|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

**Лабораторная работа №6**

по курсу «Разработка мобильных приложений».

«Разработка клиент-серверного приложения web-sockets»

Выполнил:

студент группы ИУ9-71Б

Митрошкин Алексей

Проверил:

Посевин Д.П.

Москва, 2024

1. **Цель**

Разработать клиент-серверное приложение с применением dart и flutter.

1. **Задание**

* Разработка клиента на flutter.
* Разработка сервера на dart.

1. **Реализация**

Исходный код сервера приведён в Листинге 1.

Листинг 1

|  |
| --- |
| import 'dart:convert';  import 'dart:io';  const String filePath = 'number.json';  void main() async {    final server = await HttpServer.bind(InternetAddress.anyIPv4, 8000);    print('WebSocket server is listening on ws://localhost:8000');    await for (HttpRequest request in server) {      if (WebSocketTransformer.isUpgradeRequest(request))        WebSocketTransformer.upgrade(request).then(handleWebSocket);    }  }  void handleWebSocket(WebSocket socket) {    socket.listen((message) async {      Map<String, dynamic> data = jsonDecode(message);      if (data['action'] == 'setNumber' && data.containsKey('number')) {        final int number = data['number'];        await setNumber(number);        print("Num saved: $number");        socket.add(jsonEncode({'status': 'ok', 'message': 'Num saved: $number'}));      } else if (data['action'] == 'getNumber') {        int number = await getNumber();        print("Num retrieved: $number");        socket.add(jsonEncode({'status': 'ok', 'number': number}));      }    });  }  Future<void> setNumber(int number) async {    await File(filePath).writeAsString(jsonEncode({'number': number}));  }  Future<int> getNumber() async {    if (await File(filePath).exists()) {      String content = await File(filePath).readAsString();      Map<String, dynamic> data = jsonDecode(content);      return data['number'] ?? 0;    } else {      return 0;    }  } |

В листиге 2 приведён исходный код клиентского приложения.

Листинг 2

|  |
| --- |
| import 'dart:convert'; import 'package:flutter/material.dart'; import 'package:web\_socket\_channel/io.dart'; import 'package:web\_socket\_channel/web\_socket\_channel.dart';  class WS\_Screen extends StatelessWidget {  const WS\_Screen({super.key});   @override  Widget build(BuildContext context) {  return MaterialApp(  title: 'WS Screen',  theme: ThemeData(  primarySwatch: Colors.*blue*,  ),  home: const NumberForm(),  );  } }  class NumberForm extends StatefulWidget {  const NumberForm({super.key});   @override  \_NumberFormState createState() => \_NumberFormState(); }  class \_NumberFormState extends State<NumberForm> {  final \_formKey = GlobalKey<FormState>();  final \_numberController = TextEditingController();  int \_currentNumber = 0;   late WebSocketChannel \_channel;   @override  void initState() {  super.initState();   try {  \_channel = IOWebSocketChannel.connect('ws://192.168.191.18:8000');  } catch (e) {  print("Error in WebSocket connection: $e");  }  }   void sendNumber(int number) {  final data = jsonEncode({  'action': 'setNumber',  'number': number,  });  print("Sending number: $number");  \_channel.sink.add(data);  }   void getNumber() {  final data = jsonEncode({  'action': 'getNumber',  });  \_channel.sink.add(data);  }   void increment() {  setState(() {  \_currentNumber++;  });  sendNumber(\_currentNumber);  }   void decrement() {  setState(() {  \_currentNumber--;  });  sendNumber(\_currentNumber);  }   @override  void dispose() {  \_channel.sink.close();  super.dispose();  }   @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(  title: const Text('WS Screen'),  ),  body: Padding(  padding: const EdgeInsets.all(16.0),  child: Form(  key: \_formKey,  child: Column(  children: <Widget>[  TextFormField(  controller: \_numberController,  decoration: const InputDecoration(labelText: 'Запишите число'),  keyboardType: TextInputType.*number*,  validator: (value) {  if (value == null || value.isEmpty) {  return 'Запишите число';  }  return null;  },  ),  const SizedBox(height: 20),  ElevatedButton(  onPressed: () {  if (\_formKey.currentState!.validate()) {  final number = int.*parse*(\_numberController.text);  sendNumber(number);  }  },  child: const Text('Отправить'),  ),  const SizedBox(height: 20),  ElevatedButton(  onPressed: getNumber,  child: const Text('Получить число'),  ),  const SizedBox(height: 20),  Text('Current: $\_currentNumber', style: const TextStyle(fontSize: 20)),  const SizedBox(height: 20),  Row(  mainAxisAlignment: MainAxisAlignment.center,  children: [  ElevatedButton(  onPressed: decrement,  child: const Text('Уменьшить'),  ),  const SizedBox(width: 20),  ElevatedButton(  onPressed: increment,  child: const Text('Увеличить'),  ),  ],  ),  ],  ),  ),  ),  );  } } |

На рисунках 1-2 приведёны результаты работы программы

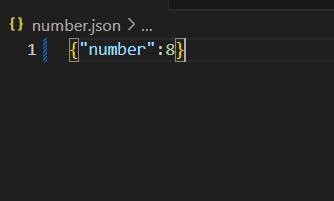


Рисунок 1

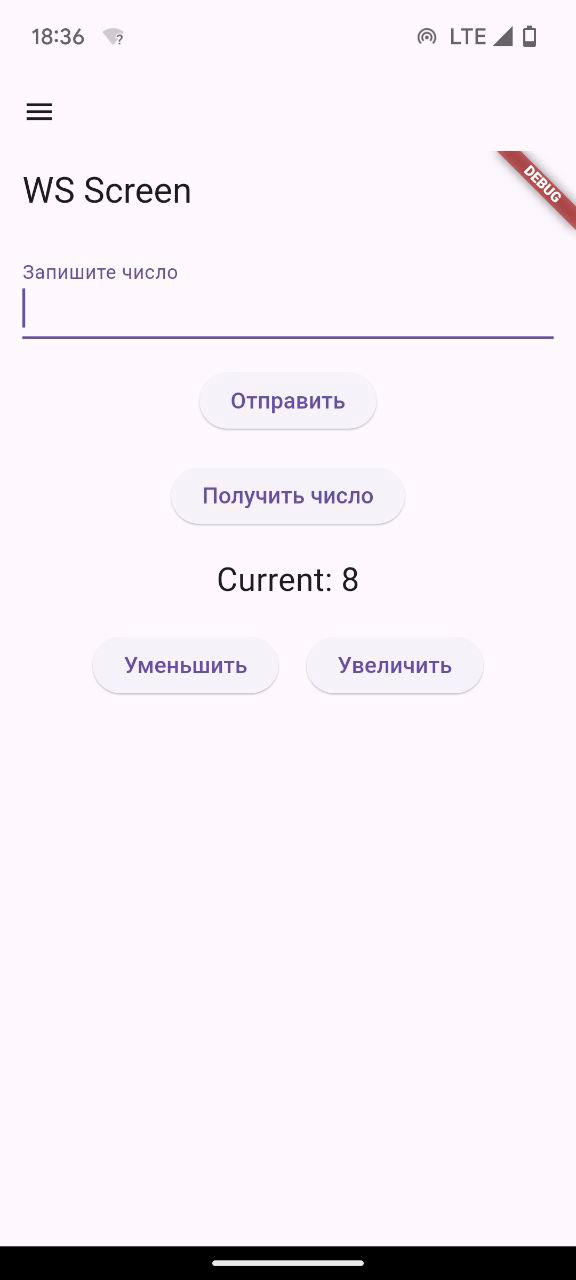


Рисунок 2