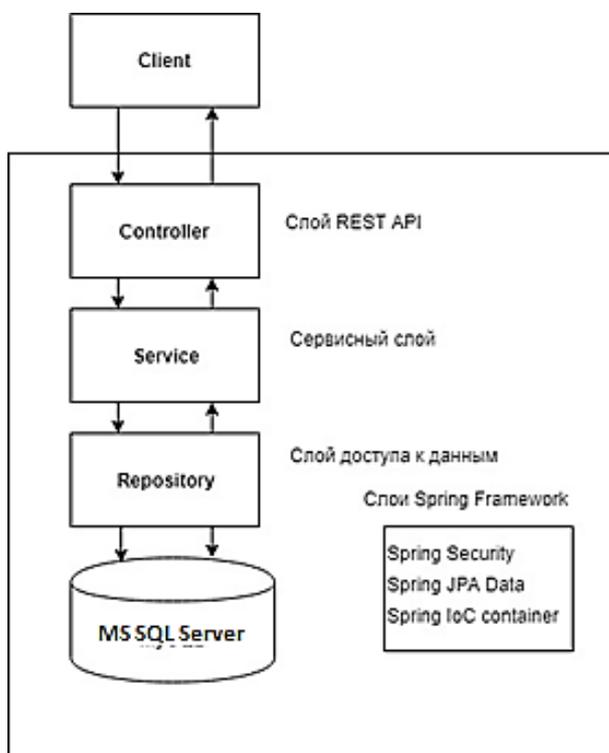


Рисунок 1 – Архитектура REST сервиса

Фреймворк легко позволяет реализовать взаимодействие между слоями, путем внедрения одного слоя в другой.

Для того, чтобы защитить API будет реализована интеграция с сервисом авторизации.

Чтобы получить доступ к API необходимо получить токен у сервиса авторизации и установить его значение в заголовок Authorization с каждым запросом к API. Сервис авторизации будет проверять токен на валидность и давать доступ к соответствующему API.

Таким образом выбрана архитектура для реализации REST-сервиса деканата университета. Для расширения API достаточно реализовать соответствующую бизнес-логику и предоставить url для доступа.

Список использованных источников

1. Что такое API? [Электронный ресурс]. – Режим доступа: <https://dev.by/news/chto-takoe-api-prostym-yazykom>. – Дата доступа: 04.04.2020.
2. Alex Rodriguez. RESTful Web services: The basics. IBM [Электронный ресурс]. – Режим доступа: <http://www.ibm.com/developerworks/library/ws-restful/>. – Дата доступа: 8.05.2020.
3. Spring Framework [Электронный ресурс]. – Режим доступа: <https://spring.io/projects/spring-framework>. – Дата доступа: 07.04.2020.