

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ОТСЛЕЖИВАНИЯ РАСПИСАНИЯ СТУДЕНТАМИ ФАКУЛЬТЕТА РАДИОФИЗИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Е. Д. Искренков

Белорусский государственный университет, г. Минск;

e.iskrenkov@gmail.com;

науч. рук. – Ю. А. Бондаренко, ассистент

В данной работе была выполнена разработка веб-приложения для просмотра и редактирования расписания учебных занятий студентами факультета радиофизики и компьютерных технологий на основе веб-фреймворка Ruby on Rails [1]. Предусмотрен доступ к данным, хранящимся на сервере приложения, при помощи как веб-браузера, так и популярного на данный момент мессенджера Telegram, с использованием чат-бота. Разработан открытый API, предоставляющий любому желающему доступ к данным и, следовательно, позволяющий разрабатывать клиенты для любых операционных систем (ОС) с использованием соответствующих технологий. Приложение было введено в эксплуатацию в 2019 году и активно используется по сей день.

Ключевые слова: веб-приложение; Ruby; Ruby on Rails; Telegram; API; MySQL; реляционные базы данных.

ВВЕДЕНИЕ

На данный момент в БГУ не существует единого, контролируемого самим университетом портала для публикации изменений в текущем учебном расписании, что несет собой неудобство как для самих студентов, так и для старост групп, которым собственноручно приходится любым возможным способом доносить расписание и изменения в нем до учащихся. Целью данной работы было решить данный вопрос, создав удобный, доступный и легко масштабируемый инструмент для сохранения, обновления и просмотра расписания студентами факультета.

ТЕХНОЛОГИИ

При выборе технологий для реализации задуманного проекта важнейшими критериями являлись, в первую очередь, простота и скорость разработки. Выбор языка программирования пал на язык Ruby, разработанный японским разработчиком Юкихио Мацумото в 1995 году [2]. Философией языка является удобство и минимизация затрат труда программиста при разработке программы, освобождение программиста от рутинной работы, которую компьютер может выполнять быстрее и качественнее, что как нельзя лучше соответствует нашим требованиям. Кроме того, неоспори-

мым преимуществом Ruby перед другими объектно-ориентированными языками программирования является наличие реализованного на нем и свободно распространяемого веб-фреймворка Ruby on Rails, который был создан с уважением к концепциям и принципам самого Ruby – простоте и легкости в использовании. Ruby on Rails обладает минимальным значением Time to Production (время до релиза приложения) и значительно расширяет функциональные возможности языка, предоставляя огромный инструментарий для создания типовых веб-приложений.

Для сохранения данных, введенных пользователем, приложение использует свободную реляционную базу данных MySQL [3]. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы. Для удобного доступа к данным в коде приложения используется технология программирования ORM (Object-Relational Mapping), реализованная в виде отдельного фреймворка, входящего в состав Ruby on Rails, – ActiveRecord.

Кроме того, приложение взаимодействует с популярным мессенджером Telegram для быстрой и удобной доставки расписания. Для этой цели используется Telegram API [4], который требует предварительной регистрации чат бота и получения уникального API-токена для дальнейшей работы.

СХЕМА БАЗЫ ДАННЫХ

Как было упомянуто ранее, приложение взаимодействует с базой данных MySQL посредством ORM-фреймворка. Схема базы данных приложения приведена на рис. 1.

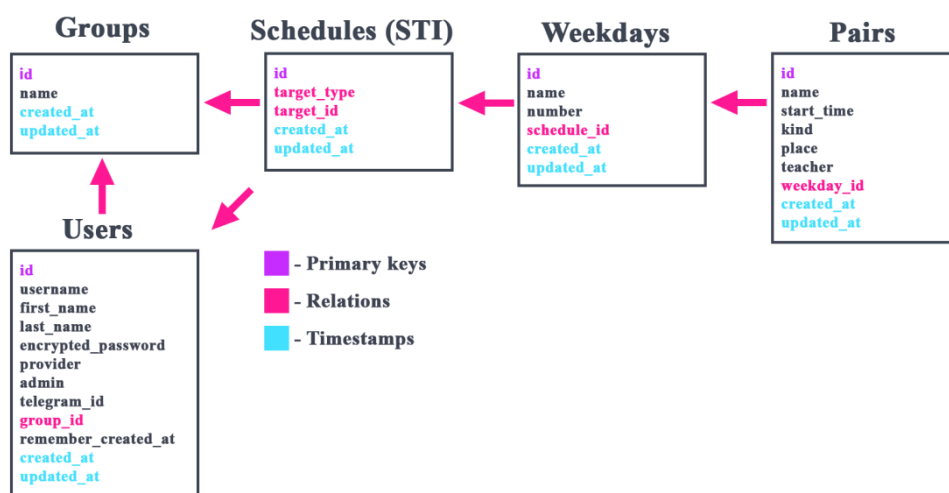


Рис. 1. Схема базы данных приложения

Главная сущность приложения – User, она представляет собой конкретного пользователя, зарегистрированного в системе. Как видно на

схеме, в таблице Users присутствует колонка *admin*, благодаря которой осуществляется проверка прав пользователя на редактирование расписания. Таким образом, любой пользователь может являться администратором и иметь доступ к созданию новых групп и изменению расписания существующих. Также, каждый User принадлежит своей группе, с которой, в свою очередь, связано конкретное расписание. В нем содержится 7 дней недели, каждому из которых соответствует список пар. Интересным моментом в схеме является то, что модель Schedule поддерживает механизм STI (Single Table Inheritance), что позволяет Расписанию принадлежать как всей группе, так и отдельному пользователю.

СТРУКТУРНАЯ СХЕМА ПРИЛОЖЕНИЯ

Приложение использует концепцию MVC (Model - View - Controller), что позволяет удобно и грамотно разделить его логику. Модели полностью соответствуют приведенной ранее схеме базы данных, так как каждая модель связана со своей таблицей. Рассмотрим схему классов контроллеров приложения, представленных на рис. 2.

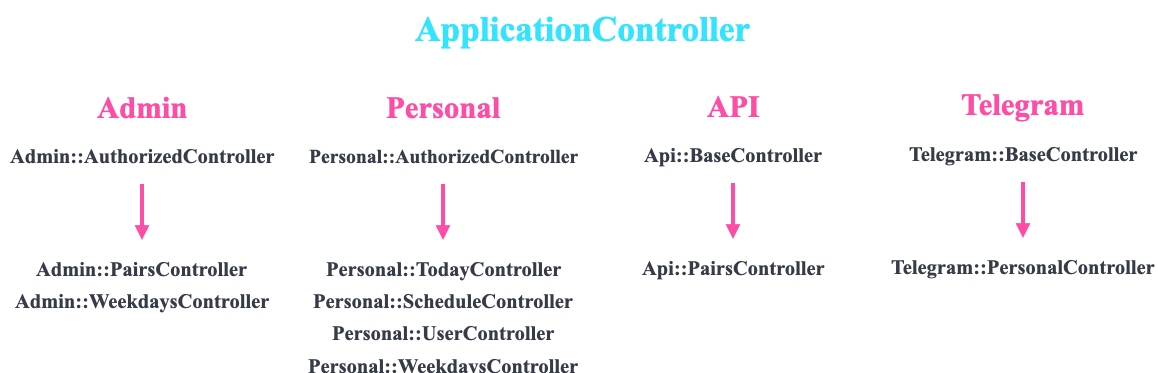


Рис. 2. Схема наследования контроллеров приложения

Как мы видим, все контроллеры наследуются от базового класса ApplicationController, предоставляемого фреймворком. Далее все контроллеры разбиты на 4 именные области, каждая из которых отвечает за свою часть приложения. Контроллеры, определенные в именной области Admin, наследуются от AuthorizedController, который содержит авторизационную логику администратора. Таким образом, все Admin контроллеры будут доступны только при наличии соответствующих прав. В области имен Personal также содержится базовый контроллер, проверяющий аутентификацию пользователя. Гость не имеет доступа к контроллерам этой области. Контроллеры области имен API предоставляют открытый API для разработчиков в то время, как контроллеры в области имен Telegram принимают и обрабатывают запросы Telegram бота.

ПРИНЦИП РАБОТЫ TELEGRAM БОТА

В процессе регистрации Telegram генерирует для каждого созданного бота уникальный API токен. Используя его, мы можем получить текущую очередь сообщений, отправленных кем угодно в чат с нашим ботом. Наша задача – запустить процесс-демон, который будет с некоторым интервалом опрашивать Telegram на предмет наличия обновлений. На данном этапе из всех предоставленных нам данных самым важным для нас является поле *id*. По его значению система находит зарегистрированного пользователя в базе данных. Определив пользователя, отправившего сообщение, нам не составит труда найти расписание, принадлежащее группе пользователя, и отправить ему текстовый ответ.

ВЫВОДЫ

Благодаря своей простоте и интуитивности фреймворк Ruby on Rails является выдающейся базой для создания любых по сложности, и что немаловажно, расширяемых веб-приложений. В данной работе с его помощью удалось с относительной простотой объединить веб-часть и часть, ответственную за обработку сообщений Telegram ботом в одном приложении. Кроме того, удалось создать удобную и интуитивную систему для отслеживания расписания студентами факультета радиофизики и компьютерных технологий, которая полностью открыта для разработчиков и имеет API для работы, что позволяет использовать ее для разработки клиентов под любые необходимые платформы.

Библиографические ссылки

1. Rails Guides [Электронный ресурс]. URL: <https://guides.rubyonrails.org/> (дата обращения: 17.04.2020).
2. Ruby Language Wiki [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/Ruby> (дата обращения – 20.04.2020).
3. MySQL 8.0 Reference Manual [Электронный ресурс]. URL: <https://dev.mysql.com/doc/refman/8.0/en/> (дата обращения – 19.04.2020).
4. Telegram API docs [Электронный ресурс]. URL: <https://core.telegram.org/bots/> (дата обращения – 19.04.2020).
5. Russ Olsen. Eloquent Ruby. Pearson Education, Inc., 2011.