Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных сетей и систем

Кафедра Информатики

ПРОЕКТ

по курсу «Современные технологии разработки программного обеспечения»

ТЕЛЕГРАМ-БОТ ДЛЯ ПОЛУЧЕНИЯ ИНФОРМАЦИИ О МУЗЫКАЛЬНЫХ ИСПОЛНИТЕЛЯХ

Студент:
гр. 7М2631
Лавник Т.С.

Проверил:
Стержанов М.В.

Минск, 2017

СОДЕРЖАНИЕ

1.ПОСТАНОВКА ЗАДАЧИ

1.1.Цель

Реализовать приложения для получения информации о музыкальных артистах на базе функциональности мессенджера Telegram, а именно, реализовать телеграм-бота с использованием технологий Ruby on Rails, Telegram Bot API, last.fm API, Ruby-библиотеки telegram-bot. Выбор вышеописанных технологий обсуловлен качествами, которые они предоставляют, а именно, быстрота и удобство разработки.

1.2.Задачи

1. Создать приложение Ruby on Rails, используя флаги для ограничения модулей, которые не используются в разработке телеграм-бота, а так же добавить зависимости для использования библиотеки telegram-bot.
2. Создать сервис, который будет служить соединительным звеном между приложением и last.fm API.
3. Реализовать модуль для создания телеграм-бота посредством использования библиотеки telegram-bot и подключить в него сервис для работы с last.fm API.

2.ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ

2.1.Ruby

Ruby это динамический, рефлексивный, объектно-ориентированный язык, который сочетает синтаксис, вдохновленный особенностями Perl с Smalltalk. Ruby возник в Японии в середине 1990х и впервые был разработан и спроектирован Юкихиро "Matz" Мацумото. Он (язык) находился главным образом под влиянием Perl, Smalltalk, Eiffel и Lisp.

Имеет лаконичный и простой синтаксис, частично разработанный под влиянием Ада, Eiffel и Python.

Позволяет обрабатывать исключения в стиле Java и Python.

Позволяет переопределять операторы, которые на самом деле являются методами.

Полностью объектно-ориентированный язык программирования. Все данные в Ruby являются объектами в понимании Smalltalk. Например, число «1» — это экземпляр класса Integer. Единственное исключение — управляющие конструкции, которые в Ruby, в отличие от Smalltalk, не являются объектами. Также поддерживается добавление методов в класс и даже в конкретный экземпляр во время выполнения программы.

Не поддерживает множественное наследование, но вместо него может использоваться концепция «примесей», основанная в данном языке на механизме модулей.

Содержит автоматический сборщик мусора. Он работает для всех объектов Ruby, в том числе для внешних библиотек.

Создавать расширения для Ruby на Си очень просто частично из-за сборщика мусора, частично из-за несложного и удобного API.

Поддерживает замыкания с полной привязкой к переменным.

Поддерживает блоки кода (код заключается в { … } или do … end). Блоки могут использоваться в методах или преобразовываться в замыкания.

Целые переменные в Ruby автоматически конвертируются между типами Fixnum (32-разрядные) и Bignum (больше 32 разрядов) в зависимости от их значения, что позволяет производить целочисленные математические расчёты со сколь угодно большой точностью.

Не требует предварительного объявления переменных, но для интерпретатора желательно, чтобы переменным присваивалось пустое значение nil (тогда интерпретатор знает, что идентификатор обозначает переменную, а не имя метода).

Язык использует простые соглашения для обозначения области видимости. Пример: просто var — локальная переменная, @var — переменная экземпляра (член или поле объекта класса), @@var — переменная класса, $var — глобальная переменная.

В Ruby непосредственно в языке реализованы многие шаблоны проектирования, так, например, «одиночка» (singleton) может быть (хотя и не обязан) реализован добавлением необходимых методов к одному конкретному объекту (см. ниже).

Может динамически загружать расширения, если это позволяет операционная система.

Имеет независимую от ОС поддержку невытесняющей многопоточности.

Перенесён на множество платформ. Он разрабатывался на Linux, но работает на многих версиях Unix, DOS, Microsoft Windows (в частности, Win32), Mac OS, BeOS, OS/2 и т. д.

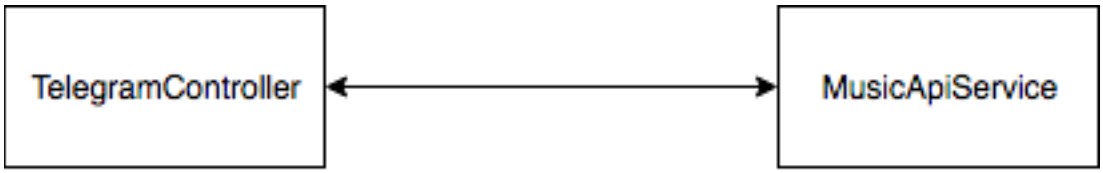

Рис. 1. Взаимодействие контроллера и сервиса

2.2. Ruby on Rails

Ruby on Rails (RoR) — фреймворк, написанный на языке программирования Ruby, реализует архитектурный шаблон Model-View-Controller для веб-приложений, а также обеспечивает их интеграцию с веб-сервером и сервером баз данных. Является открытым программным обеспечением и распространяется под лицензией MIT.

Создан Давидом Хейнемейером Ханссоном на основе его работы в компании 37signals над средством управления проектами Basecamp и выпущен в июле 2004 года. 23 декабря 2008 года команда проекта Merb объединилась с командой Rails с целью создания следующей версии Rails 3, которая объединит в себе лучшие черты обоих фреймворков.

Базируется на следующих принципах разработки приложений:

- максимальное использование механизмов повторного использования, позволяющих минимизировать дублирование кода в приложениях (принцип Don’t repeat yourself);
- по умолчанию используются соглашения по конфигурации, типичные для большинства приложений (принцип Convention over configuration) — явная спецификация конфигурации требуется только в нестандартных случаях.

Основными компонентами приложений на Ruby on Rails являются модель (англ. model), представление (англ. view) и контроллер (англ. controller). Ruby on Rails использует REST-стиль построения веб-приложений.

Модель предоставляет остальным компонентам приложения объектно-ориентированное отображение данных (таких как каталог продуктов или список заказов). Объекты модели могут осуществлять загрузку и сохранение данных в реляционной базе данных, а также реализуют бизнес-логику.

Для хранения объектов модели в реляционной СУБД по умолчанию в Rails 3 использована библиотека ActiveRecord. Конкурирующий аналог — DataMapper. Существуют плагины для работы с нереляционными базами данных, например Mongoid для работы с MongoDB.

Представление создаёт пользовательский интерфейс с использованием полученных от контроллера данных. Представление также передает запросы пользователя на манипуляцию данными в контроллер (как правило, представление не изменяет непосредственно модель).

В Ruby on Rails представление описывается при помощи шаблонов ERB — файлов HTML с дополнительными включениями фрагментов кода Ruby (Embedded Ruby или ERb). Вывод, сгенерированный встроенным кодом Ruby, включается в текст шаблона, после чего получившаяся страница HTML возвращается пользователю. Кроме ERB возможно использовать ещё около 20 шаблонизаторов, в том числе Haml.

Контроллер в Rails — это набор логики, запускаемой после получения HTTP-запроса сервером. Контроллер отвечает за вызов методов модели и запускает формирование представления.

Соответствие интернет-адреса с действием контроллера (маршрут) задается в файле config/routes.rb.

Контроллером в Ruby on Rails является класс, наследованный от ActionController::Base. Открытые методы контроллера являются так называемыми действиями (actions). Действия часто соответствует отдельному представлению. Например, по запросу пользователя admin/index будет вызван метод index класса AdminController и затем использовано представление index.html.erb из директории views/admin.

2.3. Telegram

Telegram — кроссплатформенный мессенджер, позволяющий обмениваться сообщениями и медиафайлами многих форматов. Используются проприетарная серверная часть с закрытым кодом, работающая на мощностях нескольких компаний США и Германии, финансируемых Павлом Дуровым в объёме порядка 13 млн долларов США ежегодно, и несколько клиентов с открытым исходным кодом, в том числе под лицензией GNU GPL.

Количество ежемесячных активных пользователей сервиса по состоянию на конец марта 2018 года составляет более 200 млн человек. В августе 2017 года в своем Telegram-канале Павел Дуров сообщил, что количество пользователей увеличивается на более чем 600 тысяч ежедневно.

По данным исследовательского холдинга Romir на февраль 2018 года в среднем пользователи Telegram в России тратят на него 10-11 минут в день. Самая большая доля пользователей приходится на россиян в возрасте 18-24 лет. В Москве Telegram в два раза популярнее, чем в России в целом, особенно среди аудитории от 35 до 44 лет.

Помимо стандартного обмена сообщениями в диалогах и группах, в мессенджере можно хранить неограниченное количество файлов, вести каналы (микроблоги), создавать и использовать ботов.

2.4. API как средство интеграции приложений

API определяет функциональность, которую предоставляет программа (модуль, библиотека), при этом API позволяет абстрагироваться от того, как именно эта функциональность реализована.

Если программу (модуль, библиотеку) рассматривать как чёрный ящик, то API — это множество «ручек», которые доступны пользователю данного ящика и которые он может вертеть и дёргать.

Программные компоненты взаимодействуют друг с другом посредством API. При этом обычно компоненты образуют иерархию — высокоуровневые компоненты используют API низкоуровневых компонентов, а те, в свою очередь, используют API ещё более низкоуровневых компонентов.

По такому принципу построены протоколы передачи данных по Интернет. Стандартный стек протоколов (сетевая модель OSI) содержит 7 уровней (от физического уровня передачи бит до уровня протоколов приложений, подобных протоколам HTTP и IMAP). Каждый уровень пользуется функциональностью предыдущего («нижележащего») уровня передачи данных и, в свою очередь, предоставляет нужную функциональность следующему («вышележащему») уровню.

Важно заметить, что понятие протокола близко по смыслу к понятию API. И то, и другое является абстракцией функциональности, только в первом случае речь идёт о передаче данных, а во втором — о взаимодействии приложений.

API библиотеки функций и классов включает в себя описание сигнатур и семантики функций.

3 РАЗРАБОТКА ПРИЛОЖЕНИЯ

3.1 Архитектура приложения

Приложение построено на базе архитектуры Ruby on Rails, которая в свою очередь строится на паттерне MVC - Model, View, Controller. MVC архитектура имеет ряд преимуществ, главным из которых является возможность разделять ответственность между разными частями приложения.

В данной релизация отсутствуют привычные для Rails-приложений модели, однако присутствует слой сервисов, который отвечает за обработку данных.

Среди преимуществ Ruby и Ruby on Rails можно выделить богатый набор всевозможных библиотек, помогающих решать различные задачи. Одна из библиотек была использована в данном проекте. Библиотека устроена таким образом, что реализует базовый функционал по работе с Telegram Bot API, являясь своеобразной обёрткой для этого программного интерфейса. Задача разработчика в данном случае состоит в том, чтобы описать логику взаимодействия пользователя и бота.

Стоит сказать, что среди библиотек, которые реализуют обёртки для Telegram Bot API, существуют различные подходы к реализации и построению приложений. Некоторые предлагают достаточно прямой и незамысловатый подход, который заключается в написании больших веток с условиями, другие же, как, например, библиотека, telegram-bot, предлагает модульное решение, с разделениям по типам респонсов и дополнительными возможностями.

Для непосредственного получения информации об артистах был использован сервис Last.fm, который предоставляет API для разработчиков. В данном проекте реализована собственная обёртка с помощью библиотеки HTTParty для конструирования HTTP запросов.

3.2 TelegramController

TelegramController является управляющей частью приложения и наследуется от Telegram::Bot::UpdatesController - класса, который является частью библиотеки telegram-bot, и методы которого переопределяются в дочернем контроллере. Данный контроллер отвечает за обработку взаимодействия с пользователем, а именно обрабатывает команды пользователя, получает необходимые данные, передаёт данные для обработки в MusicApi сервис, и, получив данные для пользователя и подготовив их, возвращает.

Управление сессией

В отличие от других библиотек, работающих с Telegram Bot API, несомненным преимуществом библиотеки telegram-bot является наличие возможности работы с сессиями пользователей. Это необходимо для того, чтобы сохранять состояние пользователя и на основе этого состояния делать определённые действия. Например, в данном приложении сессия используется для сохранения выбранного пользователем артиста, а именно его имени и уникального музыкального идентификатора. Таким образом, с использованием сессия в этом приложении есть возможность сохранять контекст и использовать его в последующих telegram-командах.

Управляющие команды

Управляющими командами можно назвать команды, которые пользователь может послать через телеграм бота и они будут влиять на дальнейшее развитие взаимодействия. К таким командам относятся, например, ‘/start’, ‘/help’,’/tracks’ и так далее. Эти команды начинаются с символа ‘/‘ и этим отличаются от других. Для TelegramController вышеописанные команды являются знаком того, что нужно создать объект контроллера и запустить соответствующий метод объекта контроллера. Другие же попытки пользовательского ввода будут обработаны методом ‘message’.

Респонс

TelegramBot оснащён функциональностью, которая позволяет кастомизировать респонс, который будет отправлен пользователю. В данном приложении есть два основных вида респонса: текст и текст с отображением клавиатуры. Клавиатура отображается для удобства пользователя и скрывается после её использования. Однако, всё, что делает клавиатура - это эмуляция пользовательского ввода, поэтому она с лёгкостью может быть заменена правильно введённой управляющей командой.

Методы взаимодействия с сервисом

В TelegramController так же присутствуют методы, которые передают пользовательский запрос в соответствующий метод MusicApi сервис. Таким образом, эти методы являются своего рода диспетчерами между пользовательским вводом и сервисом, который получает актуальную информацию.

3.2 MusicApi Service

Данный сервис используется для получения информации об артистах, из композициях, альбомах, а также поиска похожих исполнителей. Для получения необходимой функциональности был выбран сервис Last.fm, который предоставляет API. Для того, чтобы пользоваться API необходимо получить токен, который должен подставляться в каждый вопрос для аутентификации и контроля ограничения количества запросов. Данный API работает по HTTP протоколу.

HTTParty

Данная библиотека используется для реализация взаимодействия с Last.fm API по HTTP протоколу на языке руби. Для работы библиотеки необходимо сделать подключение её в соответствующий сервис (класс), в данном случае MusicApi, и прописать base_url, который будет использоваться как начало URL для всех запросов.

В конкретных запросах необходимо выбрать HTTP-метод и вызвать соответствующий метод на объекте класса. Аргументами данного метода будут служить path для URL, а также необходимые опции, среди которых всегда будет токен, необходимый для авторизации.

Методы для информации об артистах

В приложении используются методы для получения информации об артистах: поиск артистов, альбомов, композиций и т.д. Данные методы выполняют две функции: выполнение HTTP-запроса и обработка полученной информации. Для удобство использования API был выбран тип данных JSON. Зачастую в качестве ограничения количества берётся некоторое число, чтобы пользователю бота было удобно просматривать полученную информацию.

4 ВЫВОД

В процессе работы над приложением были изучены библиотеки для работы с HTTP протоколом, Telegram Bot API и было создано приложение, которое представляет собой телеграм-бот для получения информации о музыкальных артистах. Для реализации приложения были выбраны технологии, которые позволяют быстро и удобно строить процесс разработки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Ruby [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ru.wikipedia.org/wiki/Ruby

[2] Ruby on Rails [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ru.wikipedia.org/wiki/Ruby_on_Rails.

[3] Application Programming Interface [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ru.wikipedia.org/wiki/API.

[4] Telegram [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ru.wikipedia.org/wiki/Telegram.