



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 _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа №3
по курсу «Разработка мобильных приложений»
«Разработка Web-сервера на Dart и использование запросов к
нему из мобильного приложения.»

Студент группы ИУ9-71Б Окутин Д.А.

Преподаватель Посевин Д. П.

Москва 2024

1 Задание

Цель данной лабораторной работы: используя средства языка Dart, реализовать простейший Web-сервер, который умеет принимать на вход число, сохранять его в файл и отдавать по запросу. Использовать этот Web-сервер для усовершенствования приложения "кликер".

2 Реализация

Исходный код программы представлен в листингах 1– 4.

Листинг 1 — Мобильное приложение

```
1 import 'package:flutter/material.dart';
2 import 'package:http/http.dart' as http;
3
4 void main() {
5   runApp(MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      title: 'Lab2',
13      theme: ThemeData(
14        primarySwatch: Colors.blue,
15      ),
16      home: NumberForm(),
17    );
18  }
19 }
20
21 class NumberForm extends StatefulWidget {
22   @override
23   _NumberFormState createState() => _NumberFormState();
24 }
25
26 class _NumberFormState extends State<NumberForm> {
27   final _formKey = GlobalKey<FormState>();
28   final _numberController = TextEditingController();
29   int _currentNumber = 0;
30
31   Future<void> sendNumber() async {
32     final url = Uri.parse('http://195.19.42.184:8080/');
33     final response = await http.post(
34       url,
35       headers: {
36         'Content-Type': 'text/plain',
37       },
38       body: '$_currentNumber',
39     );
40
41     if (response.statusCode == 200) {
42       print('Num sent: ${_currentNumber}');
43     } else {
44       print('Error: ${response.statusCode}');
45     }
46   }
```

Листинг 2 — Мобильное приложение

```
1 Future<void> sendNumberFromNumberController() async {
2   final url = Uri.parse('http://195.19.42.184:8080/');
3   final response = await http.post(
4     url,
5     headers: {
6       'Content-Type': 'text/plain',
7     },
8     body: _numberController.text,
9   );
10
11   if (response.statusCode == 200) {
12     print('Num sent: ${_numberController.text}');
13   } else {
14     print('Error: ${response.statusCode}');
15   }
16 }
17
18 Future<void> getNumber() async {
19   final url = Uri.parse('http://195.19.42.184:8080/');
20
21   final response = await http.get(url);
22
23   if (response.statusCode == 200) {
24     setState(() {
25       _currentNumber = int.parse(response.body);
26     });
27
28     print('Num: $_currentNumber');
29   } else {
30     print('Error: ${response.statusCode}');
31   }
32 }
33
34 void increment() {
35   setState(() {
36     _currentNumber++;
37   });
38
39   sendNumber();
40 }
41
42 void decrement() {
43   setState(() {
44     _currentNumber--;
45   });
46
47   sendNumber();
48 }
```

Листинг 3 — Мобильное приложение

```

1  @override
2  Widget build(BuildContext context) {
3    return Scaffold(
4      appBar: AppBar(
5        title: Text('Lab2'),
6      ),
7      body: Padding(
8        padding: const EdgeInsets.all(16.0),
9        child: Form(
10         key: _formKey,
11         child: Column(
12           children: <Widget>[
13             TextFormField(
14               controller: _numberController,
15               decoration: InputDecoration(labelText: 'Write num'),
16               keyboardType: TextInputType.number,
17               validator: (value) {
18                 if (value == null || value.isEmpty) {
19                   return 'Write num';
20                 }
21                 return null;
22               },
23             ),
24             const SizedBox(height: 20),
25             ElevatedButton(
26               onPressed: () {
27                 if (_formKey.currentState!.validate()) {
28                   sendNumberFromNumberController();
29                 }
30               },
31               child: Text('Text'),
32             ),
33             const SizedBox(height: 20),
34             ElevatedButton(
35               onPressed: getNumber,
36               child: Text('Get num'),
37             ),
38             const SizedBox(height: 20),
39             Text('Current: $_currentNumber', style: TextStyle(fontSize
40               : 20)),
41             const SizedBox(height: 20),
42             Row(
43               mainAxisAlignment: MainAxisAlignment.center,
44               children: [
45                 ElevatedButton(
46                   onPressed: decrement,
47                   child: Text('- Less'),
48                 ),
49                 const SizedBox(width: 20),
50                 ElevatedButton(
51                   onPressed: increment,
52                   child: Text('+ Bigger'),
53                 ),
54               ],
55             ),
56           ],
57         ),
58       ),
59     );
60   }
61 }

```

Листинг 4 — Web-server

```

1
2 import 'dart:convert';
3 import 'dart:io';
4
5 Future<void> main() async {
6   final server = await HttpServer.bind(InternetAddress.anyIPv4, 8080);
7   print('http://${server.address.host}:${server.port}/');
8
9   await for (var request in server) {
10    if (request.method == 'POST') {
11      await handlePost(request);
12    } else if (request.method == 'GET') {
13      await handleGet(request);
14    } else {
15      request.response.statusCode = HttpStatus.methodNotAllowed;
16      request.response.write('Not allowed');
17      request.response.close();
18    }
19  }
20 }
21
22 Future<void> handlePost(HttpRequest request) async {
23   try {
24     String content = await utf8.decoder.bind(request).join();
25     int number = int.parse(content);
26
27     final file = File('data.txt');
28     await file.writeAsString('$number');
29
30     print('Saved: $number');
31
32     request.response
33       ..statusCode = HttpStatus.ok
34       ..write(number)
35       ..close();
36   } catch (e) {
37     request.response
38       ..statusCode = HttpStatus.badRequest
39       ..write('Error: ${e.toString()}')
40       ..close();
41   }
42 }
43
44 Future<void> handleGet(HttpRequest request) async {
45   final file = File('data.txt');
46   if (await file.exists()) {
47     String content = await file.readAsString();
48     print('Num: $content');
49     request.response
50       ..statusCode = HttpStatus.ok
51       ..write(content)
52       ..close();
53   } else {
54     request.response
55       ..statusCode = HttpStatus.notFound
56       ..write('File not found')
57       ..close();
58   }
59 }

```

3 Результаты

Результат представлен на рисунках 1 - 2.

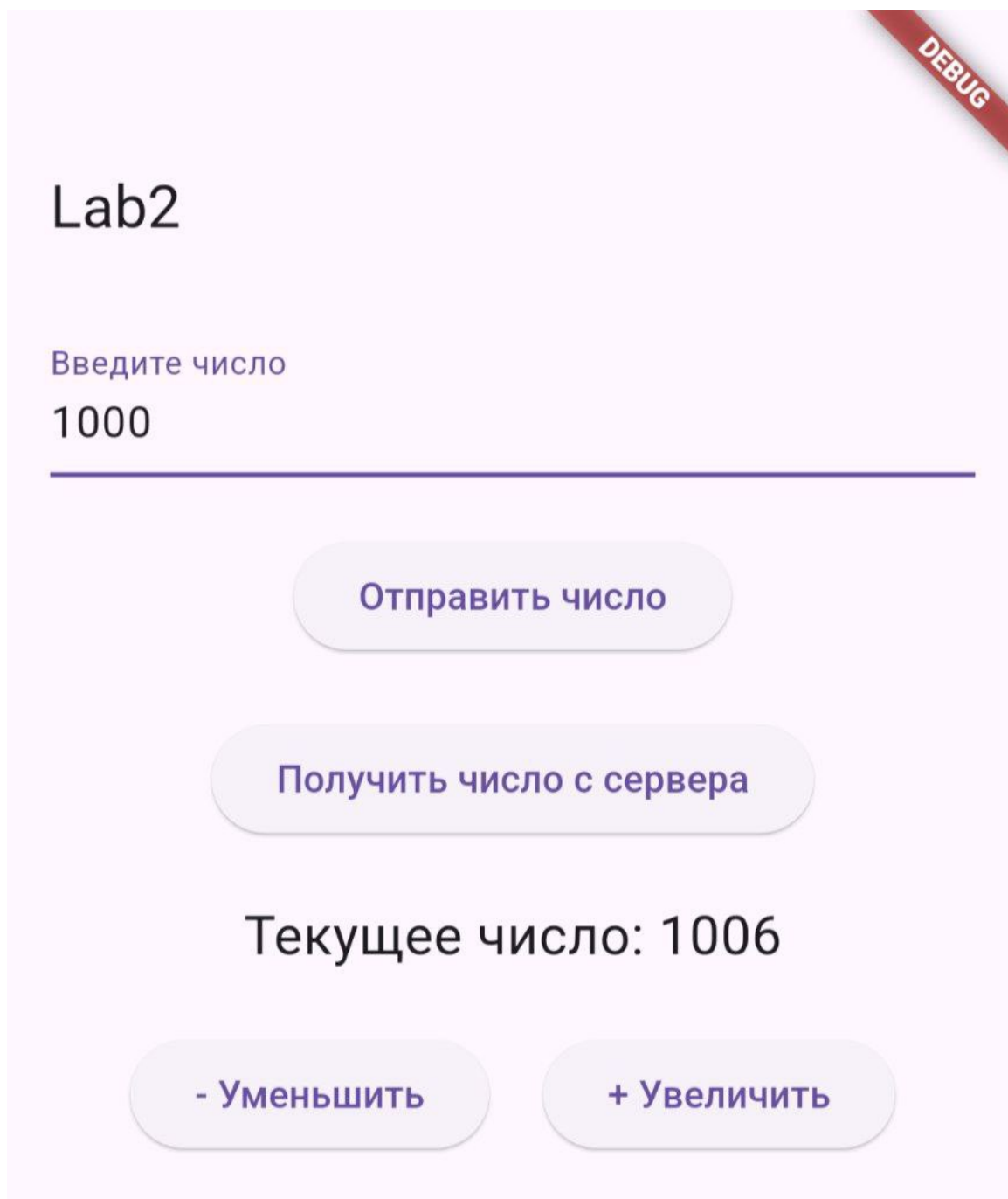


Рис. 1 — Интерфейс приложения

```
Сервер запущен на http://0.0.0.0:8080/  
Число сохранено: 100  
Число сохранено: 123  
Число сохранено: 23  
Сохраненное число: 23  
Сохраненное число: 23  
Сохраненное число: 23  
Сохраненное число: 23  
Сохраненное число: 23
```

Рис. 2 — Логи сервера

4 Выводы

В результате выполнения данной работы был реализован Web-сервер, который поддерживает сохранение числа в файл и считывание из него. Помимо этого был доработан кликер, который теперь способен брать значение с сервера и увеличивать и уменьшать его.