

УДК 004.021

**ПРОЕКТИРОВАНИЕ API И БАЗЫ ДАННЫХ ДЛЯ ВЕБ-ПРИЛОЖЕНИЯ,
РАЗРАБОТАННОГО ДЛЯ ОБМЕНА ИНФОРМАЦИЕЙ И ФАЙЛАМИ
В ОГРАНИЧЕННОМ КРУГУ ЛИЦ****П.В. МАХАНЬКО***(Представлено: Ю.Н. КРАВЧЕНКО)**Рассматриваются цели и принципы моделирования API и базы данных для веб-приложения.*

Введение. В наше время трудно представить хорошее веб-приложение без разработанной базы данных и быстрого, оптимизированного API. База данных позволяет удобно хранить большие объемы информации не на устройстве пользователя, а на удаленном сервере, а API позволяет обрабатывать запросы веб-приложения, разграничивать права доступа и выдавать данные из базы данных, на любое устройство, где находится веб-приложение.

Данная статья направлена на описание базы данных и API для разработанного приложения.

Основная часть. Основной целью разработки API, является создание промежуточного звена между клиентской частью веб-приложения и базой данных. Это является отличным способом защиты пользовательских данных, а также обеспечение быстрого доступа к информации, не нагружая устройство пользователя.

В качестве защиты используется JSON Web token. Это открытый стандарт, создания токенов доступа основываясь на json. В момент авторизации пользователя, сервер генерирует хеш-ключи, которые отправляет клиентской части веб-приложения. В дальнейшем, каждый запрос к серверу, должен содержать в себе хеш-ключ для подтверждения прав доступа к ресурсам.

В качестве точек для запросов, сервер предоставляет веб-приложению конечные адреса, обращаясь к которым, веб-приложение получает определенные данные. Запросы проходят по стандартному HTTP протоколу, используя методы GET, POST. В качестве точек доступа, сервером представлены следующие адреса:

- Get /api/classmates - получение списка одноклассников;
- Get /api/classmate?classmate_id - получение информации об однокласснике по id;
- Get /api/user?user_id - получение персональной информации по id;
- Get /api/dialog?user_id получение диалогов по id;
- Get /api/new_messages?user_id получение новых сообщений с момента последнего визита в приложение;
- Get /api/last_visit_dialog?user_id - получение 5 последних диалогов с учетом новых сообщений;
- Get /api/newsfeed?user_id - получение новостной ленты определенного пользователя;
- Post /auth/login - авторизация пользователя;
- Post /auth/register - регистрация пользователя;
- Post /auth/register_complete завершение регистрации.

Чат и обмен файлами реализованы по технологии сокетов (протокол WebSocket). Сокет - протокол дуплексной связи между клиентом и сервером в режиме реального времени. В данном случае, веб-приложение подписывается на новые сообщения при успешном входе в приложение. Затем при получении нового сообщения оно мгновенно будет отправлено пользователю.

В качестве базы данных была выбрана документно-ориентированная база данных MongoDB. Синтаксис MongoDB схож с форматом json, а это в свою очередь не задает жестких рамок хранения данных, что позволяет удобно расширять хранимые данные. Вся база данных состоит из коллекций, а хранимые в ней данные - документы. Документ представляет из себя json-объект в виде пар ключ/значение, где в качестве значения могут храниться строки, числа, двоичные данные, массивы или другие вложенные объекты.

Реляционная модель представляет собой базу данных в виде множества взаимосвязанных отношений (таблиц), которые используются для хранения информации об объектах, представленных в базе данных.

В соответствии с требованиями программы, определены коллекции (набор) необходимых таблиц базы данных, представленный в таблице.

Таблица. – Перечень необходимых таблиц базы данных

№ п/п	Название коллекции	Описание
1	Users	Список всех пользователей
2	Groups	Список всех учебных групп
3	Pulpits	Список всех кафедр
4	Faculties	Список все факультетов
5	Messages	Сообщения пользователей
6	Newsfeeds	Новостные ленты пользователей

Важное достоинство баз данных – возможность хранения сгруппированных данных в разных коллекциях и в разных форматах.

Заключение. В данной статье были рассмотрены основные цели и принципы моделирования базы данных, построение быстрого, удобного API для веб-приложения. Были рассмотрены основные сущности базы данных. Спроектированы адреса для обращения клиентской части веб-приложения к серверной части.