9 a) Fetch data from a REST API.

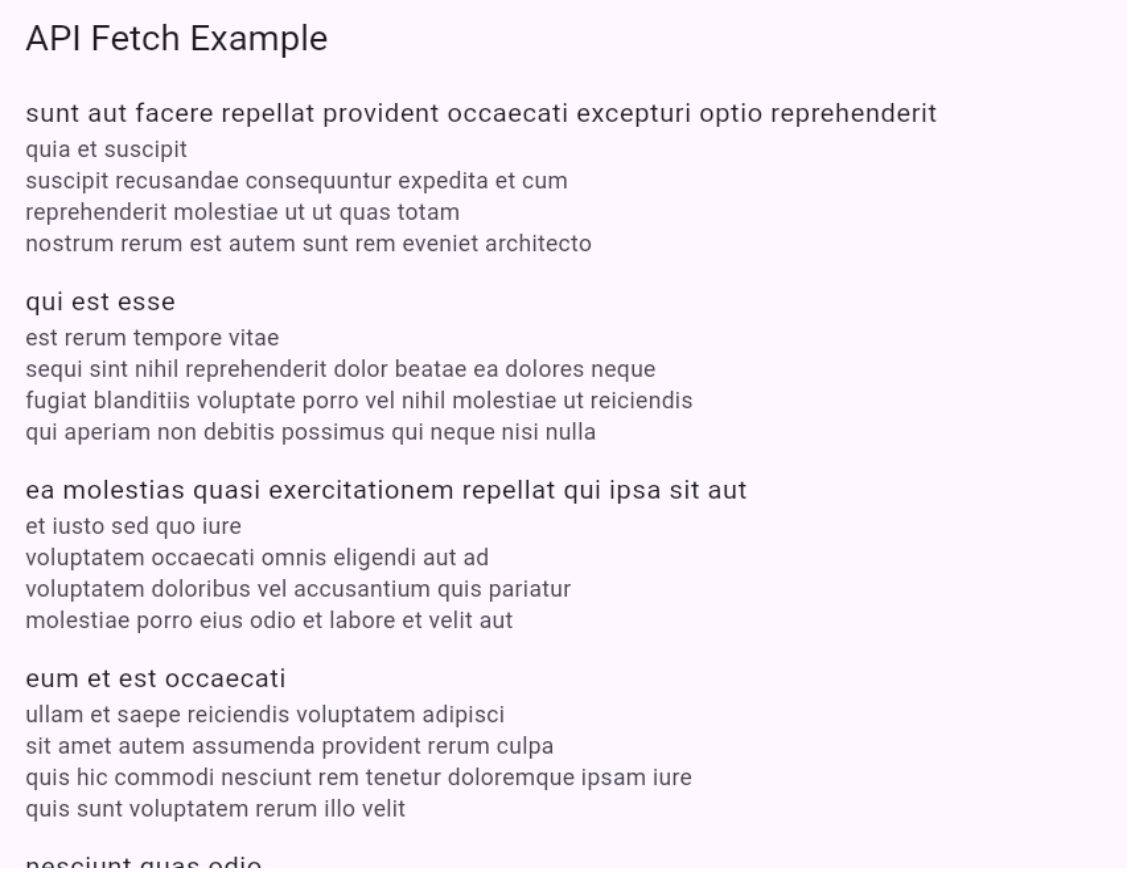
```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';
void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
return MaterialApp(
title: 'API Fetch Example',
theme: ThemeData(
primarySwatch: Colors.blue,
),
home: MyApiFetchWidget(),
);
}
}
class MyApiFetchWidget extends StatefulWidget {
@override
_MyApiFetchWidgetState createState() => _MyApiFetchWidgetState();
}
class _MyApiFetchWidgetState extends State<MyApiFetchWidget> {
late Future<List<Post>> _posts;
@override
void initState() {
super.initState();
_posts = fetchPosts();
}
Future<List<Post>> fetchPosts() async {
```

```
final response =
await http.get(Uri.parse('https://jsonplaceholder.typicode.com/posts'));
if (response.statusCode == 200) {
// If the server returns a 200 OK response,
// parse the JSON and return a list of posts.
List<dynamic> data = json.decode(response.body);
List<Post> posts = data.map((post) => Post.fromJson(post)).toList();
return posts;
} else {
// If the server did not return a 200 OK response,
// throw an exception.
throw Exception('Failed to load posts');
}
}
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text('API Fetch Example'),
),
body: FutureBuilder<List<Post>>(
future: _posts,
builder: (context, snapshot) {
if (snapshot.connectionState == ConnectionState.waiting) {
return CircularProgressIndicator();
} else if (snapshot.hasError) {
return Text('Error: ${snapshot.error}');
} else {
return ListView.builder(
itemCount: snapshot.data!.length,
itemBuilder: (context, index) {
```

```
return ListTile(
title: Text(snapshot.data![index].title),
subtitle: Text(snapshot.data![index].body),
);
},
);
}
},
),
);
}
}
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'],
);
}
```

}

Output:



9 b) Display the fetched data in a meaningful way in the UI.

```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';
void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
return MaterialApp(
title: 'API Fetch Example',
```

```
theme: ThemeData(
primarySwatch: Colors.blue,
),
home: MyApiFetchWidget(),
);
}
}
class MyApiFetchWidget extends StatefulWidget {
@override
_MyApiFetchWidgetState createState() => _MyApiFetchWidgetState();
}
class _MyApiFetchWidgetState extends State<MyApiFetchWidget> {
late Future<List<Post>> _posts;
@override
void initState() {
super.initState();
_posts = fetchPosts();
}
Future<List<Post>> fetchPosts() async {
final response =
await http.get(Uri.parse('https://jsonplaceholder.typicode.com/posts'));
if (response.statusCode == 200) {
List<dynamic> data = json.decode(response.body);
List<Post> posts = data.map((post) => Post.fromJson(post)).toList();
return posts;
} else {
throw Exception('Failed to load posts');
}
}
@override
Widget build(BuildContext context) {
```

```
return Scaffold(
appBar: AppBar(
title: Text('API Fetch Example'),
),
body: FutureBuilder<List<Post>>(
future: _posts,
builder: (context, snapshot) {
if (snapshot.connectionState == ConnectionState.waiting) {
return Center(child: CircularProgressIndicator());
} else if (snapshot.hasError) {
return Center(child: Text('Error: ${snapshot.error}'));
} else {
return PostList(posts: snapshot.data!);
}
},
),
);
}
}
class PostList extends StatelessWidget {
final List<Post> posts;
PostList({required this.posts});
@override
Widget build(BuildContext context) {
return ListView.builder(
itemCount: posts.length,
itemBuilder: (context, index) {
return PostItem(post: posts[index]);
},
);
}
```

```
}
class PostItem extends StatelessWidget {
final Post post;
PostItem({required this.post});
@override
Widget build(BuildContext context) {
return Card(
margin: EdgeInsets.all(10),
elevation: 3,
child: Padding(
padding: EdgeInsets.all(15),
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Text(
post.title,
style: TextStyle(
fontSize: 18,
fontWeight: FontWeight.bold,
),
),
SizedBox(height: 10),
Text(
post.body,
style: TextStyle(fontSize: 16),
),
],
),
),
);
}
```

```
}
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'],
);
}
}
```

Output:

API Fetch Example

sunt aut facere repellat provident occaecati excepturi optio reprehenderit

quia et suscipit
suscipit recusandae consequuntur expedita et cum
reprehenderit molestiae ut ut quas totam
nostrum rerum est autem sunt rem eveniet architecto

qui est esse

est rerum tempore vitae
sequi sint nihil reprehenderit dolor beatae ea dolores neque
fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis
qui aperiam non debitis possimus qui neque nisi nulla

ea molestias quasi exercitationem repellat qui ipsa sit aut

et iusto sed quo iure
voluptatem occaecati omnis eligendi aut ad
voluptatem doloribus vel accusantium quis pariatur
molestiae porro eius odio et labore et velit aut