



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 № 7 по курсу «Компьютерные сети»

«Информационная система мониторинга блоков в блокчейн  
Ethereum»

Студент группы ИУ9-32Б Тараканов В. Д..

Преподаватель Посевин Д. П.

Москва 2024

# 1 Задание

Реализовать приложение мониторинга состояния заданных блоков блокчейн Ethereum. Данные результатов мониторинга должны записываться в Firebase. Аккаунт в infura.io необходимо зарегистрировать свой, также создать собственную Realtime Database.

## 2 Результаты

Исходный код программы представлен в листингах 1.

Листинг 1: client.go

```
1  package main
2
3  import (
4      "context"
5      firebase "firebase.google.com/go/v4"
6      "fmt"
7      "github.com/ethereum/go-ethereum/core/types"
8      "github.com/ethereum/go-ethereum/ethclient"
9      "google.golang.org/api/option"
10     "log"
11     "math/big"
12     "time"
13 )
14
15 func senddata(block *types.Block) {
16     opt := option.WithCredentialsFile("./lab7.json")
17     config := &firebase.Config{
18         DatabaseURL: "https://lab7-6e4bd-default-rtdb.firebaseio.com",
19     }
20     app, err := firebase.NewApp(context.Background(), config, opt)
21     if err != nil {
22         log.Fatalf("error initializing app: %v\n", err)
23     }
24     client, err := app.Database(context.Background())
25     if err != nil {
26         log.Fatalf("error initializing app: %v\n", err)
27     }
28
29     ref := client.NewRef("/") + block.Number().String()
30     send_data := map[string]interface{}{
31         "number":      block.Number().String(),
```

```

32         "time":          block.Time() ,
33         "difficulty":    block.Difficulty().String() ,
34         "hash":          block.Hash().String() ,
35     }
36
37     if err := ref.Set(context.Background(), send_data); err != nil {
38         log.Fatalf("error sending data: %v\n", err)
39     }
40
41     for _, tx := range block.Transactions() {
42         if tx.To() == nil {
43             continue
44         }
45         ref = client.NewRef("/") + block.Number().String() + "/"
transactions/" + tx.Hash().String())
46         transaction_data := map[string]interface{}{
47             "chainID":    tx.ChainId() ,
48             "hash":        tx.Hash().String() ,
49             "to":           tx.To().Hex() ,
50             "value":       tx.Value().String() ,
51             "cost":        tx.Cost().String() ,
52             "gas":          tx.Gas() ,
53             "gas price": tx.GasPrice().String() }
54         if err := ref.Set(context.Background(), transaction_data); err
!= nil {
55             log.Fatalf("error sending data: %v\n", err)
56         }
57     }
58
59     fmt.Println("sent data block number: " + block.Number().String())
60 }
61
62 func main() {
63     client, err := ethclient.Dial("https://mainnet.infura.io/v3/1
cb3c9bac5ad452eb9e994c328660611")
64     if err != nil {
65         log.Fatalln(err)
66     }
67
68     if err != nil {
69         log.Fatal(err)
70     }
71     header, err := client.HeaderByNumber(context.Background(), nil)
72     prev := header.Number.String()
73     for {
74         header, err := client.HeaderByNumber(context.Background(), nil)

```

```

75         if err != nil {
76             log.Fatal(err)
77         }
78         if prev != header.Number.String() {
79             prev = header.Number.String()
80             fmt.Println("New block")
81             fmt.Println(header.Number.String()) // The lastes block in
blockchain because nil pointer in header
82
83             blockNumber := big.NewInt(header.Number.Int64())
84             block, err := client.BlockByNumber(context.Background(),
blockNumber) //get block with this number
85             if err != nil {
86                 log.Fatal(err)
87             }
88             // all info about block
89             fmt.Println(block.Number().String())
90             fmt.Println(block.Time())
91             fmt.Println(block.Difficulty().String())
92             fmt.Println(block.Hash().String())
93             fmt.Println(len(block.Transactions()))
94             go senddata(block)
95             fmt.Println()
96         }
97         time.Sleep(1000 * time.Millisecond)
98     }
99 }
100
101

```