|  |  |  |  |
| --- | --- | --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  ФГБОУ ВО «Пермский государственный национальный  исследовательский университет» | | | |
|  | | | |
| ОТЧЕТ  по лабораторной работе №1  «Создание клиентского приложения по выполнению CRUD-операций на популярном информационном ресурсе»  по дисциплине «Технологии разработки распределенных приложений» | | | |
| Работу выполнили студенты группы ФИТ-4-НБ 4 курса:  Бродникова Вадима Сергеевича  «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2021 г. |  | | Проверил к.ф.-м.н., доцент кафедры ПМИ  Деменев Алексей Геннадьевич  «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2021 г. |
|  | |  | |
| Пермь 2021 | | | |

1. Постановка задачи

***Цель***: Изучение возможностей использования API современных популярных информационных ресурсов внешними приложениями. ***Формируемая компетенция:*** способность применять на практике теоретические основы и общие принципы разработки распределенных систем.

***Организация выполнения работы:*** Каждый студент выполняет индивидуальное задание.

***Требования к выполнению работы:***

1. Разработанное приложение должно предоставлять возможности создания, чтения, обновления и удаления некоторых объектов выбранного информационного ресурса (выполнения CRUD-операций).
2. CRUD-операции должны выполняться при помощи API выбранного ресурса, причём по крайней мере один тип операции должен требовать аутентификации на нём.
3. Должна быть возможность выполнения по крайней мере двух CRUD-операций для одного и того же объекта выбранного информационного ресурса.

По окончании выполнения задания каждый студент должен подготовить отчет.

Отчет по выполнению задания должен включать:

1. Общее описание приложения. Постановка задачи, введение в предметную область.
2. Архитектура системы. Обоснование выбора данного типа архитектуры распределенного приложения. Алгоритм работы приложения в целом.
3. Архитектура каждого из логических компонент системы (серверы, клиенты). Подходы к реализации. Алгоритмы работы.
4. Методы коммуникаций компонентов системы (клиент→сервер, сервер→клиент и т.д.).
5. Структура передаваемых данных.
6. Исходный код.

Распределение баллов за выполнение работы представлено в следующей таблице:

|  |  |
| --- | --- |
| **Критерий оценивания** | **Оценка** |
| Приложение позволяет выполнять CRUD-операции над объектами выбранного ресурса | 1  (за каждый тип операции) |
| Приложение позволяет выполнять по крайней мере две CRUD-операции для одного и того же объекта выбранного ресурса | 1 |
| Приложение выполняет по крайней мере одну из CRUD-операций после успешной аутентификации | 2 |
| Приложение не требует повторной аутентификации при перезапуске программы | 1 |
| Параметры успешной аутентификации сохраняются при перезапуске программы в зашифрованном виде | 2 |

1. Описание приложения. Введение в предметную область

Предметная область – работа с текстовыми постами (записями) на стене определенного сообщества на ВК.

Для людей VK отличная русскоязычная (и не только) социальная сеть, которую можно использовать для общения с другими пользователями, а для разработчиков - сервис, позволяющий внедрять на свои сайты код программы с подсветкой разметки кода. Предположим, вам дали задание нарисовать на канве шахматную доску, вы написали такой код. Дальше заходите на gist.github.com, размещаете там ваш код. Код удобно скачивать на ваш компьютер. У вас есть возможность получать комментарии к нему. Ваш код могут оценивать, дорабатывать. Вы можете отправить ссылку вашему преподавателю.

Но, удобнее всего то, что вы можете получить код скрипта, для того чтобы разместить код вашей программы где угодно. Например, на странице вашего блога. И когда вы будете править ваш код, то везде, где он размещен, он будет меняться до актуальной версии.

Само приложение имеет простой интерфейс и удобство использования, без лишних деталей.

1. Лицензия

Приложение разработано на языке программирования Python и распространяется с лицензией Python Software Foundation License (URL: <https://docs.python.org/3/license.html> (дата обращения: 14.02.21)). Сервер вк написан на языке программирования KPHP (kPHP, KittenPHP), основанный на языке PHP и распространяется с лицензией (URL: https://github.com/VKCOM/kphp/blob/master/LICENSE (дата обращения: 14.02.21)).

1. Архитектура системы

На рисунке 1 изображена архитектура системы.

User

Client

Resource server

Authorization server

Рисунок 1 – Архитектура системы

**Алгоритм работы системы следующий:**

1. Пользователь использует адрес http-запрос вида –https://api.vk.com/method/METHOD\_NAME?PARAMETERS&access\_token=ACCESS\_TOKEN&v=V для использования технологии API Вконтакте. И с помощью этих запросов клиент реализует создание, удаление, обновление или получение ресурса при помощи CRUD-операций.
2. Все операции доступны, при использовании уникального токена, который был заранее получен. Тем самым и осуществляется авторизация по протоколу OAuth 2.0.
3. В приложении используется интерфейс командной строки(CLI). Чтобы создать, удалить, обновить, получить ресурс нужно соответственно в программе прописать create, delete, edit, read.
4. Операция «create». Программа запрашивает сообщение, который пользователь хочет создать: «Пожалуйста, введите новое сообщение». Подтверждением операции «create» будет служить сообщение «Запись на стене успешно была добавлена».
5. Операция «delete». Программа запрашивает id сообщения, который пользователь хочет удалить: «Пожалуйста, введите существующий id сообщения, который хотите удалить». Если id существует, операция осуществится. Подтверждением операции «delete» будет служить сообщение: «Запись на стене успешно была удалена, если вы ввели правильный id».
6. Операция «edit». Программа запрашивает id сообщения, который пользователь хочет изменить: «Пожалуйста, введите существующий id сообщения, который собирайтесь изменить». Затем программа запрашивает, какое новое сообщение хотите записать: «Пожалуйста, введите новое измененное сообщение». Если id существует, операция осуществится. Подтверждением операции «edit» будет служить сообщение: «Запись на стене успешно была изменена, если вы ввели правильный id».
7. Операция «read». Программа показывает все записи стены, а именно их текстовое содержание и id сообщения в таком формате: «id поста номер id,Текст сообщения = содержание сообщения»
8. Для выхода из приложения, необходимо написать программе exit.
9. Приложение позволяет выполнять 4 CRUD-операции для одного и того же объекта, а именно записи на стене конкретной группы в ВК.
10. Архитектура клиента

Клиент представляет собой CLI-приложение. Архитектура клиента изображена на рисунке 2.

Задачи клиента – организовать правильную работу CRUD-операций.

Client

Authorization server

Resource server

Коммуникационный модуль

Рисунок 2 – Архитектура клиента

6 Архитектура сервера

Сервер представляет собой консольное приложение, архитектура сервера изображена на рисунке 3.

Задача сервера: обеспечивать корректность выполнения OAuth 2.0 авторизации.

Client

Server

Resource server (VK API)

Рисунок 3 – Архитектура сервера

1. Методы коммуникаций компонентов системы

Для все коммуникации в системе используется синхронное взаимодействие по протоколу HTTP. Передаваемый формат данных - JSON

1. Структура передаваемых данных

Формат данных описан в документации VK API 5.130 (https://github.com/VKCOM/vk-api-schema)

9 Исходный код

Реализация приложена рядом с отчетом.

Проект - <https://github.com/vadikTheBest/vki_api_python>