**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИИ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГАОУ ВО СевГУ)**

Кафедра информационных технологий и компьютерных систем

Лабораторная работа №5  
дисциплина: Вычислительные системы  
тема: «http-сервер в Ruby».

Выполнил: ст. группы: ИВТ-17-о  
Коломбет В. А.  
Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Севастополь,

2020

**Цель работы:** ознакомиться с основами использования API для организации http-сервера в языке Ruby.

**Постановка задачи.** Написать скрипт, в котором переписать банкомат из ЛЗ№4 на работу с сервером. Заменить ввод-вывод из консоли на запросы.

**Результат выполнения:**

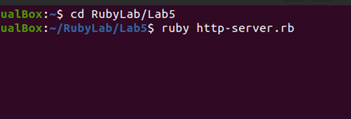


Рисунок 1 – Запуск сервера.

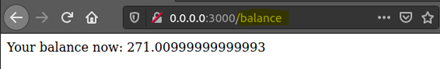


Рисунок 2 - Проверка баланса и подключение к серверу через браузер.

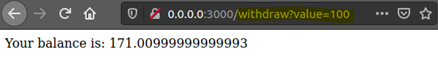


Рисунок 3 - Результат снятия 100 у.е. при помощи get-запроса.

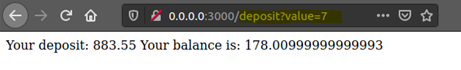


Рисунок 4 - Результат взятия депозита get-запросом.

**Код программы**

**Файл http-server.rb:**

|  |
| --- |
| require 'socket' |
|  | require 'rack' |
|  | require 'rack/utils' |
|  | load 'script1.rb' |
|  |  |
|  | server = TCPServer.new('0.0.0.0', 3000) |
|  |  |
|  | class App |
|  |  |
|  | def initialize |
|  | @machine = CashMachine.new |
|  | end |
|  |  |
|  | def call(env) |
|  | req = Rack::Request.new(env) |
|  |  |
|  | case req.path |
|  | when '/deposit' |
|  | value = req.query\_string.split('=')[1].to\_f.floor(2) |
|  | @machine.deposit(value)? err = '': err = "Error! You input wrong number (#{value}). Change value for take deposit & try again" |
|  | [200, {'Content-Type' => 'text/html'}, ["Your deposit: #{@machine.get\_deposit} Your balance is: #{@machine.money}\n#{err}"]] |
|  | when '/withdraw' |
|  | value = req.query\_string.split('=')[1].to\_f.floor(2) |
|  | @machine.withdraw(value)? err = '': err = "Error! You input wrong number (#{value}). Change value for take withdraw & try again" |
|  | [200, {'Content-Type' => 'text/html'}, ["Your balance is: #{@machine.money}\n#{err}"]] |
|  | when '/balance' |
|  | [200, {'Content-Type' => 'text/html'}, ["Your balance now: #{@machine.money}"]] |
|  | else |
|  | [404, {'Content-Type' => 'text/html'}, ["Not found\nreq.path = #{req.path}"]] |
|  | end |
|  | end |
|  | end |
|  |  |
|  | app = App.new |
|  |  |
|  | while connection = server.accept |
|  | request = connection.gets |
|  | method, full\_path = request.split(' ') |
|  | path, query = full\_path.split('?') |
|  |  |
|  | status, headers, body = app.call({ |
|  | 'REQUEST\_METHOD' => method, |
|  | 'PATH\_INFO' => path, |
|  | 'QUERY\_STRING' => query |
|  | }) |
|  |  |
|  | connection.print("HTTP/1.1 #{status}\r\n") |
|  |  |
|  | headers.each { |key, value| connection.print("#{key}: #{value}\r\n") } |
|  |  |
|  | connection.print "\r\n" |
|  |  |
|  | body.each { |part| connection.print(part) } |
|  |  |
|  | connection.close |
|  | end |

**Файл http-server.rb:**

|  |
| --- |
| CONST\_START\_BALANCE = 100.0 #Дефолтное значение денег экземпляра класса CashMachine |
|  |  |
|  | class CashMachine |
|  | attr\_reader :money, :my\_deposit |
|  | def initialize |
|  | @@my\_deposit ||= rand(0.0 .. 999.99).floor(2) #Одна сумма для всех экземпляров этого класса. Каждый берет из n(которое для всех общее) |
|  | f\_name = "balance.txt" |
|  | if(File.exists?(f\_name)) |
|  | f = File.open(f\_name, "r+") |
|  | @money = f.read |
|  | if(@money.empty?) |
|  | @money = CONST\_START\_BALANCE |
|  | else |
|  | @money = @money.to\_f |
|  | end |
|  | else |
|  | f = File.open(f\_name, "w") |
|  | end |
|  | f.close |
|  | end |
|  | def get\_deposit |
|  | return @@my\_deposit |
|  | end |
|  | def deposit(new\_d) # Действие при выборе пользователем D |
|  | if ((new\_d <= @@my\_deposit) && (new\_d >= 0)) |
|  | @@my\_deposit = @@my\_deposit - new\_d |
|  | @money += new\_d |
|  | else return false |
|  | end |
|  | f = File.open('balance.txt', "w") |
|  | f.write(@money.to\_s) |
|  | f.close |
|  | return true |
|  | end |
|  |  |
|  | def withdraw(\_money) # Действие при выборе пользователем W |
|  | if ((\_money <= @money) && (\_money > 0)) |
|  | @money -= \_money |
|  | else return false |
|  | end |
|  | f = File.open('balance.txt', "w") |
|  | f.write(@money.to\_s) |
|  | f.close |
|  | return true |
|  | end |
|  |  |
|  | def balance # Действие при выборе пользователем В |
|  | f = File.open('balance.txt', "w") |
|  | f.write(@money.to\_s) |
|  | f.close |
|  | end |
|  |  |
|  | def err(ch) # Действие в случае ошибки |
|  | puts "You input encorrect value: #{ch}\nPlease try again\n" |
|  | end |
|  |  |
|  |  |
|  | def menu |
|  | ch = nil |
|  | loop do |
|  | print "D - deposit; W - withdraw\nB - balance; Q - quit\nEnter: " |
|  | ch = gets.to\_s |
|  | ch = ch[0] |
|  | if ch >= 'a' |
|  | ch = ch.ord - 32 |
|  | ch = ch.chr |
|  | end |
|  |  |
|  | case ch |
|  | when 'D' |
|  | self.deposit |
|  | when 'W' |
|  | self.withdraw |
|  | when 'B' |
|  | self.balance |
|  | when 'Q' |
|  | break |
|  | else self.err(ch) |
|  | end |
|  |  |
|  | if ch == 'Q' |
|  | break |
|  | end |
|  | end |
|  | end |
|  |  |
|  | def init |
|  | f\_name = "balance.txt" |
|  | if(File.exists?(f\_name)) |
|  | f = File.open(f\_name, "r+") |
|  | @money = f.read |
|  | if(@money.empty?) |
|  | @money = CONST\_START\_BALANCE |
|  | else |
|  | @money = @money.to\_f |
|  | end |
|  | else |
|  | f = File.open(f\_name, "w") |
|  | end |
|  | f.close |
|  | self.menu# Результат работы пользователя |
|  | # Обновляем файл |
|  | new\_f = File.open("tmpScrpt2.txt", "w") |
|  | new\_f.write(@money.to\_s) |
|  | new\_f.close |
|  | File.delete(f\_name) |
|  | File.rename("tmpScrpt2.txt", f\_name) |
|  | end |
|  | end |

**Вывод:** в ходе выполнения работы было проведеноознакомление с основами использования API для организации http-сервера в языке Ruby.