

## **EX-8**

## **A Web Server Supporting Push Notifications**

### **Aim**

To develop a web server supporting push notifications 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 server**

A web server is a computer system (software and hardware) that responds to client requests via HTTP (or HTTPS) and delivers web content, such as webpages, images, and videos, to users over the internet. It's the backbone of the internet, enabling users to access and interact with websites and online applications.

#### **Push Notifications**

Push notifications are messages sent by an application or website to a user's device, even when the app isn't open or the user is not actively browsing the website. These notifications are displayed as pop-ups on the device's screen, usually in the notification center or lock screen, and can contain text, images, and buttons for user engagement.

## 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\_push-notify’ -> press ‘enter’)**
7. **Open terminal in android -> type “flutter pub add html” and “flutter pub add web\_socket\_channel” (This command will add html and web\_socket\_channel dependencies in pubspec.yaml file) -> press enter**
8. **Type the following codes in the ws\_push-notify.dart file**

**ws\_push-notify.dart**

```
import 'package:flutter/material.dart';
import 'package:web_socket_channel/html.dart';
import 'package:web_socket_channel/web_socket_channel.dart';
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'WebSocket Demo',
      home: WebSocketDemo(),
    );
  }
}

class WebSocketDemo extends StatefulWidget {
  @override
  _WebSocketDemoState createState() => _WebSocketDemoState();
}

class _WebSocketDemoState extends State<WebSocketDemo> {
  final channel = HtmlWebSocketChannel.connect('wss://echo.websocket.org');
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('WebSocket Demo'),
```

```

),
body: Center(
  child: StreamBuilder(
    stream: channel.stream,
    builder: (context, snapshot) {
      return snapshot.hasData
        ? Text(
            'Received: ${snapshot.data}',
            style: TextStyle(fontSize: 24),
          )
        : CircularProgressIndicator();
    },
  ),
),
floatingActionButton: FloatingActionButton(
  onPressed: () {
    channel.sink.add('Hello, WebSocket!');
  },
  child: Icon(Icons.send),
),
);
}
@override
void dispose() {
  channel.sink.close();
  super.dispose();
}
}

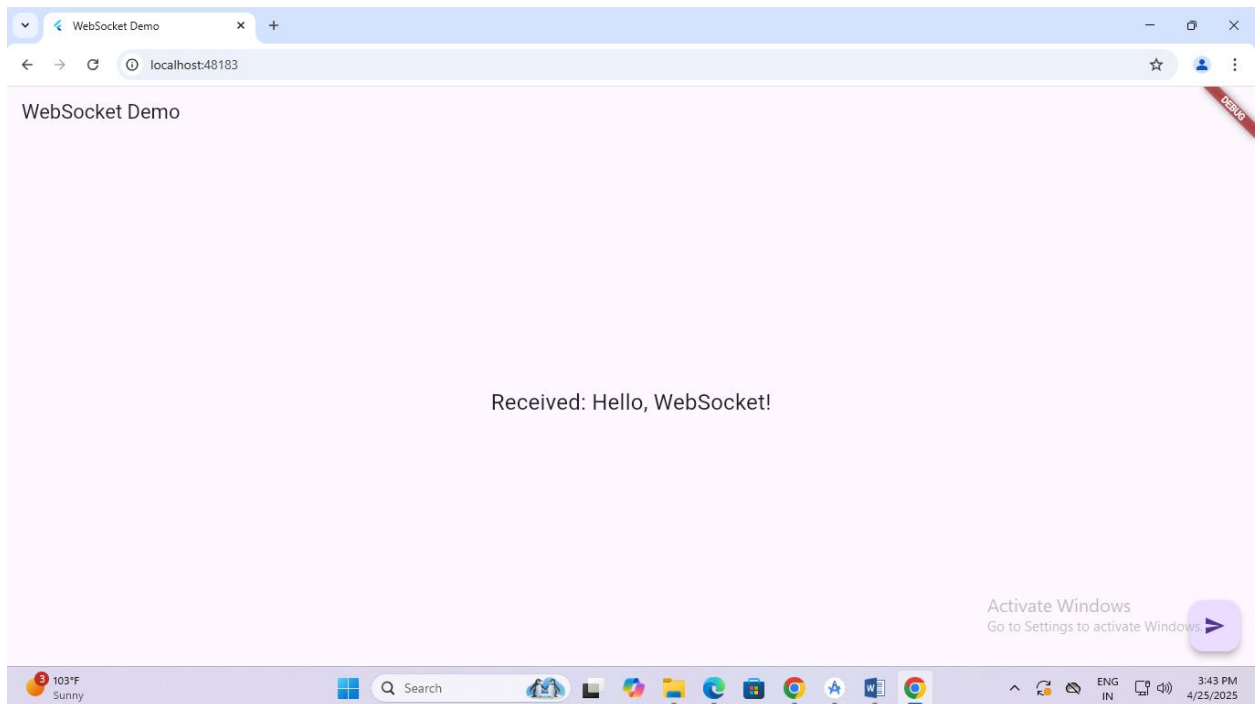
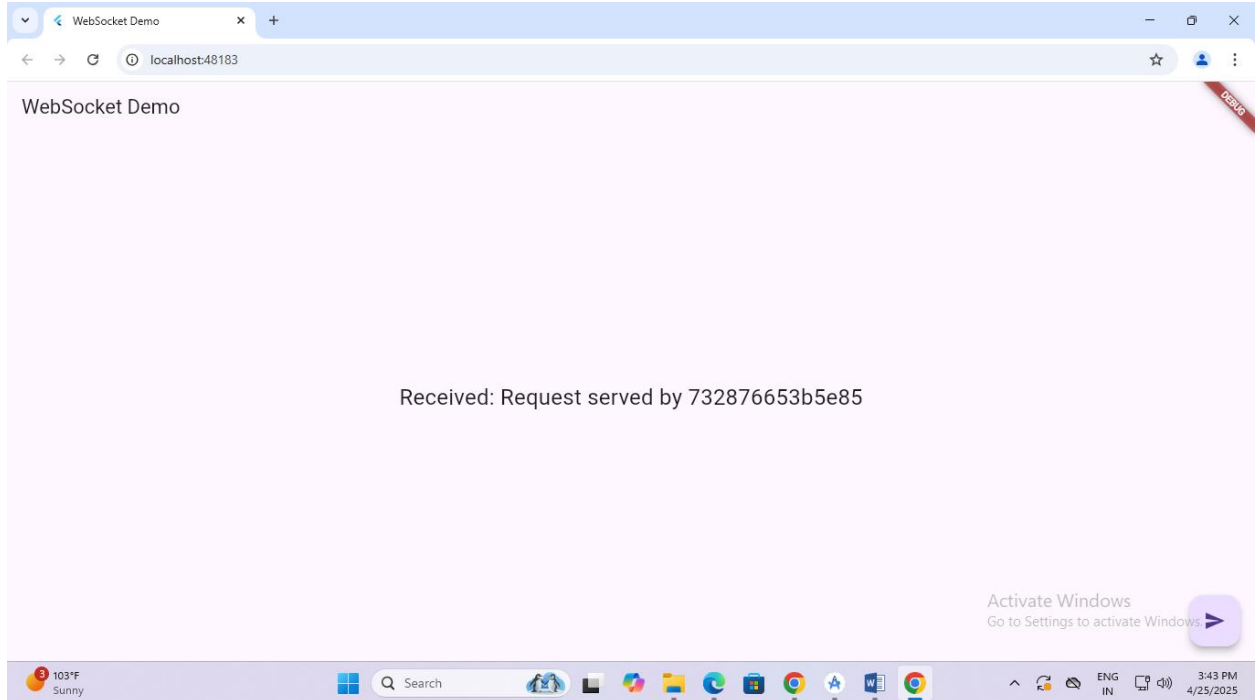
```

**9. Save the file ws\_push-notify.dart (click main menu -> saveall)**

**10. Select device as 'chrome(web)'**

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

## Output



## **Result**

Thus, a web server supporting push notifications using flutter has been developed.