

EX-6 An Application To Connect To A Web Service And To Retrieve Data With HTTP

Aim

To develop an application to connect to a web service and to retrieve data with HTTP using flutter.

Definitions

Flutter

Flutter is not a programming language. It's a software development kit (SDK) with prewritten code, consisting of ready-to-use and customizable widgets, as well as libraries, tools, and documentation that together serve to build cross-platform apps.

Flutter plugin

A Flutter plugin is a special kind of package that enables Flutter apps to interact with platform-specific APIs (iOS, Android, web, desktop). Plugins can include Dart code, but crucially, they also contain platform-specific implementation code written in Kotlin/Java for Android and Swift/Objective-C for iOS.

Dart plugin

The Dart plugin adds Dart support to IntelliJ Platform-based IDEs developed by JetBrains. These IDEs provide features unique to specific development technologies. The IDEs recommended for Dart and Flutter development include: IntelliJ IDEA which specializes in JVM-based language development.

Flutter SDK

Flutter is Google's free, open-source software development kit (SDK) for cross-platform mobile application development. Using a single platform-agnostic codebase, Flutter helps developers build high-performance, scalable applications with attractive and functional user interfaces for Android or iOS.

Web service

A web service is a software system designed to allow different applications to interact with each other over the internet. They use standardized protocols, like HTTP, and XML to exchange data, enabling communication and data sharing between diverse systems.

HTTP

HTTP stands for Hypertext Transfer Protocol. It's the foundation of data communication on the World Wide Web, allowing web browsers and servers to exchange information and enabling users to access websites. HTTP is a request-response protocol, meaning a client (like a web browser) sends a request to a server, and the server responds with the requested content.

Procedure

1. **Open android studio**
2. **Click ‘new flutter project’**
3. **Select ‘flutter’ at the left side of the window**
4. **Add ‘flutter sdk’ from the desired location**
5. **Click ‘next’ and specify the project name and select language ‘java’, check only Android, Web and Windows under platforms then click ‘create’**
6. **Create a new dart file under ‘lib’ folder in the projects window (right click over lib folder -> new -> dart file -> specify the file name as ‘ws_http’ -> press ‘enter’**
7. **Open terminal in android -> type “flutter pub add http” (This command will add http: ^1.3.0 dependency in pubspec.yaml file) -> press enter**
8. **Type the following codes in the ws_http.dart file**

ws_http.dart

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
void main() {
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'SpaceX Launches',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: LaunchListScreen(),
    );
  }
}
class LaunchListScreen extends StatefulWidget {
  @override
  _LaunchListScreenState createState() => _LaunchListScreenState();
}
class _LaunchListScreenState extends State<LaunchListScreen> {
  List<dynamic> _launches = [];
  bool _isLoading = true;
  @override
  void initState() {
    super.initState();
    fetchData();
  }
}
```

```

}
Future<void> fetchData() async {
  final response = await http.get(Uri.parse('https://api.spacexdata.com/v4/launches/upcoming'));
  if (response.statusCode == 200) {
    setState(() {
      _launches = jsonDecode(response.body);
      _isLoading = false;
    });
  } else {
    setState(() {
      _isLoading = false;
      _launches = [];
    });
    showDialog(
      context: context,
      builder: (context) => AlertDialog(
        title: Text('Error'),
        content: Text('Failed to load launches.'),
        actions: [
          TextButton(
            onPressed: () => Navigator.pop(context),
            child: Text('OK'),
          ),
        ],
      ),
    );
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('SpaceX Launches'),
    ),
    body: _isLoading
      ? Center(
        child: CircularProgressIndicator(),
      )
      : _launches.isEmpty
        ? Center(
          child: Text('No launches available.'),
        )
        : ListView.builder(
          itemCount: _launches.length,
          itemBuilder: (context, index) {
            final launch = _launches[index];

```

```

return ListTile(
  title: Text(launch['name']),
  subtitle: Text('Date: ${launch['date_local']}'),
  onTap: () {
// Add onTap functionality if needed
  },
);
},
),
);
}
}

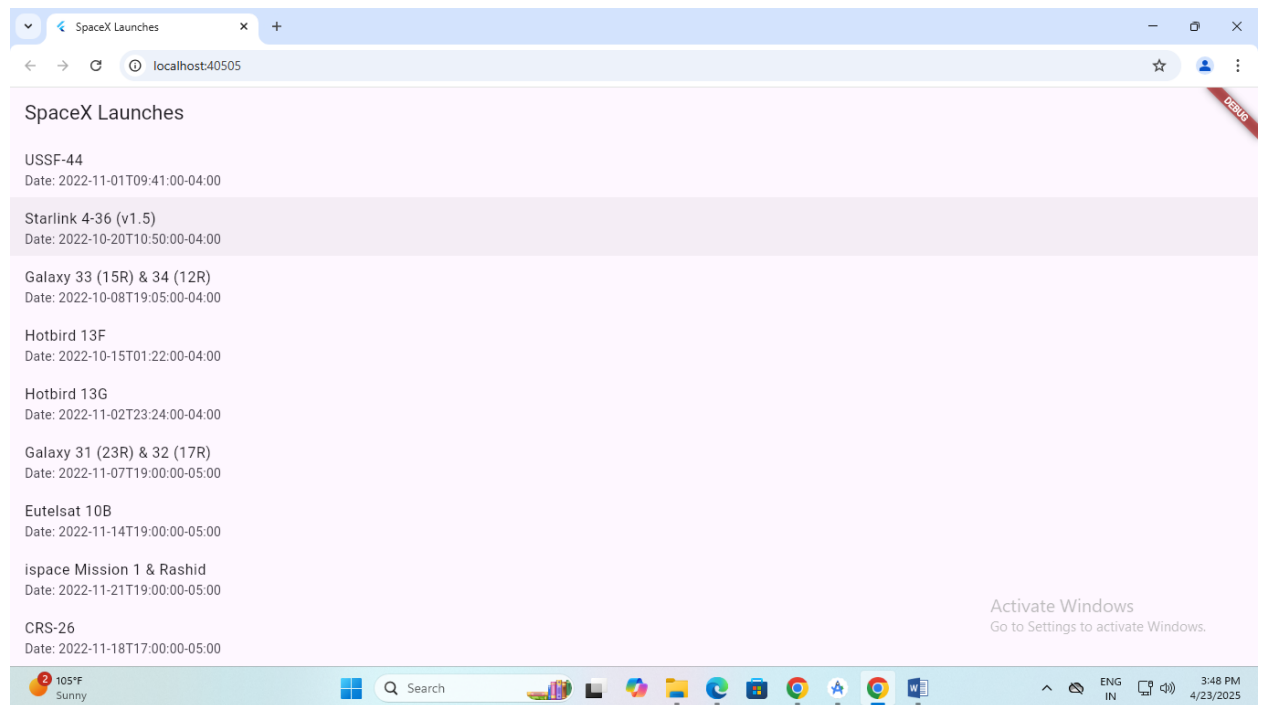
```

9. Save the file ws_http.dart (click main menu -> saveall)

10. Select device as 'chrome(web)'

11. Click on run/debug configuration -> edit configurations -> specify dart file name (ws_http.dart) -> browse and set dart entrypoint as ws_http.dart -> click ok -> click run

Output



Result

Thus, an application to connect to a web service and to retrieve data with HTTP using flutter has been developed.