

Flutter & Dart Complete Guide

Prepared by Eng. Yousef Salah Kassem

DAFL 2
Practical Lecture

Lecture 4



Table Of Content

- Services
- Models
- Routes
- Future Builder



Flutter provides various ways to fetch and manage data asynchronously. One of the most common approaches is using **Future** and **FutureBuilder**.

NewsService Class Overview

The **NewsService** class is responsible for handling data retrieval for the application. It fetches news articles and categories from local JSON files and returns the data as Dart objects.



Key Features of `NewsService`:

1. **Fetching News (`getNews()`)**
 - Loads news from a JSON file (`assets/jsons/news.json`).
 - Parses the data and converts it into a list of `News` objects.
 - Introduces a **3-second delay** to simulate network latency.
2. **Fetching Categories (`getCategories()`)**
 - Loads categories from a JSON file (`assets/jsons/categories.json`).
 - Converts the data into a list of `Category` objects.
 - Also introduces a **3-second delay**.
3. **Fetching News by Category (`getNewsByCategory(categoryId)`)**
 - If `categoryId == 0`, it returns all news.
 - Otherwise, it filters the news based on the selected category.
4. **Fetching a Single News Article by ID (`getNewsById(id)`)**
 - Retrieves a specific news article by its ID.



Models

- Models represent data structures used in Flutter applications.
- Models encapsulate data and provide methods for serialization.
- Example: Category model (from category.dart)
- Used to structure and manipulate data efficiently.



Models

```
class Category {  
    final int id;  
    final String name;  
    const Category({required this.id, required this.name});  
    static fromMap(Map<String, dynamic> map) {  
        return Category(id: map['id'], name: map['name']);  
    }  
    static List<Category> fromList(List<dynamic> list) {  
        return list.map((item) => Category.fromMap(item)).toList();  
    }  
}
```



Routes

- Routing helps in navigation between screens in a Flutter app.
- Flutter uses Navigator and GoRouter for handling navigation.
- The GoRouter package provides declarative routing.
- Example: GoRouter configuration (routes.dart).



Routing

```
final GoRouter appRouter = GoRouter(  
  initialLocation: '/',  
  routes: [  
    GoRoute(  
      path: '/',  
      builder: (context, state) => const NewsScreen(),  
      routes: [  
        GoRoute(  
          path: 'details/:id',  
          builder: (context, state) => DetailsScreen(  
            id: int.parse(state.pathParameters['id']!),  
          ),  
        ),  
      ],  
    ),  
  ],  
);
```



Future Builder

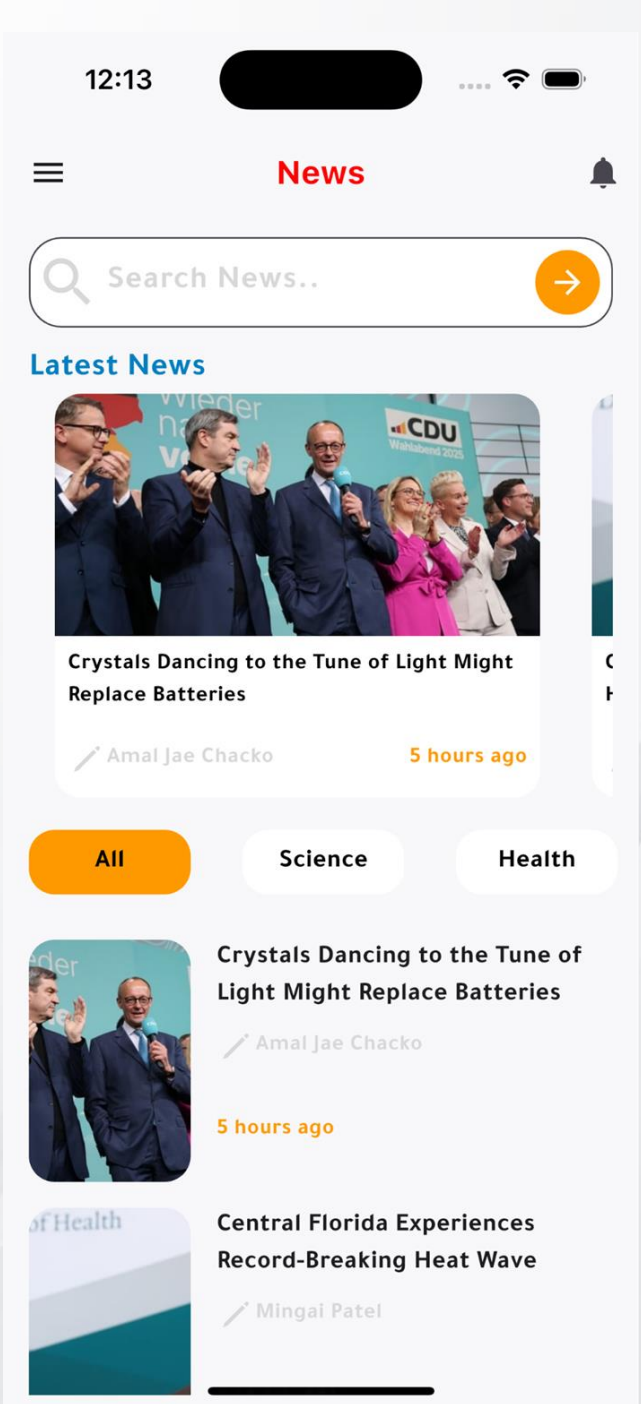
- Models represent data structures used in Flutter applications.
- Models encapsulate data and provide methods for serialization.
- Example: Category model (from category.dart)
- Used to structure and manipulate data efficiently.

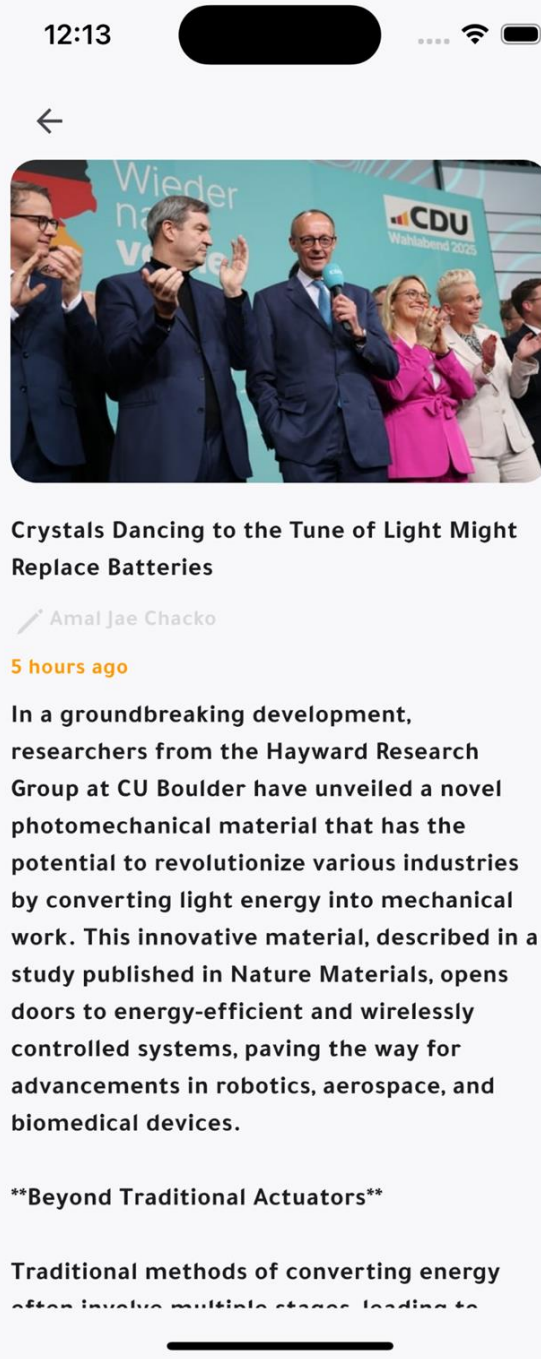


Models

```
class Category {  
    final int id;  
    final String name;  
    const Category({required this.id, required this.name});  
    static fromMap(Map<String, dynamic> map) {  
        return Category(id: map['id'], name: map['name']);  
    }  
    static List<Category> fromList(List<dynamic> list) {  
        return list.map((item) => Category.fromMap(item)).toList();  
    }  
}
```







Live Coding

What is Next ?



Thank You