### Документація проекту Top News APP

Top News APP є додатком для мобільних пристроїв, який надає користувачам доступ до останніх новин через інтерфейс Flutter. Цей додаток використовує фреймворк Flutter для створення користувацького інтерфейсу та взаємодії з серверною частиною, яка базується на newsapi.ai.

### Структура проекту

Проект поділен на декілька рівнів згідно з принципами Clean Architecture:

* core: Містить загальний код, який може використовуватися в усіх рівнях додатку.
* domain: Містить бізнес-логіку додатку, абстраговану від будь-якої конкретної імплементації.
* data: Відповідає за отримання та збереження даних, включаючи взаємодію з зовнішніми джерелами, такими як API.
* presentation: Містить код, який відповідає за представлення даних користувачу, включаючи UI компоненти та логіку взаємодії з ними.
* shared: Містить загальні ресурси, які можуть бути використані у різних частинах додатку.
* start: Містить основний код, який ініціалізує додаток.

### Використані пакети

Додаток також використовує наступні пакетів для покращення функціональності та продуктивності:

* freezed, freezed\_annotation: Генерація коду для імутабельних (immutable) об'єктів та зразків (pattern matching).
* dio, retrofit: Бібліотеки для роботи з HTTP запитами.
* flutter\_bloc: Для управління станом додатку за допомогою BLoC (Business Logic Component).
* flutter\_expanded\_tile, flutter\_adaptive\_scaffold: Допоміжні компоненти для роботи з UI.
* cool\_dropdown, cached\_network\_image, oktoast, shimmer, card\_loading, table\_calendar, gif: Різноманітні UI компоненти та бібліотеки для поліпшення візуального досвіду користувача.
* loggy, stack\_trace: Інструменти для обробки та виведення логів.
* equatable, dartz, get\_it, go\_router, url\_strategy: Допоміжні бібліотеки для різних аспектів розробки.

### Рівні та їх відповідальності

На рівні data маємо datasources і models:  
Клас NewsRemoteDatasource відповідає за взаємодію з віддаленим джерелом даних для отримання новин. Він визначає контракти для виконання різних HTTP-запитів до сервера, який містить дані про новини. Основна відповідальність цього класу полягає в тому, щоб забезпечити зручний інтерфейс для отримання даних, таких як недавні активні статті, популярні статті, категорії, локації, пошук за заданими параметрами.

А також відповідні моделі для формування запитів та серіалізації отриманого результату (response).

На рівні domain маємо repositories і entities:

Репозиторій NewsRemoteRepository відповідає за взаємодію з віддаленим джерелом даних та форматування цих даних у відповідні об'єкти сутностей для подальшого використання в додатку. Ось пояснення до класу та його методів:

Відповідні моделі конвертуються в сутності (entity) для зручного використання в застосунку.

На рівні presentation маємо окремі сторінки, що будують користувацький інтерфейс (UI). До кожної з сторінок прикріплений окремий Bloc (Cubit).

Bloc – це один з підходів архітектурного рішення, що відповідає за управлінням станами інтерфейсу (state management). Він працює за принципом MVI:

Основна ідея полягає в тому, що весь сторінки знаходиться в одному об'єкті Bloc. Зміни стану ініціюються подіями, які обробляються в Bloc і відображаються на UI. Bloc має два основних елементи: події (Events) та стани (States). Події вказують на зміни, які потрібно внести в стан додатку, а стани представляють різні стани, в яких може перебувати додаток. Зазвичай, для кожної сторінки в додатку створюється свій власний Bloc, який відповідає за управління станом цієї сторінки.

Cubit є спрощеною версією Bloc, яка призначена для роботи зі станами додатку без потреби використання подій. В основі роботи Cubit лежить принцип одностороннього потоку даних. Він отримує дані, обробляє їх та повертає новий стан додатку. Cubit не має окремих подій для керування змінами стану. Замість цього, стан Cubit може змінюватися прямими викликами методів.

Сторінки додатку:  
articles – список популярних/останніх новин, що відображаються на домашній сторінці

home – домашня сторінка, що містить в собі макет для відображення списку новин, пошуку, детальної інформації про новину. А також відповідає за зміну стану між списком новин та пошуком.

one\_article – сторінка детальної інформація про новину

search – сторінка пошуку, складається з 2 частин: відповідно сам пошук з фільтрами і результату у вигляді списку новин.

На рівні shared маємо constants і widgets:  
Це рівень, де зібрані константи, а також компоненти (в контексті Flutter – віджети) як окремі елементи UI.

На рівні start маємо загальні налаштування застосунки в вигляді Dependency Injection, початкові налаштування для запуску сервісів, налаштування теми, кольорів, шрифтів, навігації.

На рівні core маємо спеціальні налаштування застосунки в вигляді окремих сервісів, enums, розширення для мови Dart (extension), загальних інтерфейсів, а також допоміжних класів для спрощення коду на рівнях data, domain, presentation.

Також маємо рівень generated, який є згенерований і описує поведінку для кодогенерованих класів.

### Кросплатформеність застосунку

Для адаптивності застосунку використовується 2 підходи:

Responsive Design – адаптивні розміри компонентів в залежності від розміру екрану. Завдяки цьому, додаток може оптимально відображатися як на малих мобільних екранах, так і на великих планшетах або desktop-платформах

Flutter – як платформа, що дозволяє створювати нативні додатки з одним кодом для різних операційних систем