|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА — Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт информационных технологий

Кафедра математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе №12**

по дисциплине

«Разработка мобильных приложений»

|  |  |
| --- | --- |
| **Выполнил:**  Студент группыИКБО-33-21 | Привезенцев Д.А. |
| **Проверил:**  Ассистент | Новичков Д.Е. |

Москва 2023 г.

**СОДЕРЖАНИЕ**

[1. ЗАДАНИЕ 3](#_Toc136827218)

[1.1 Условие задания 3](#_Toc136827219)

[1.2 Решение задания 3](#_Toc136827220)

[ВЫВОД 7](#_Toc136827221)

# ЗАДАНИЕ

## Условие задания

Реализовать, как минимум три запроса данных с сайта согласно предметной области. Запросы должны включать как минимум один GET и один POST.

## Решение задания

Для решения данной задачи были добавлены три кнопки на главный экран приложения, каждая из которых выполняет определенный запрос данных с веб-сайта.

Первая кнопка выполняет GET-запрос с указанием конкретного идентификатора поста и отображает полученные данные в текстовом поле. В данном случае был использован метод getPostWithID(id), который возвращает данные конкретного поста с указанным идентификатором.

Вторая кнопка выполняет GET-запрос и получает список всех постов. Полученные данные отображаются в текстовом поле, где перечислены идентификаторы всех постов. Для этого был использован метод getAllPosts(), который возвращает список всех постов.

Третья кнопка выполняет POST-запрос с использованием данных, полученных из первого запроса. Для этого был использован метод postData(data), который принимает объект поста в качестве параметра и отправляет его на сервер. После успешной отправки данных, полученный пост отображается в текстовом поле.

Для выполнения запросов к серверу был использован интерфейс JSONPlaceHolderApi, который определяет методы для каждого запроса. Для отправки и получения данных использовалась библиотека Retrofit, которая позволяет упростить работу с сетью.

Таким образом, реализация трех запросов данных с использованием GET и POST методов была успешно выполнена. Кнопки на главном экране приложения позволяют выполнять запросы и отображать полученные данные. Ниже на Рисунках 1-2 представлен код основных модулей приложения, отвечающих за экран с кнопками и за обработку JSON. Итоговый вид приложения представлен на Рисунке 3.

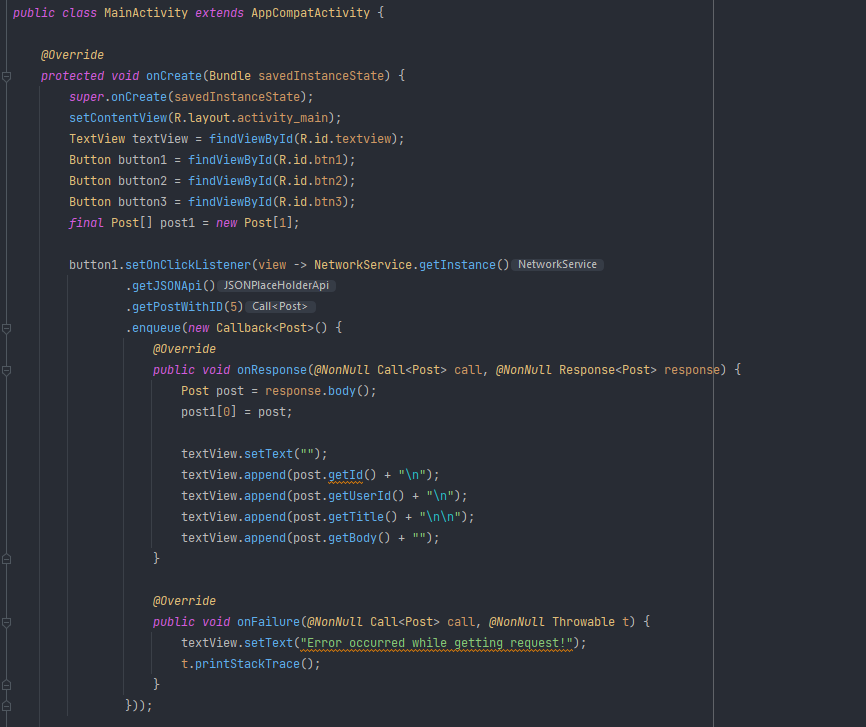


Рисунок 1 — Код навигации в модуле MainActivity

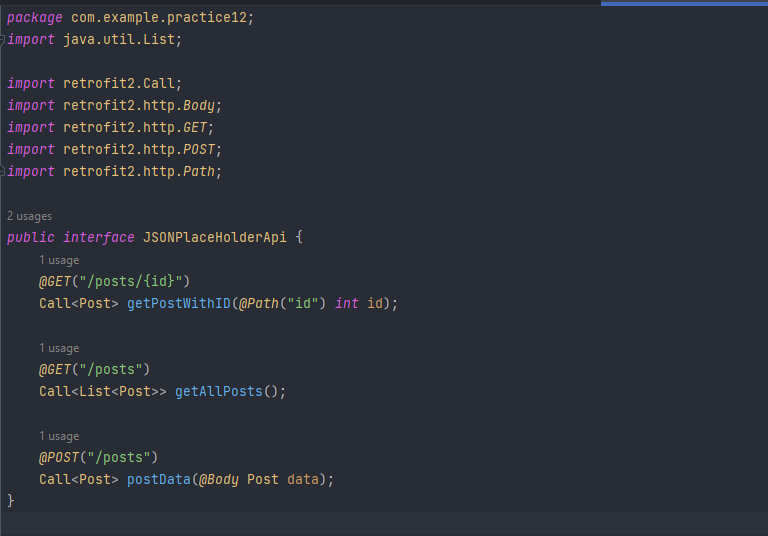


Рисунок 2 — Код в модуле JSONPlaceHolderApi

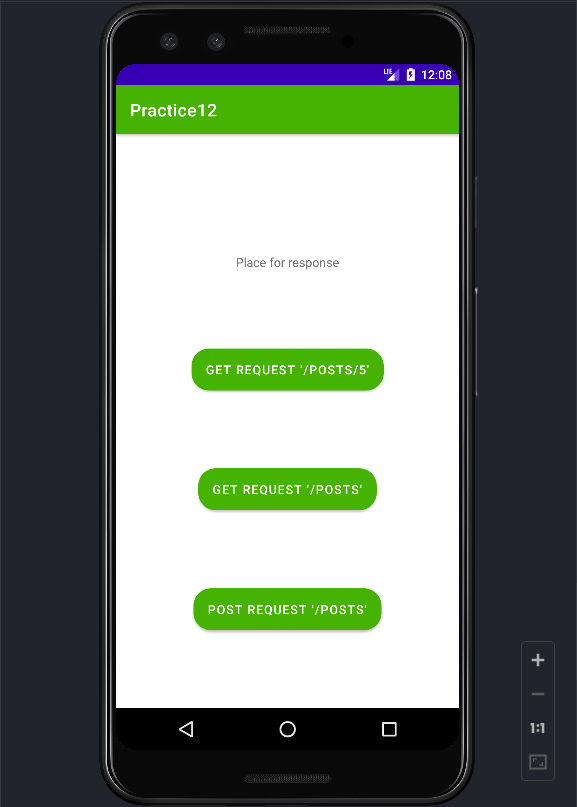


Рисунок 3 — Вид приложения

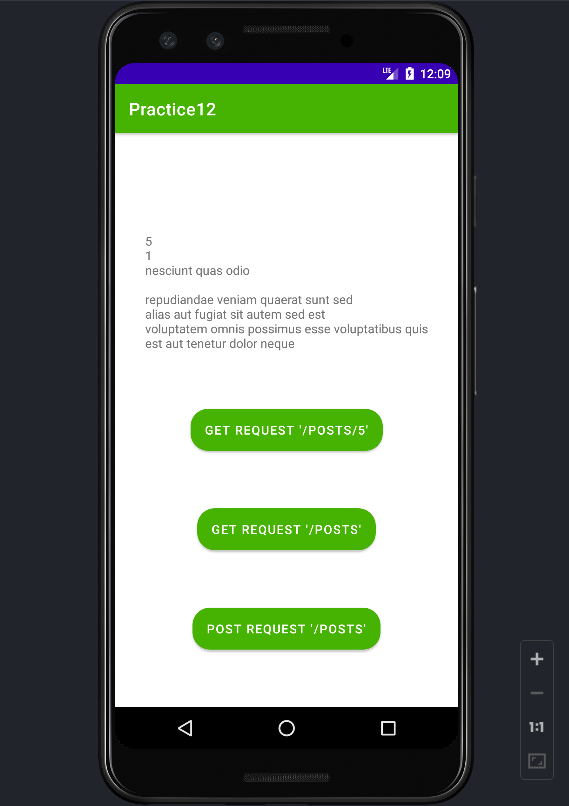


Рисунок 4 — Вид приложения с нажатой кнопкой GET REQUEST

# ВЫВОД

Таким образом, реализация трех запросов данных с использованием GET и POST методов была успешно выполнена. Кнопки на главном экране приложения позволяют выполнять запросы и отображать полученные данные.

Добавление кнопок и реализация запросов данных с веб-сайта позволяют пользователю взаимодействовать с приложением и получать информацию, которая ему необходима. GET-запросы позволяют получить данные по определенным критериям, а POST-запросы позволяют отправлять данные на сервер для их обработки.

Использование Retrofit упрощает работу с сетью и облегчает выполнение запросов, так как позволяет автоматически преобразовывать полученные данные в объекты Java.

Таким образом, выполнены все требования задания, реализованы запросы данных с веб-сайта с использованием GET и POST методов, и разработан интерфейс пользователя для взаимодействия с приложением.