

## Practical 12: Create and application Parsing JSON data from REST API in Flutter.

Code :

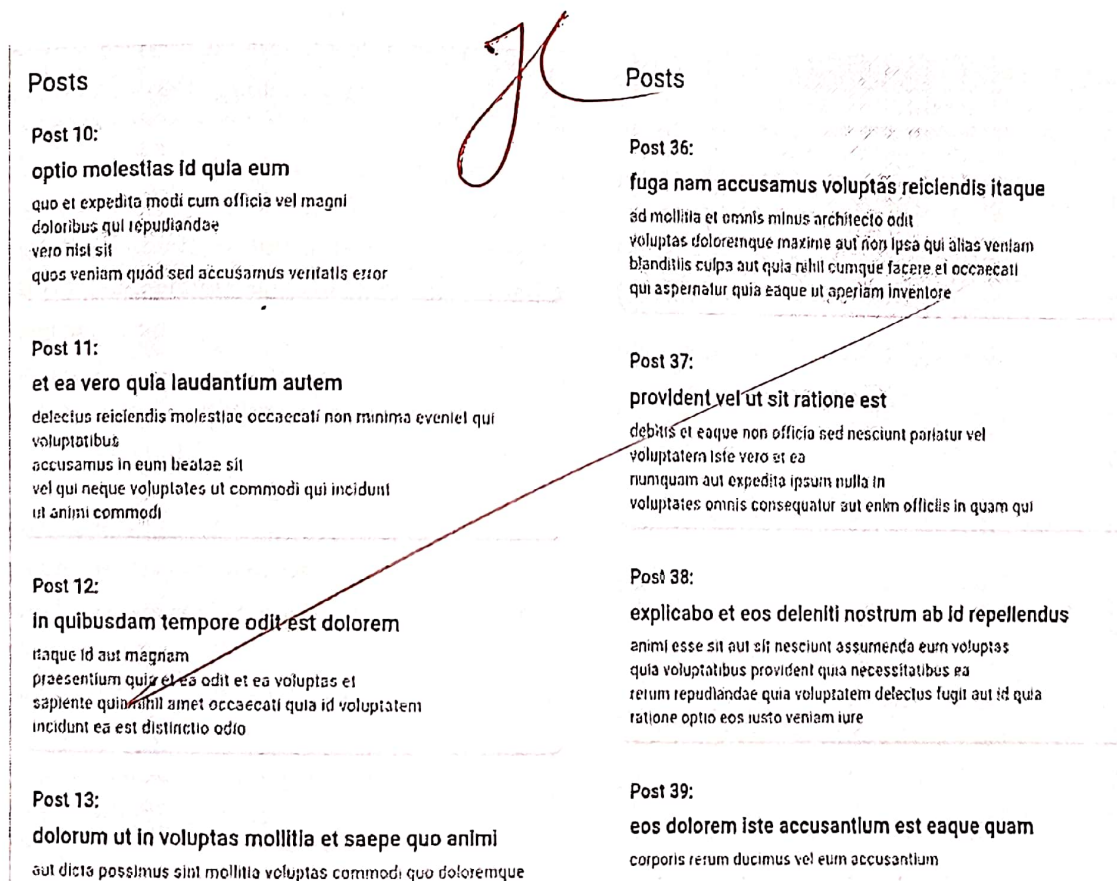
```
import 'package:flutter/material.dart';
import 'package:resetapi/data_screen.dart';
void main() { runApp(MyApp()); } class
MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp( debugShowCheckedModeBanner: false,
    title: 'Flutter REST API Demo', theme: ThemeData(
    primarySwatch: Colors.blue, ),
    home: DataScreen(), ); } }

import 'dart:convert'; import
'package:http/http.dart' as http; class
Post { final int userId; final int id; final
String title; final String body; Post({
required this.userId, required this.id, required this.title, required this.body, });
factory Post.fromJson(Map<String, dynamic> json) { return Post(
userId: json['userId'], id: json['id'], title: json['title'], body:
json['body'], ); } }
class ApiService {
static const String baseUrl = 'https://jsonplaceholder.typicode.com/todos/1'; static
Future<List<Post>> fetchPosts() async {
final response = await http.get(Uri.parse('$baseUrl/posts'));
if (response.statusCode == 200) {
List<dynamic> jsonResponse = json.decode(response.body); return jsonResponse.map((post) =>
Post.fromJson(post)).toList();
} else { throw Exception('Failed to load posts'); } } }
import 'package:flutter/material.dart'; import
'package:resetapi/api_service.dart'; class DataScreen
extends StatefulWidget { @override _DataScreenState
createState() => _DataScreenState(); } class
_DataScreenState extends State<DataScreen> { late
Future<List<Post>> posts;
@override
void initState() { super.initState(); posts
= ApiService.fetchPosts(); }
@override

Widget build(BuildContext context) { return
Scaffold(
```

```
appBar: AppBar( title: Text('Posts'), ),
body: Center(
  child: FutureBuilder<List<Post>>( future: posts,
    builder: (context, snapshot) { if (snapshot.hasData) { return ListView.builder(
      itemCount: snapshot.data!.length, itemBuilder: (context, index) { return
        Card( elevation: 3,
          margin: EdgeInsets.all(10), child: Padding(
            padding: EdgeInsets.all(10), child: Column( crossAxisAlignment: CrossAxisAlignment.start,
              children: [ Text( 'Post ${index + 1}:', // Add label here style: TextStyle(
                fontWeight: FontWeight.bold, fontSize: 16, ), ),
                SizedBox(height: 5), Text( snapshot.data![index].title,
                  style: TextStyle( fontWeight: FontWeight.bold,
                    fontSize: 18, ), ),
                  SizedBox(height: 5), Text(snapshot.data![index].body), ], ), ), ); },
            ) else if (snapshot.hasError) { return Text("${snapshot.error}"); }, ), ), ); }
```

## • Output:



The screenshot shows a mobile application interface with a list of posts. The title 'Posts' is at the top. There are six visible post cards, each with a title, a body of text, and a date. The posts are numbered 10 through 39. A large red 'X' is drawn across the entire screenshot, likely indicating a failed test or a placeholder for a correct output.

**Posts**

**Post 10:**  
optio molestias id quia eum  
quo et expedita modi cum officia vel magni  
doloribus qui repudiandae  
vero nisi sit  
quos veniam quod sed accusamus veritatis error

**Post 11:**  
et ea vero quia laudantium autem  
delectus reiciendis molestiae occaecati non minima eveniet qui  
voluptatibus  
accusamus in eum beatae sit  
vel qui neque voluptates ut commodi qui incidunt  
ut animi commodi

**Post 12:**  
in quibusdam tempore odit est dolore  
itaque id aut magnam  
praesentium qui et ea odit et ea voluptas et  
sapiente quia nihil amet occaecati quia id voluptatem  
incidunt ea est distinctio odio

**Post 13:**  
dolorum ut in voluptas mollitia et saepe quo animi  
aut dicta possimus sint mollitia voluptas commodi quo doloremque

**Post 36:**  
fuga nam accusamus voluptas reiciendis itaque  
ad mollitia et omnis minus architecto odit  
voluptas dolorumque maxime aut non ipsa qui alias veniam  
blanditis culpa aut quia nihil cumque facere et occaecati  
qui aspernatur quia eaque ut aperiam inventore

**Post 37:**  
provident vel ut sit ratione est  
debitis et eaque non officia sed nesciunt pariatur vel  
voluptatem iste vero et ea  
numquam aut expedita ipsum nulla in  
voluptates omnis consequatur aut enim officiis in quam qui

**Post 38:**  
explicabo et eos deleniti nostrum ab id repellendus  
animi esse sit aut sit nesciunt assumenda eum voluptas  
quia voluptatibus provident quia necessitatibus ea  
rerum repudiandae quia voluptatem delectus fugit aut id quia  
ratione optio eos iusto veniam iure

**Post 39:**  
eos dolore iste accusantium est eaque quam  
corporis rerum ducimus vel eum accusantium