Министерство образования республики Молдова

Технический Университет Молдовы

Департамент Программной Инженерии и Автоматики

Отчёт

Индивидуальня работа

По предмету: Programarea aplicațiilor mobile

Тема: Кэширование данных

Выполнил студ. гр. TI-207 Бордя В.

Проверил Лях А.

Кишинёв – 2022

**Varianta 3:**

1. Încărcarea datelor se vor face din REST-API utilizind protocolul http.
2. Necesar este sa folosim paginare infinita la preluarea feed-ului de date.
3. Implementare o noua pagina UI - detalii
4. Adaptare proiect existent la Clean Architecture - separare pe cele 3 layer-e: Domain, Data, Presentation
5. Кэширование данных

News app

Rest-API e aici:

<https://news-app-api.k8s.devebs.net/>

Design preluati de aici: <https://www.figma.com/file/vR3YnmX5hNTElDSnMqi5aD/PAM---Laborator-4?node-id=0%3A1>

**Используемая технология:**

**- Flutter**

**Ход работы:**

Структура проекта поменялась от прошлой на Clean Architecrture: Domain(Rest запросы с превращение dto в model), Data(модели обьектов, используемых в приложеии), Presentation(UI, на котором показываются данные, согласно дизайну). Структура проекта представлена на рискунке 1.

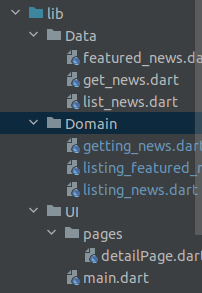


Рисунок 1 – Структура проекта

Рассмотрим 3 модели, необходимые в данном приложение:

1. featured\_news
2. list\_news
3. get\_news

Первая модель данных необходима для carousel. В ней необходимые данные – линк картинки, заголовок и id(необходимые для ридеректа при клики на read now). Данный класс представлен на рисунке 2.

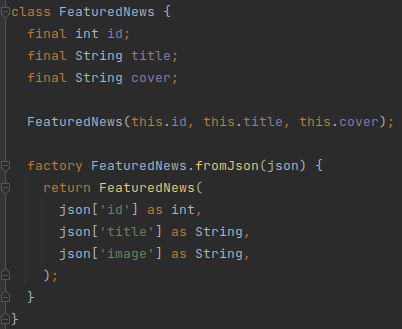


Рисунок 2 – Model FeaturedNews

Вторая модель данных необходима для ListView. В ней необходимые данные для отображения согласно дизайну и id(необходимые для ридеректа при клики на read now). Данный класс представлен на рисунке 3.



Рисунок 3 – Model ListNews

Третья модель данных необходима для News DetailPage. В ней необходимые данные для отображения согласно дизайну. Данный класс представлен на рисунке 3.



Рисунок 4 – Model GetNews

Следующее, что стоит расмотреть это BuisnessLogic, которые достают данные из Rest API, используя http package и преобразуют их в models, описанные выше:

1. listing\_featured\_news
2. listing\_news
3. getting\_news

Каждая BL имеет две функции:

1. Получает данные из API
2. Парсит данные в необходмые для Presentation Layer данные

Каждый request проходит проверку статуса и в случае, если он отличается от 200(успешно), будет выведена ошибка.

Первая BL достает все Featured News используя query string – is\_featrued=true в запросе.

Данный класс представлен на рисунке 5.



Рисунок 5 – BL ListingFeaturedNews

Вторая BL принимает два параметра в зависимости, от которых получает определенное кол-во данных, а имеено page(номер страницы) и newsPerPage(кол-во entry на странице).

Данные параметры необходимы, чтоб не подргужать каждый раз уже подргуженные данные, а только новые с дальнейшем добавление их в уже подргуженным.

Данный класс представлен на рисунке 6.

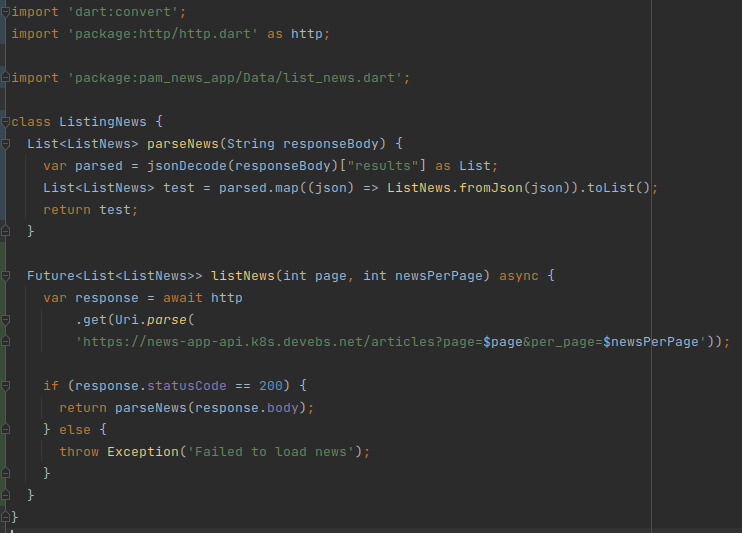


Рисунок 6 – BL ListingNews

Последняя BL принимает один параметр – id, который используется как параметр в path, чтобы достать неоходимый нам news. Данный класс представлен на рисунке 7.

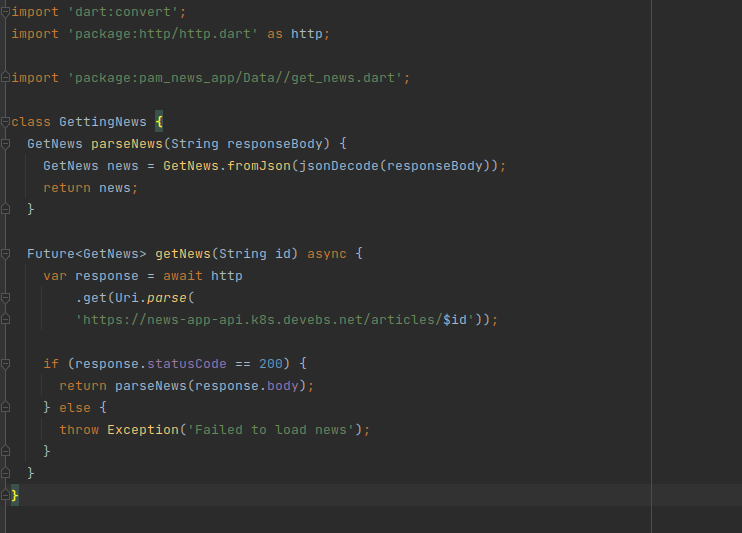


Рисунок 7 – BL GettingNews

В HomePage были добавленно несколько state, необходимых для подгрузки данных при скроле:

1. \_loadingMore(bool значение, показывающее грузиятся ли уже данные)
2. \_hasMoreItems(bool значение, показывающее есть ли еще данные, которые можно получить с API)
3. \_currentPage(int значение, показывающее текущую старницу, необходимо для ListingNews - 1 параметр)
4. \_newsPerPage(const значение, показывающее кол-во news, загружаемых за один request)

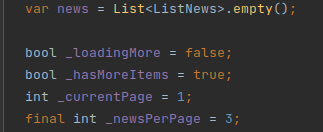


Рисунок 8 – Переменные HomePage

Также был добавлен ScrollController, за счет которого мы будем знать, текущее положение в ListView и тем самым при достижении конца страницы – подгружать новую, в случае если уже не происходит request и есть еще не подргруженные данные в API. Имплементация представлена на рисунке 9.

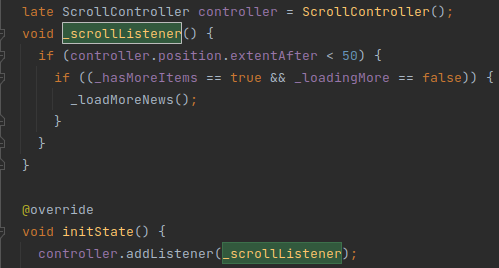


Рисунок 9 – Имплементация ScrollController

Подгрузка новых news реализована с помощью функции \_loadMoreNews, которая в случае, если уже не запущена(flag \_loadginMore), подгружает данные, и добавляет их в news. Далее идут два случая: 1 – если данных пришло столько, сколько мы рсчитывали – flag \_loadingMore ставим обратно в false и инкрементирем currentPage, 2 – данных пришло меньше – значит данных в API больше нет => ставим flag \_hasMoreItems false. Код функции представлен на рисунке 10

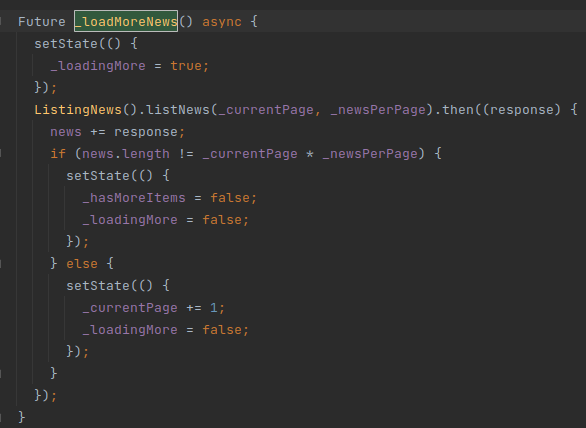


Рисунок10 – Имплементация функции \_loadMoreNews

При запуски приложения загружаются первые news при помощи функции \_getNews. На рисунке 11 представлен код.

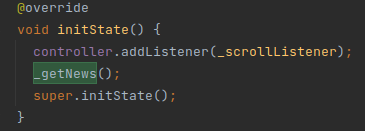
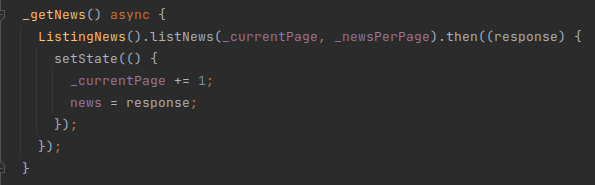


Рисунок11 – Имплементация подрузки первых news при запуске

Подгрузка данных для FeaturedNews в CarouselSlider просходит за счет FutureBuilder. Имплементация представлена на рисунке 12.

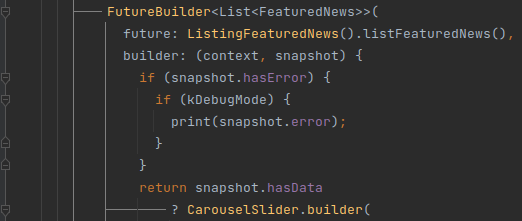


Рисунок 12 – имплементация подгрузки данных для carousel

То, как выглядит HomePage на данный момент представлен на рисунке 13.

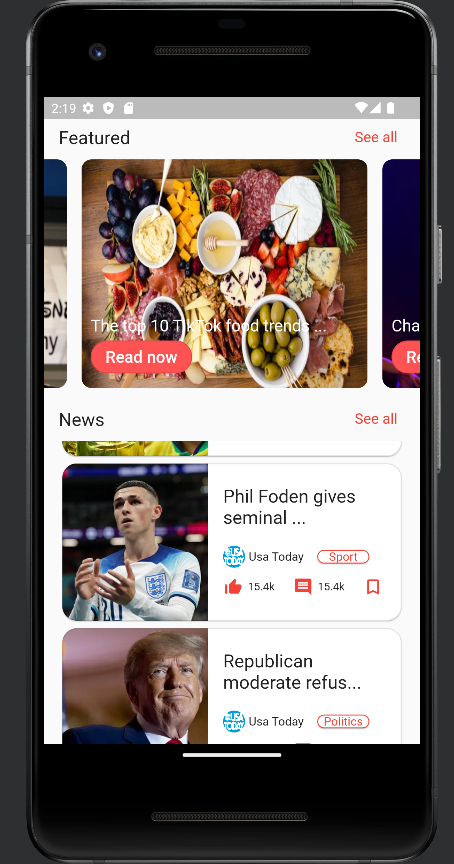


Рисунок 13 – Визуализация HomePage

При нажатии на Read Now или на любую новость из листа, происходит редирект на DetailPage. Реализация этого представленна на рисунке 14.

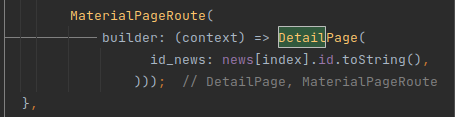


Рисунок 14 – реализцаия редиректа на DetailPage

DetailPage принимает id\_news и с помощью FutureBuilder достает GetNews.

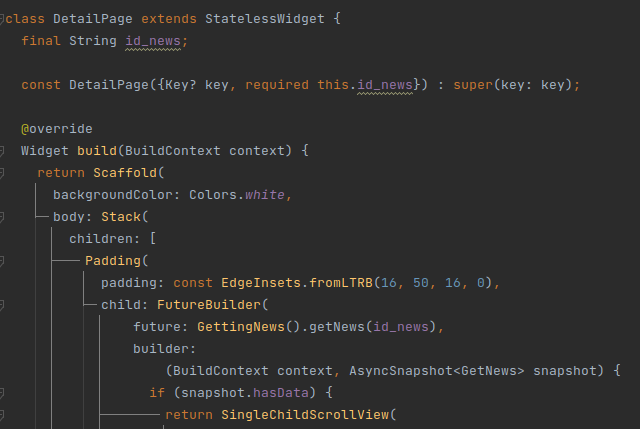


Рисунок 15 – реализация подгрузки GetNews

То, как выглядит DetailPage на данный момент представлен на рисунке 16.

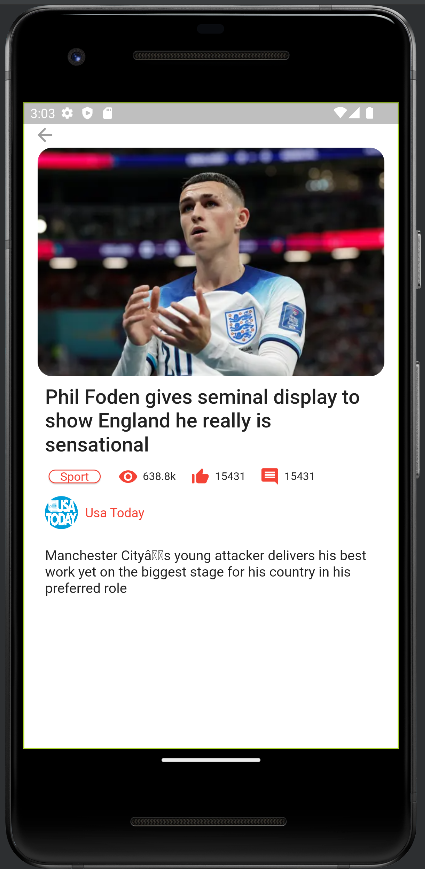


Рисунок 16 – визуализация DetailPage News

Для добавления кэша необходимо было изменить BL. Кэширование будет осуществляться за счет json файлов. Расмотрим кэширование GetNews.

BL проверяет существует ли необходимый нам файл(файл – это news + id).

В случае, если нет – осуществляется подгрузка из API и записывание данных в файл news + id. В случае, если да – осуществляется подгрузка данных из json файла. Код представлен на рисунке 17.



Рисунок 17 – Релазиция кэширования для GetNews

Для иллюстарции работы кэширования откроем дважды один и тот же news. В первый раз news подгружается из API, во второй – из cache(json файла). Рисунок log-ов иллюстрирующих работу кэширования представлено на рисунке 18.

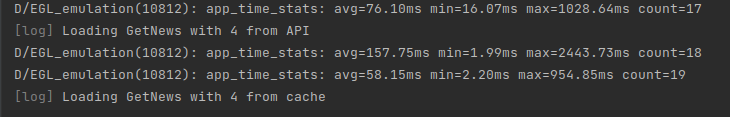


Рисунок 18 – Иллюстрация кэширования для GetNews

Кэширования для FeaturedNews реализовано таким же образом, только файл в данном случае один – featured\_news.json. Код представлен на рисунке 19.



Рисунок 19 – Иллюстрация кэширования для FeaturedNews

Кэширования для News немного иначе, так как существует подгрузка при скроле. Файл у нас один – news.json, только мы не можем быть уверены, что в данном файле будут все необходимые нам данные, поэтому существует проверка кол-ва данных из файла и в случае, если их недостаточно – то мы подгружаем их из API. Код представлен на рисунке 20.

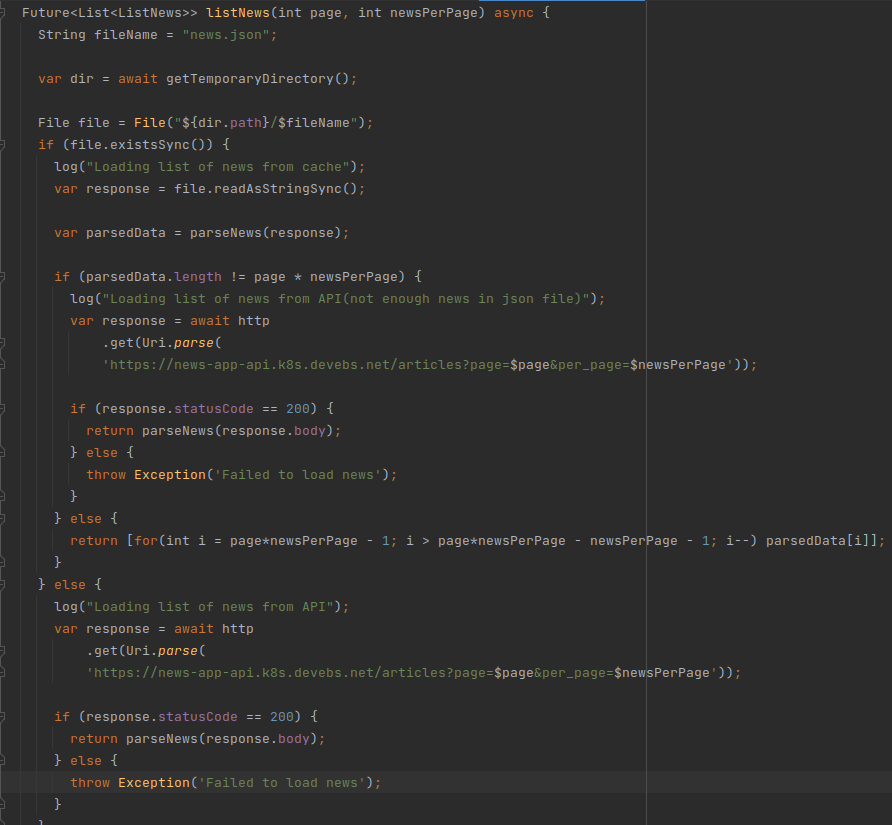


Рисунок 20 – Иллюстрация кэширования для ListNews

Код приложения представлен на моем GitHub - https://github.com/forg002-ctrl/PAM\_news\_app

Вывод: в ходе данной лабораторной работы было дополненно приложение news и реализовано кэширование за счет json файлов.