МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

Кафедра «Информационные системы»

Дисциплина «Программирование мобильных устройств»

Лабораторная работа №5

Взаимодействие с REST API.

Выполнил:

студент гр. ПИбд-31

Распаев Н.И.

Проверил:

доценткафедры ИС

Филиппов А.А.

Ульяновск, 2023 г

## Задание:

1. Создать сервис для работы с REST API с помощью Retrofit. В качестве сервера можно использовать json-server или самостоятельную реализацию.
2. Создать репозитории для абстрагирования от реализации механизма доступа к данным. Новые репозитории должны реализовывать интерфейсы-репозитории из ЛР 4, а также использовать классы-репозитории для работы с БД.
3. Для работы со страничными данными (paging) необходимо реализовать RemoteMediator в соответствующем новом репозитории.
4. Адаптировать ViewModel для работы с новыми репозиториями. Бизнеслогика должна быть изменена минимально.
5. Отчет и изменения проекта загрузить в репозиторий по адресу <http://student.git.athene.tech> В ходе выполнения работы использовался json-server. Для хранения данных использовался json файл «data.json», который содержал в себе следующие таблицы:
6. «orders»
7. «order\_with\_products»
8. «user\_with\_products»
9. «categories»
10. «users»
11. «products»

Был реализован класс MyServerService и статические методы для его использования:

companion object {

private const val BASE\_URL = "http://10.0.2.2:8079/"

@Volatile

private var INSTANCE: MyServerService? = null

fun getInstance(): MyServerService {

return INSTANCE ?: synchronized(this) {

val json = Json {

ignoreUnknownKeys = true

}

val logger = HttpLoggingInterceptor()

logger.level = HttpLoggingInterceptor.Level.BASIC

val client = OkHttpClient.Builder()

.addInterceptor(logger)

.build()

return Retrofit.Builder()

.baseUrl(BASE\_URL)

.client(client)

.addConverterFactory(json.asConverterFactory("application/json".toMediaType()))

.build()

.create(MyServerService::class.java)

.also { INSTANCE = it }

}

}

}

Для конвертации данных были созданы модели, к котором в конце было дописано «Remote». В дальнейшем были реализованы Репозитории, которые работают с API json-server.

Код можно посмотреть по ссылкам:

StudentGit: <http://student.git.athene.tech/Nikolay-Raspaev/PMU/src/branch/master>

GitHub: <https://github.com/keyssen/mobile_app/tree/master>