

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«МИРЭА - Российский технологический университет»**

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИППО)

**ОТЧЁТ ПО**

**ЛАБОРАТОРНЫМ РАБОТАМ**

**по дисциплине**

«Технологии разработки программного обеспечения»

|  |  |  |
| --- | --- | --- |
| Выполнил студент группы ИКБО-01-15  Принял | | Матюшин Д.С.  Торхов А.Е. |
| Работы выполнены  Работы зачтены | «15» мая 2018г.  « » \_\_\_\_ 2018г. | (*подпись студента)*  *(подпись преподавателя)* |

Москва 2018

Оглавление

[Лабораторная работа №1 4](#_Toc515259997)

[Задание 4](#_Toc515259998)

[Описание работы программы 4](#_Toc515259999)

[Скриншоты работы программы 5](#_Toc515260000)

[Лабораторная работа №2 6](#_Toc515260001)

[Задание 6](#_Toc515260002)

[Описание работы программы 6](#_Toc515260003)

[Скриншоты работы программы 7](#_Toc515260004)

[Лабораторная работа №3 9](#_Toc515260005)

[Задание 9](#_Toc515260006)

[Описание работы программы 9](#_Toc515260007)

[Скриншоты работы программы 11](#_Toc515260008)

[Лабораторная работа №4 15](#_Toc515260009)

[Задача 15](#_Toc515260010)

[Описание работы программы 15](#_Toc515260011)

[Скриншоты работы программы 16](#_Toc515260012)

[Приложение 1. Исходный код программ 18](#_Toc515260013)

[Файл date.java 18](#_Toc515260014)

[Файл dateReg.java 19](#_Toc515260015)

[Файл emailReg.java 19](#_Toc515260016)

[Файл timeReg.java 19](#_Toc515260017)

[Файл Parser.java 20](#_Toc515260018)

[Файл connections.php 21](#_Toc515260019)

[Файл main.css 21](#_Toc515260020)

[Файл ajax.js 22](#_Toc515260021)

[Файл all.php 23](#_Toc515260022)

[Файл searchLayout.php 24](#_Toc515260023)

[Файл search.php 25](#_Toc515260024)

[Файл newLayout.php 26](#_Toc515260025)

[Файл new.php 26](#_Toc515260026)

[Файл changeLayuot.php 27](#_Toc515260027)

[Файл change.php 29](#_Toc515260028)

[Файл replace.php 30](#_Toc515260029)

[Файл deleteLayout.php 31](#_Toc515260030)

[Файл deleteSearch.php 32](#_Toc515260031)

[Файл delete.php 32](#_Toc515260032)

[Файл index.html 33](#_Toc515260033)

# Лабораторная работа №1

## Задание

*Часть №1*

Реализовать программу, которая считывает дату в формате ISO: “YYYY-MM-DDTHH:mm:SSZ", например:

2018-01-12T12:12:43Z

2018-01-12T12:12:43+03:00

Вывести следующие даты:

● Начало следующего дня

● Начало следующего месяца

● Начало недели (в которую входит указанный день)

● Конец недели (в которую входит указанный день)

*Часть №2*

Написать программу которая вводит строку с клавиатуры и проверяет с помощью регулярных выражений соответствует ли данная строка формату:

● Даты

● Даты + времени

● E-Mail

## Описание работы программы

*Часть №1*

В программе создается экземпляр класса Date, с которым будет осуществляться работа. Выставляется заданный в задании формат даты. Работа с датой осуществляется при помощи класса Calendar, а именно:

* Начало следующего дня –

nextDay.add(Calendar.DAY\_OF\_YEAR, 1);

* Начало следующего месяца –

nextMonthDay.add(Calendar.MONTH, 1);  
nextMonthDay.add(Calendar.DAY\_OF\_YEAR, -((nextMonthDay.get(Calendar.DAY\_OF\_MONTH))-1));  
nextMonthDay.set(Calendar.HOUR, 0);  
nextMonthDay.set(Calendar.MINUTE, 0);  
nextMonthDay.set(Calendar.SECOND, 0);

Переводим текущую дату на 1 месяц вперед, выставляем первый день месяца, и время 00:00:00

* Начало текущей недели –

int numDay = weekDays.get(Calendar.DAY\_OF\_WEEK) - 1;  
weekDays.add(Calendar.DAY\_OF\_YEAR, -(numDay-1));

Вычитаем из текущего дня недели количество дней до понедельника

Конец текущей недели –

weekDays.add(Calendar.DAY\_OF\_YEAR, 6);

Прибавляем 6 дней до воскресенья .

*Часть №2*

Для решения задачи используются классы Patern и Matcher. Регулярное выражение задается строкой. Пользователь вводит строку, которая проверяется по исходному регулярному выражению и при совпадении выводит на экран совпадения.

Регулярные выражения:

**Для даты**: [1|2][0-9]{3}-(([0][1-9])|([1][1-2]))-(([0-2][0-9])|[3][0-1])

**Для даты и времени**: [1-9][0-9]{3}[-](([0][1-9])|([1][1-2]))-(([0-2][0-9])|[3][0-1])[ ][t|T][ ](([0|1][0-9])|([2][0-3]))([:]([0-5][0-9])){2}[ ][+|-][0-9]{2}[0-5][0]

**Для e-mail**: ^([a-z0-9\_\\.-]+)@\*([a-z0-9\_\\.-]+)[\\.([a-z\\.]{2,6})$](file:///\\.([a-z\\.%5d%7b2,6%7d)$)

Исходный код программы представлен в Приложении 1. Файлы date.java, dateReg.java, emailReg.java, timeReg.java.

## Скриншоты работы программы

Скриншоты работы представлены на Рисунке 1 (часть №1), Рисунке 2 (а – дата, б- дата и время, в – email).

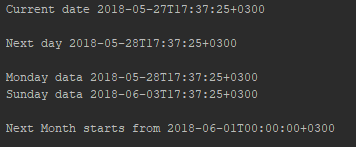


Рисунок 1. Работа с датой



Рисунок 2а. Поиск даты в строке при помощи регулярного выражения



Рисунок 2б. Поиск даты и времени в строке



Рисунок 2в. Поиск email в строке

# Лабораторная работа №2

## Задание

Реализовать программу производящую следующие действия:

1. Получить данные и разобрать с использованием стандартных библиотек.

2. Подсчитать минимальное, максимальное и среднее значение для параметров: current\_speed, temperature, salinity. Если соответствующее значение поля с постфиксом \_qc (например, для current\_speed — current\_speed\_qc) не равно 0, то это означает, что произошел сбой при считывании данных с датчика и это значение не надо учитывать. Данные значения посчитать за один проход массива записей.

3. Проектировать с учетом возможности расширения числа параметров.

4. Вывести в stdout в формате json:

{ "current\_speed": {

"start\_date": "2015-08-15",

"end\_date": "2016-12-05",

"num\_records": 10000,

"min\_current\_speed": 0.0,

"min\_time": "2015-12-10T10:40:00Z",

"max\_current\_speed": 32.81,

"max\_time": "2016-04-02T21:10:00Z",

"avg\_current\_speed": 22.45

},

...

}

Где для каждого из трех параметров выводятся следующие значения: start\_date — начальная дата

end\_date — конечная дата

num\_records — число записей

min\_COLUMN — минимальное значение

min\_time — время в которое достигнуто минимальное значение max\_COLUMN — максимальное значение

max\_time — время в которое достигнуто максимальное значение avg\_COLUMN — среднее значение

COLUMN — имя параметра

## Описание работы программы

Установив путь до файла, программа осуществляет парсинг. Создается JSONObject, который хранит в себе поле table из файла (исходные данные), далее данные разбиваются в разные массивы JSONArray – для названия данных, для типов данных и непосредственно для самих данных. Данные передаются в функцию getData(), которая принимает массивы названий и данных, а также текущий выбор параметра. Значения преобразуются в строку, а затем очищаются от лишних запятых и заносятся в массив:

String row = String.valueOf(rows.get(0));  
String mas[] = row.split(",");

Для минимального и максимального числа задается первое значения массива cтрок. Далее инициализируется переменная, отвечающая за индекс времени:

int dateindx = 0;  
for (int i = 0; i< columnNames.length();i++){  
 row = String.valueOf(columnNames.get(i));  
 if (row.equals("time")){  
 dateindx = i;  
 break;  
 }  
}  
String min\_time=String.valueOf(mas[dateindx])  
 ,max\_time=String.valueOf(mas[dateindx]);

Далее реализуется цикл поиска по всем строкам – обновляем, если необходимо значение минимального и максимального числа, добавляем значение в массив, если исходное условие выполняется –

public static boolean check(String row, JSONArray columnNames, int num) throws JSONException {  
 String field = columnNames.get(num) + "\_qc";  
 int i;  
 for(i=0;i<columnNames.length();i++){  
 if (columnNames.get(i).equals(field))  
 break;  
 }  
 String mass[] = row.split(",");  
 return Float.valueOf(mass[i]) == 0;  
}

После завершения цикла выводим результаты в формате JSON(среднее значение получаем из суммы массива всех подходящих значений на их количество)

Исходный код представлен в Приложении №1. Файл Parser.java

## Скриншоты работы программы

На Рисунке 3 изображен скриншот работы программы.

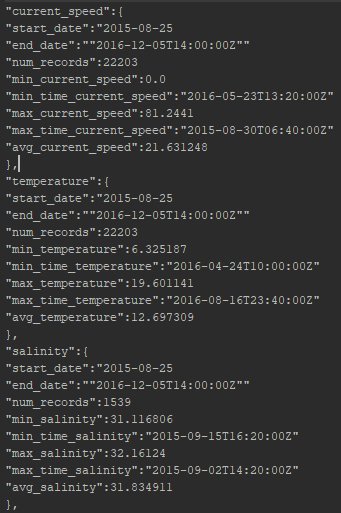


Рисунок 3. Вывод результатов в формате JSON

# Лабораторная работа №3

## Задание

Написать HTTP-сервис, который осуществляет CRUD-интерфейс к некоторому набору данных. В качестве примера, можно взять список пользователей или заметки. Должны быть определены следующие операции:

Получение списка записей

Получение деталей одной записи

Создание записи

Изменение записи

Удаление записи

## Описание работы программы

Для работы программы используется виртуальный сервер Denwer. После установки соединения откроем в браузере страницу test1.ru – это будущий интерфейс для бд. Рассмотрим реализацию каждой операции отдельно.

*Подключение к бд*

**<?php**$host = **"5.23.98.140"**;  
$database = **"hospitals"**;  
$user = **"lirveez"**;  
$password = **"\*\*\*\*\*kbo0115"**;  
**?>**

Данные для подключения к базе, которые используются во всех .php файлах.

*Получение списка записей*

**<?php  
require\_once 'connections.php'**;  
$link=*mysqli\_connect*($host,$user,$password,$database)  
 **or die**(**"Ошибка"**.*mysqli\_error*($link));  
*mysqli\_set\_charset*($link,**'utf8'**);  
*mysqli\_query*($link,**"SET NAMES 'utf8'"**);  
*mysqli\_query*($link,**"SET CHARACTER SET 'utf8'"**);  
*mysqli\_query*($link,**"SET SESSION collation\_connection = 'utf8\_general\_ci'"**);  
$query = **"SELECT \* from doctors"**;  
$result = *mysqli\_query*($link,$query) **or die**(**"Ошибка "**.  
 *mysqli\_error*($link));  
$table = **"<table border=1 width = '600px' align=center>"**;  
$color=**"#FFFFFF"**;  
$table .= **"<tr bgcolor="**.$color.**">"**;  
$table .= **"<td >"**.**"<b>Фамилия</b>"**.**"</td>"**;  
//////////////////////////////////////////////  
$table .= **"<td >"**.**"<b>ID отделения</b>"**.**"</td>"**;  
$table .= **"</tr>"**;  
$k=1;  
**while**($row = *mysqli\_fetch\_object*($result)) {  
 **if**($k%2==0)  
 $color=**"#747AFF"**;  
 **else** $color=**"#6EFF67"**;  
 $k++;  
 $table.=**"<tr bgcolor="**.$color.**">"**;  
 $table .= **"<td >"**.$row->**doctor\_surname**.**"</td>"**;  
//////////////////////////////////////////////////  
 $table .= **"<td >"**.$row->**branch\_ID**.**"</td>"**;  
 $table .= **"</tr>"**;  
}  
$table .= **"</table>"**;  
**echo** $table;  
**?>**

Отправляется запрос на подключение, выставляется штатная кодировка, затем формируется таблица-html, совершается запрос к бд, который затем разбивается на объекты и заносится в таблицу. Итоговая строка передается на html-страницу как таблица.

*Получение деталей одной записи*

**echo "<script>  
 function change(){  
 fill('"**.$str->**doctor\_surname**.**"');  
 $('#surname').text(\""**.$str->**doctor\_surname**.**"\");  
 $('#name').text(\""**.$str->**doctor\_name**.**"\");  
 $('#patronymic').text(\""**.$str->**doctor\_patronymic**.**"\");  
 $('#spec').text(\""**.$str->**doctor\_specialization**.**"\");  
 $('#worktime').text(\""**.$str->**work\_time**.**"\");  
 $('#address').text(\""**.$adress->**branch\_adress**.**"\");  
 }  
 </script>  
<li hover='' onclick=\"change();\"); \">"**.$str->**doctor\_surname**.**"</li>  
 "**;

Аналогичным образом совершается подключение к базе, запрос, который возвращает данные людей, фамилия которых совпадает со строкой, а при нажатии на фамилию запускается скрипт, который передает данные в имеющуюся форму.

*Создание записи*

$query = **"INSERT INTO doctors VALUES ('**$id**','**$surname**',  
'**$firstname**','**$patronomic**',  
'**$spec**','**$skill**','**$exp**','**$diploma**','**$epml**','**$salary**',  
'**$kids**','**$birth**',  
'**$address**','**$work\_time**','**$branch\_ID**')"**;  
$result = *mysqli\_query*($link,$query) **or die**(**"Ошибка "**.  
 *mysqli\_error*($link));

Данные собираются с формы методом POST, а затем добавляются в базу. Id при этом увеличивается на 1 от последней записи в базе.

*Изменение записи*

**echo "<script>  
 function change(){  
 fill('"**.$str->**doctor\_surname**.**"');  
  
 document.getElementById(\"sur\").value = \""**.$str->**doctor\_surname**.**"\";  
 document.getElementById(\"name\").value = \""**.$str->**doctor\_name**.**"\";  
 document.getElementById(\"patr\").value = \""**.$str->**doctor\_patronymic**.**"\";  
 ////////////////////////////////////////////////////////////////  
 document.getElementById(\"id\").value = "**.$str->**doctor\_ID**.**";  
 }  
 </script>  
<li hover='' onclick=\"change();\">"**.$str->**doctor\_surname**.**"</li>  
 "**;

После успешного поиска необходимой информации данные в форме обновляются, далее действия аналогичны действиям при добавлении в бд – методом POST данные передаются в скрипт, затем обновляется строка в бд

$query = **"UPDATE doctors SET  
 doctor\_surname = '**$surname**',  
 doctor\_name ='**$firstname**',  
 doctor\_patronymic ='**$patronomic**',  
 doctor\_specialization ='**$spec**',  
 doctor\_skill\_level ='**$skill**',  
 experience ='**$exp**',  
 diploma ='**$diploma**',  
 employment\_history ='**$epml**',  
 salary ='**$salary**',  
 amount\_of\_kids ='**$kids**',  
 date\_of\_birth ='**$birth**',  
 home\_address ='**$address**',  
 work\_time ='**$work\_time**',  
 branch\_ID ='**$branch\_ID**'  
WHERE doctor\_id ='**$id**' "**;

*Удаление записи*

После поиска записи, строка передается в скрипт, который осуществляет соответствующий запрос в бд и оповещает об успешном удалении

$surname = $\_POST[**'surname'**];  
$query = **"DELETE FROM doctors WHERE doctor\_surname LIKE '**$surname**'"**;  
*mysqli\_query*($link,$query);  
$result = *mysqli\_query*($link,$query) **or die**(**"Ошибка "**.  
 *mysqli\_error*($link));  
**if**($result){  
 **echo "<span> Успешно</span>"**;  
 *mysqli\_close*($link);  
 **exit**();  
}

Исходный код представлен в Приложении 1. Файлы:

Основные: connections.php, main.css, ajax.js,index.html

Вывод всех записей: all.php

Вывод одной записи: searchLayout.php, search.php

Создание записи: newLayout.php, new.php

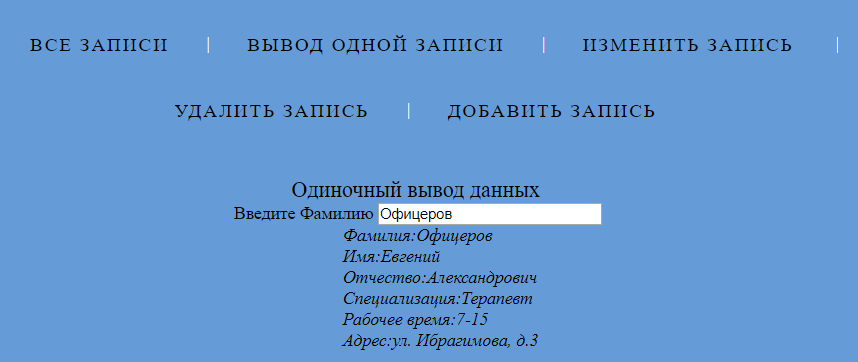
Изменение записи: changeLayuot.php, change.php, replace.php

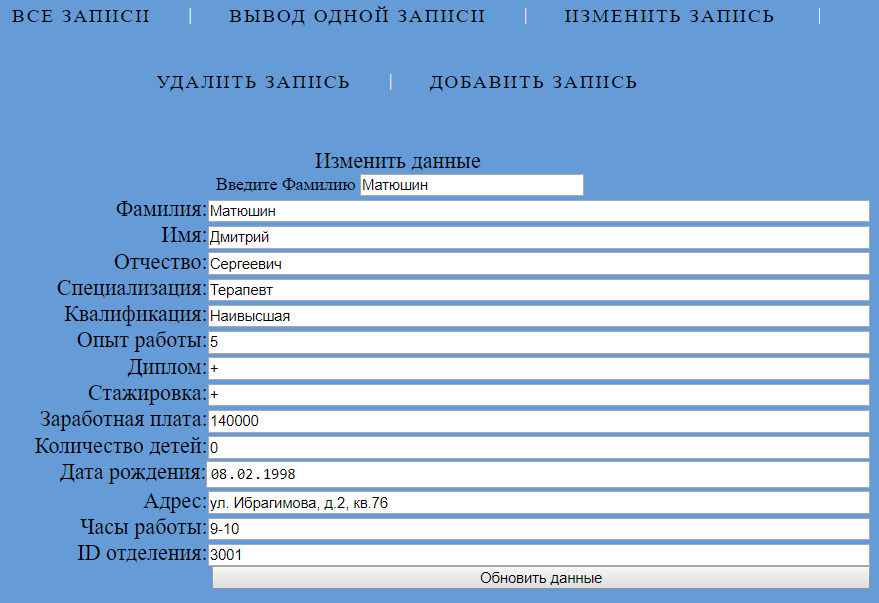
Удаление записи: deleteLayout.php, deleteSearch.php, delete.php

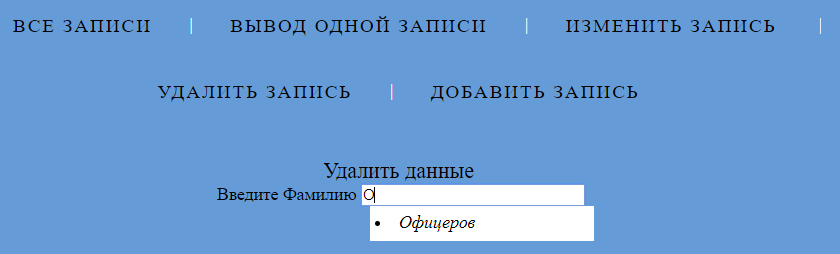
## Скриншоты работы программы

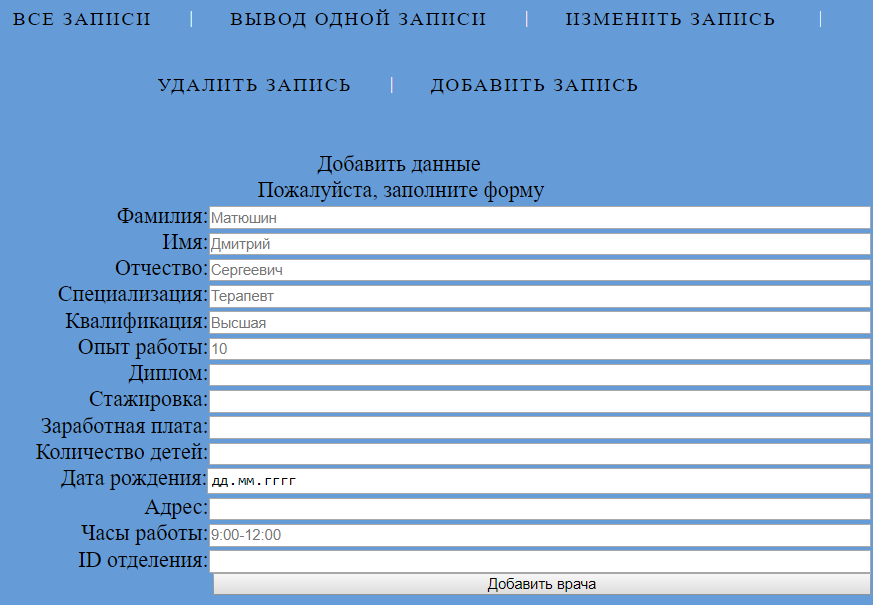
На Рисунке 4 (а – вывод всех записей, б – вывод одной записи, в – создание записи, г – изменение записи, д – удаление записи) изображены скриншоты работы программы.

Рисунок 4а. Вывод всех записей

Рисунок 4б. Вывод одной записи

Рисунок 4в. Изменение данных

Рисунок 4г. Удаление данных

Рисунок 4д. Добавление данных

# Лабораторная работа №4

## Задача

Написать HTTP-сервис, который будет периодически запрашивать API данные по нескольким криптовалютам (можно использовать https://min-api.cryptocompare.com/ ) и отправлять сообщения об изменениях с помощью websockets, либо с помощью http long polling. При подключении к серверу должны быть отправлены текущие значения данных. В качестве интерфейса можно использовать браузерные расширения для работы с websockets.

## Описание работы программы

Для корректной работы программы требуется создать maven-проект. Этот плагин позволит скачивать необходимые библиотеки в автоматическом режиме. Содержание графы dependencies файла pom.xml должно быть примерно следующим:

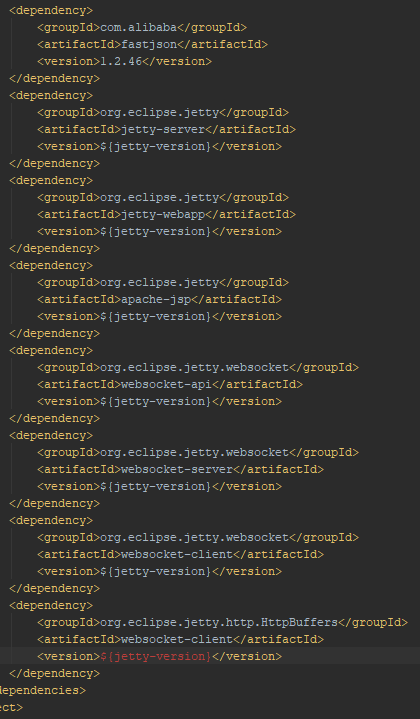


Рисунок 5. Содержимое pom.xml

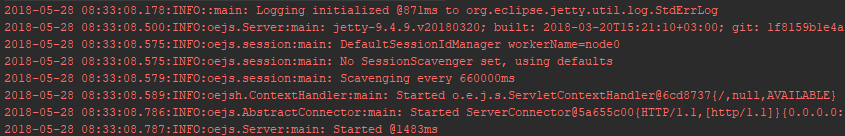
Запуск сервера осуществляется классом Launcher.java – сервер инициализируется при помощи подключаемой библиотеки jetty.server, ему задается определенный порт, в данном случае 159. Далее к серверу привязываются обработчики servlet. Класс MainServlet определяет функцию doGet(), которая вызывается при запросах к стороннему серверу. EventSocket определяет следующие методы –метод, который при загрузке добавляет Listner, метод, который при закрытии удаляет Listner и метод, который передает значения в json-формате на страницу (реализация интерфейса CurrencyListner). Класс CurrencyObserver создает новый поток, который осуществляет свою работу параллельно серверу. В этом классе описывается запрос к серверу с криптовалютой, а также переопределение формата json в более удобный для использования. Если полученные значения отличны от тех, которые есть на странице, то программа осуществит их замену и выведет соответствующее сообщение в консоль.

В описании html-страницы есть скрипт, который работает непосредственно со страницей. В нем прописан адрес локального сервера, а также какие действия необходимо произвести при получение сообщения от сокета.

Исходный код представлен в Приложении 1. Файлы: currency.html, CurrencyListner.java, CurrencyObserver.java, EventServlet.java, EventSocket.java, MainServlet.java, Launcher.java

## Скриншоты работы программы

На Рисунках 5-7 изображены скриншоты работы программы

Рисунок 5. Запуск сервера

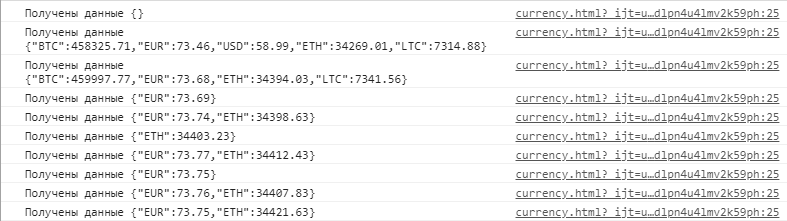


Рисунок 6. Содержимое логов консоли страницы

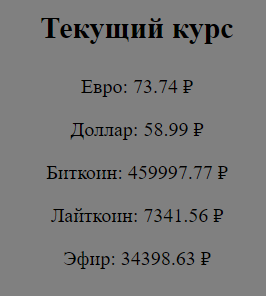


Рисунок 7. Скриншот данных

# Приложение 1. Исходный код программ

## Файл date.java

package Task1;  
import javax.xml.crypto.Data;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.Calendar;  
import java.util.Scanner;  
  
*/\*\*  
 \* Created by Lirveez on 20.02.2018.  
 \*/*public class date {  
 public static void main(String[] args) {  
 // Scanner str = new Scanner(System.in);  
 Date current = new Date();  
 SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ssZ");  
 /\*while(current==null) {  
 try{  
 System.out.println("Enter string");  
 current = format.parse(str.nextLine());  
 }  
 catch (ParseException e){  
 System.out.println("Error.");  
 }  
 }\*/  
  
 String DateToStr = format.format(current);  
 System.*out*.println("Current date "+DateToStr+'\n');  
 Calendar calendar = Calendar.*getInstance*();  
 calendar.setTime(current);  
 Calendar nextDay = (Calendar) calendar.clone();  
 nextDay.add(Calendar.*DAY\_OF\_YEAR*, 1);  
 DateToStr = format.format(nextDay.getTime());  
 System.*out*.println("Next day "+ DateToStr+'\n');  
 Calendar weekDays = (Calendar) calendar.clone();  
 int numDay = weekDays.get(Calendar.*DAY\_OF\_WEEK*) - 1;  
 weekDays.add(Calendar.*DAY\_OF\_YEAR*, -(numDay-1));  
 DateToStr = format.format(weekDays.getTime());  
 System.*out*.println("Monday data "+ DateToStr);  
 weekDays.add(Calendar.*DAY\_OF\_YEAR*, 6);  
 DateToStr = format.format(weekDays.getTime());  
 System.*out*.println("Sunday data "+ DateToStr+'\n');  
 Calendar nextMonthDay = (Calendar) calendar.clone();  
 nextMonthDay.add(Calendar.*MONTH*, 1);  
 nextMonthDay.add(Calendar.*DAY\_OF\_YEAR*, -((nextMonthDay.get(Calendar.*DAY\_OF\_MONTH*))-1));  
 nextMonthDay.set(Calendar.*HOUR*, -12);  
 nextMonthDay.set(Calendar.*MINUTE*, 0);  
 nextMonthDay.set(Calendar.*SECOND*, 0);  
 DateToStr = format.format(nextMonthDay.getTime());  
 System.*out*.println("Next Month starts from "+DateToStr);  
 }  
}

## Файл dateReg.java

package Task1;  
import java.util.Scanner;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
*/\*\*  
 \* Created by Lirveez on 20.02.2018.  
 \*/*public class dateReg {  
 public static void main(String[] args) {  
 String regex = "[1|2][0-9]{3}-(([0][1-9])|([1][1-2]))-(([0-2][0-9])|[3][0-1])";  
 System.*out*.println("Enter string");  
 Scanner str = new Scanner(System.*in*);  
 String string = str.nextLine();  
 Pattern pattern = Pattern.*compile*(regex);  
 Matcher matcher = pattern.matcher(string);  
  
 while (matcher.find()) {  
 System.*out*.println("Found: " + matcher.group(0));  
 }  
 }  
}

## Файл emailReg.java

package Task1;  
  
import java.util.Scanner;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
*/\*\*  
 \* Created by Lirveez on 20.02.2018.  
 \*/*public class emailReg {  
 public static void main(String[] args) {  
 /\* SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd' T 'HH:mm:ss Z");  
 Date current = new Date();  
 String DateToStr = format.format(current);\*/  
 String regex ="[-\\w.]+@([A-z0-9][-A-z0-9]+\\.)+[A-z]{2,4}";  
 System.*out*.println("Enter string");  
 Scanner str = new Scanner(System.*in*);  
 String string = str.nextLine();  
 Pattern pattern = Pattern.*compile*(regex);  
 Matcher matcher = pattern.matcher(string);  
  
 while (matcher.find()) {  
 System.*out*.println("Found: " + matcher.group(0));  
 }  
 }  
}

## Файл timeReg.java

package Task1;  
  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.Scanner;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
*/\*\*  
 \* Created by Lirveez on 20.02.2018.  
 \*/*public class timeReg {  
 public static void main(String[] args) {  
 /\* SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd' T 'HH:mm:ss Z");  
 Date current = new Date();  
 String DateToStr = format.format(current);\*/  
 String regex ="[1-9][0-9]{3}[-](([0][1-9])|([1][1-2]))-(([0-2][0-9])|[3][0-1])[t|T](([0|1][0-9])|([2][0-3]))([:]([0-5][0-9])){2}[+|-][0-9]{2}[:][0-5][0]";  
 System.*out*.println("Enter string");  
 Scanner str = new Scanner(System.*in*);  
 String string = str.nextLine();  
 Pattern pattern = Pattern.*compile*(regex);  
 Matcher matcher = pattern.matcher(string);  
  
 while (matcher.find()) {  
 System.*out*.println("Found: " + matcher.group(0));  
 }  
 }  
}

## Файл Parser.java

package Task2;  
 import org.json.\*;  
 import java.io.\*;  
 import java.util.\*;  
  
*/\*\*  
 \* Created by Lirveez on 30.04.2018.  
 \*/*public class Parser {  
 public static void main(String[] args) throws FileNotFoundException, JSONException {  
 String path = "src/Task2/E05\_aanderaa\_all\_1769\_d432\_5004.json";  
 FileReader file = new FileReader(path);  
 JSONTokener Tokener = new JSONTokener(file);  
 JSONObject obj = new JSONObject(Tokener);  
 JSONObject table = (JSONObject) obj.get("table");  
 JSONArray columnNames = table.getJSONArray("columnNames");  
 JSONArray columnTypes = table.getJSONArray("columnTypes");  
 JSONArray rows = table.getJSONArray("rows");  
 ArrayList<Integer> choice = new ArrayList<Integer>();  
 for (int i = 0; i < columnTypes.length(); i++){  
 if(columnTypes.get(i).toString().equals("float") &&  
 !columnNames.get(i).toString().equals("longitude") &&  
 !columnNames.get(i).toString().equals("latitude") && !columnNames.get(i).toString().equals("depth")){  
 choice.add(i);  
 }  
 }  
 for (int num: choice) {  
 if(columnNames.get(num).toString().equals("current\_speed")||  
 columnNames.get(num).toString().equals("temperature")||  
 columnNames.get(num).toString().equals("salinity")) {  
 System.*out*.println("\"" + columnNames.get(num) + "\"" + ":{");  
 *getData*(rows, columnNames, num);  
 System.*out*.println("},");  
 }  
 }  
  
  
 }  
  
 public static void getData(JSONArray rows, JSONArray columnNames, int num) throws JSONException {  
 float average = 0;  
 String row = String.*valueOf*(rows.get(0));  
 String mas[] = row.split(",");  
 float min= Float.*valueOf*(mas[num]);  
 float max= Float.*valueOf*(mas[num]);  
  
 ArrayList<Float> averageMass = new ArrayList<Float>();  
 ArrayList<String> dates = new ArrayList<>();  
 int dateindx = 0;  
 for (int i = 0; i< columnNames.length();i++){  
 row = String.*valueOf*(columnNames.get(i));  
 if (row.equals("time")){  
 dateindx = i;  
 break;  
 }  
 }  
 String min\_time=String.*valueOf*(mas[dateindx])  
 ,max\_time=String.*valueOf*(mas[dateindx]);  
 for (int i = 0; i < rows.length(); i++){  
 row = String.*valueOf*(rows.get(i));  
 String mass[] = row.split(",");  
 dates.add(String.*valueOf*(mass[dateindx]));  
 if(*check*(row,columnNames,num)) {  
 if (Float.*valueOf*(mass[num])>max){  
 max = Float.*valueOf*(mass[num]);  
 max\_time = String.*valueOf*(mass[dateindx]);  
 }  
 if (Float.*valueOf*(mass[num])<min){  
 min = Float.*valueOf*(mass[num]);  
 min\_time = String.*valueOf*(mass[dateindx]);  
 }  
 averageMass.add(Float.*valueOf*(mass[num]));  
 average = average + averageMass.get(averageMass.size()-1);  
 }  
 }  
 Collections.*sort*(averageMass);  
 Collections.*sort*(dates);  
 System.*out*.println("\"start\_date\":"+dates.get(0).substring(0,11));  
 System.*out*.println("\"end\_date\":"+"\""+dates.get(dates.size()-1)+"\"");  
 System.*out*.println("\"num\_records\":"+averageMass.size());  
 System.*out*.println("\"min\_" +columnNames.get(num) +"\":"+ min);  
 System.*out*.println("\"min\_time\_" +columnNames.get(num) +"\":"+ min\_time);  
 System.*out*.println("\"max\_" +columnNames.get(num) +"\":"+ max);  
 System.*out*.println("\"max\_time\_" +columnNames.get(num) +"\":"+ max\_time);  
 System.*out*.println("\"avg\_" +columnNames.get(num) +"\":"+ average/averageMass.size());  
 }  
  
 public static boolean check(String row, JSONArray columnNames, int num) throws JSONException {  
 String field = columnNames.get(num) + "\_qc";  
 int i;  
 for(i=0;i<columnNames.length();i++){  
 if (columnNames.get(i).equals(field))  
 break;  
 }  
 String mass[] = row.split(",");  
 return Float.*valueOf*(mass[i]) == 0;  
 }  
  
}

## Файл connections.php

**<?php**$host = **"5.23.98.140"**;  
$database = **"hospitals"**;  
$user = **"lirveez"**;  
$password = **"matushinikbo0115"**;  
  
**?>**

## Файл main.css

.**body**{  
 **background**: **blue**;  
}  
**li**{  
 **margin-bottom**: 20**px**;  
}  
**input**{  
 **width**: 70%;  
}  
.**button**{  
 **margin-top**: 20**px**;  
 **width**:30%;  
}  
.**form**{  
 **box-shadow**: 0 0 15**px rgba**(0,0,0,0.5);  
 **border-width**: 1**px**;  
 **border-style**: **solid**;  
 **border-color**: **darkgrey**;  
 **background**: **white**;  
 **width**: 50**vw**;  
 **text-align**: **right**;  
 **display**: **inline-block**;  
}  
.**title**{  
 **text-align**: **center**;  
 **margin-top**: 20**vh**;  
 */\* height: 100vh; \*/*}  
.**lab**{  
 **font-size**: 40**px**;  
}  
.**name**{  
 **font-size**: 35**px**;  
}  
.**db**{  
 **font-size**: 20**px**;  
}  
.**menu-main** {  
 **list-style**: **none**;  
 **margin**: 40**px** 0 5**px**;  
 **padding**: 25**px** 0 5**px**;  
 **text-align**: **center**;  
 **color**: **white**;  
}  
.**menu-main li** {**display**: **inline-block**;}  
.**menu-main li**:**after** {  
 **content**: **"|"**;  
 **color**: **white**;  
 **display**: **inline-block**;  
 **vertical-align**:**top**;  
}  
.**menu-main li**:**last-child**:**after** {**content**: **none**;}  
.**menu-main a** {  
 **text-decoration**: **none**;  
 **letter-spacing**: 2**px**;  
 **position**: **relative**;  
 **padding-bottom**: 20**px**;  
 **margin**: 0 34**px** 0 30**px**;  
 **font-size**: 17**px**;  
 **text-transform**: **uppercase**;  
 **color**: **white**;  
 **display**: **inline-block**;  
 **transition**: **color** .2**s**;  
}  
.**menu-main a**, .**menu-main a**:**visited** {**color**: **black**;}  
.**menu-main a**.**current**, .**menu-main a**:**hover**{**color**: **white**;}  
.**menu-main a**:**before**,  
.**menu-main a**:**after** {  
 **content**: **""**;  
 **position**: **absolute**;  
 **height**: 4**px**;  
 **top**: **auto**;  
 **right**: 50%;  
 **bottom**: -5**px**;  
 **left**: 50%;  
 **background**: **white**;  
 **transition**: .8**s**;  
}  
.**menu-main a**:**hover**:**before**, .**menu-main** .**current**:**before** {**left**: 0;}  
.**menu-main a**:**hover**:**after**, .**menu-main** .**current**:**after** {**right**: 0;}

## Файл ajax.js

**function** allElements(){  
 $.ajax({  
 url: **"../php/all.php"**, *//url страницы (action\_ajax\_form.php)* success: **function**(response) { *//Данные отправлены успешно* result = response;  
 $(**'#result\_form'**).html(result);  
 },  
 error: **function**(response) { *// Данные не отправлены* $(**'#result\_form'**).html(**'Ошибка. Данные не отправлены.'**);  
 }  
 });  
}  
  
**function** *searchElement*(){  
 **$**.**ajax**({  
 **url**: **"../php/searchLayout.php"**, *//url страницы (action\_ajax\_form.php)* success: **function**(response) { *//Данные отправлены успешно* **result** = response;  
 **$**(**'#result\_form'**).html(**result**);  
 },  
 error: **function**(response) { *// Данные не отправлены* **$**(**'#result\_form'**).html(**'Ошибка. Данные не отправлены.'**);  
 }  
 });  
}  
**function** *newElement*(){  
 **$**.**ajax**({  
 **url**: **"../php/newLayout.php"**, *//url страницы (action\_ajax\_form.php)* success: **function**(response) { *//Данные отправлены успешно* **result** = response;  
 **$**(**'#result\_form'**).html(**result**);  
 },  
 error: **function**(response) { *// Данные не отправлены* **$**(**'#result\_form'**).html(**'Ошибка. Данные не отправлены.'**);  
 }  
 });  
}  
**function** *change*(){  
 **$**.**ajax**({  
 **url**: **"../php/change.php"**, *//url страницы (action\_ajax\_form.php)* success: **function**(response) { *//Данные отправлены успешно* **result** = response;  
 **$**(**'#result\_form'**).html(**result**);  
 },  
 error: **function**(response) { *// Данные не отправлены* **$**(**'#result\_form'**).html(**'Ошибка. Данные не отправлены.'**);  
 }  
 });  
}  
**function** *del*(){  
 $.ajax({  
 url: **"../php/delLayout.php"**, *//url страницы (action\_ajax\_form.php)* success: **function**(response) { *//Данные отправлены успешно* result = response;  
 $(**'#result\_form'**).html(result);  
 },  
 error: **function**(response) { *// Данные не отправлены* $(**'#result\_form'**).html(**'Ошибка. Данные не отправлены.'**);  
 }  
 });  
}

## Файл all.php

**<?php  
require\_once 'connections.php'**;  
$link=*mysqli\_connect*($host,$user,$password,$database)  
 **or die**(**"Ошибка"**.*mysqli\_error*($link));  
*mysqli\_set\_charset*($link,**'utf8'**);  
*mysqli\_query*($link,**"SET NAMES 'utf8'"**);  
*mysqli\_query*($link,**"SET CHARACTER SET 'utf8'"**);  
*mysqli\_query*($link,**"SET SESSION collation\_connection = 'utf8\_general\_ci'"**);  
$query = **"SELECT \* from doctors"**;  
$result = *mysqli\_query*($link,$query) **or die**(**"Ошибка "**.  
 *mysqli\_error*($link));  
$table = **"<table border=1 width = '600px' align=center>"**;  
$color=**"#FFFFFF"**;  
$table .= **"<tr bgcolor="**.$color.**">"**;  
$table .= **"<td >"**.**"<b>Фамилия</b>"**.**"</td>"**;  
$table .= **"<td >"**.**"<b>Имя</b>"**.**"</td>"**;  
$table .= **"<td >"**.**"<b>Отчество</b>"**.**"</td>"**;  
$table .= **"<td >"**.**"<b>Специализация</b>"**.**"</td>"**;  
$table .= **"<td >"**.**"<b>Квалификация</b>"**.**"</td>"**;  
$table .= **"<td >"**.**"<b>Опыт</b>"**.**"</td>"**;  
$table .= **"<td >"**.**"<b>Наличие диплома</b>"**.**"</td>"**;  
$table .= **"<td >"**.**"<b>Стажировка</b>"**.**"</td>"**;  
$table .= **"<td >"**.**"<b>Зарплата</b>"**.**"</td>"**;  
$table .= **"<td >"**.**"<b>Количество детей</b>"**.**"</td>"**;  
$table .= **"<td >"**.**"<b>Дата рождения</b>"**.**"</td>"**;  
$table .= **"<td >"**.**"<b>Адрес</b>"**.**"</td>"**;  
$table .= **"<td >"**.**"<b>Время работы</b>"**.**"</td>"**;  
$table .= **"<td >"**.**"<b>ID отделения</b>"**.**"</td>"**;  
$table .= **"</tr>"**;  
$k=1;  
**while**($row = *mysqli\_fetch\_object*($result)) {  
 **if**($k%2==0)  
 $color=**"#747AFF"**;  
 **else** $color=**"#6EFF67"**;  
 $k++;  
 $table.=**"<tr bgcolor="**.$color.**">"**;  
 $table .= **"<td >"**.$row->**doctor\_surname**.**"</td>"**;  
 $table .= **"<td >"**.$row->**doctor\_name**.**"</td>"**;  
 $table .= **"<td >"**.$row->**doctor\_patronymic**.**"</td>"**;  
 $table .= **"<td >"**.$row->**doctor\_specialization**.**"</td>"**;  
 $table .= **"<td >"**.$row->**doctor\_skill\_level**.**"</td>"**;  
 $table .= **"<td >"**.$row->**experience**.**"</td>"**;  
 $table .= **"<td >"**.$row->**diploma**.**"</td>"**;  
 $table .= **"<td >"**.$row->**employment\_history**.**"</td>"**;  
 $table .= **"<td >"**.$row->**salary**.**"</td>"**;  
 $table .= **"<td >"**.$row->**amount\_of\_kids**.**"</td>"**;  
 $table .= **"<td >"**.$row->**date\_of\_birth**.**"</td>"**;  
 $table .= **"<td >"**.$row->**home\_address**.**"</td>"**;  
 $table .= **"<td >"**.$row->**work\_time**.**"</td>"**;  
 $table .= **"<td >"**.$row->**branch\_ID**.**"</td>"**;  
 $table .= **"</tr>"**;  
}  
$table .= **"</table>"**;  
**echo** $table;  
**?>**

## Файл searchLayout.php

**<?php***/\*  
\*  
 \* Created by PhpStorm.  
 \* User: Lirveez  
 \* Date: 11.05.2018  
 \* Time: 18:35  
 \*/***echo "<span class='db'> Одиночный вывод данных</span> <br><style>.suggestionsBox {  
 position: relative;  
 left:40px;  
 width: 200px;  
 background-color: #ffffff;  
 border: 2px solid #ffffff;  
 color: black;  
 }  
.info{  
 text-align: left;  
 padding-left: 42%;  
 font-style: italic;  
 }  
 .suggestionList {  
 margin: 0px;  
 padding: 0px;  
 }  
  
 .suggestionList li {  
 margin: 0px 0px 3px 0px;  
 padding: 3px;  
 cursor: pointer;  
 }  
 .suggestionList li:hover {  
 background-color: white;  
 }</style><script>  
function lookup(inputString) {  
 if(inputString.length == 0) {  
 // Hide the suggestion box.  
 $('#suggestions').hide();  
 } else {  
 $.post(\"../php/search.php\", {queryString: \"\"+inputString+\"\"},  
 function(data){  
 if(data.length >0) {  
 $('#suggestions').show();  
 $('#autoSuggestionsList').html(data);  
  
 }  
 });  
 }  
 } // lookup  
  
function fill(thisValue) {  
 $('#inputString').val(thisValue);  
 setTimeout(\"$('#suggestions').hide();\", 200);  
}  
  
</script><span>Введите Фамилию</span>  
 <input style=\"width: 200px\" type=\"text\" value=\"\"  
 id=\"inputString\" onkeyup=\"lookup(this.value);\" onblur=\"fill();\" /><br>  
  
  
 <div class=\"info\">  
 <div class=\"suggestionsBox\" id=\"suggestions\" style=\"display: none;\">  
 <div class=\"suggestionList\" id=\"autoSuggestionsList\">  
 </div>  
 </div>  
   
 <span >Фамилия:</span><span id=\"surname\"></span><br>  
 <span >Имя:</span><span id=\"name\"></span><br>  
 <span >Отчество:</span><span id=\"patronymic\"></span><br>  
 <span >Специализация:</span><span id=\"spec\"></span><br>  
 <span >Рабочее время:</span><span id=\"worktime\"></span><br>  
 <span >Адрес:</span><span id=\"address\"></span><br>  
 </div>"**;  
**?>**

## Файл search.php

**<?php  
require\_once '../php/connections.php'**;  
$link=*mysqli\_connect*($host,$user,$password,$database)  
**or die**(**"Ошибка"**.*mysqli\_error*($link));  
*mysqli\_set\_charset*($link,**'utf8'**);  
*mysqli\_query*($link,**"SET NAMES 'utf8'"**);  
*mysqli\_query*($link,**"SET CHARACTER SET 'utf8'"**);  
*mysqli\_query*($link,**"SET SESSION collation\_connection = 'utf8\_general\_ci'"**);  
  
**if**(**isset**($\_POST[**'queryString'**])) {  
 $queryString = $\_POST[**'queryString'**];  
 **if**(*strlen*($queryString) > 0) {  
  
 $query = **"SELECT doctor\_surname,doctor\_name,doctor\_patronymic,doctor\_specialization,work\_time,branch\_ID   
FROM doctors  
 WHERE doctor\_surname LIKE '**$queryString**%' LIMIT 10"**;  
 $result = *mysqli\_query*($link,$query);  
 **if**($result) {  
 *// Пока в полученном ресурсе есть результирующая информация проходимся по всему ресурсу вытягивая обьект - результат .* **while** ($str = *mysqli\_fetch\_object*($result)) {  
 $query = **"SELECT branch\_adress FROM adresses WHERE branch\_ID='**$str->**branch\_ID'"**;  
 $res = *mysqli\_query*($link,$query);  
 $adress = *mysqli\_fetch\_object*($res);  
 **echo "<script>  
 function change(){  
 fill('"**.$str->**doctor\_surname**.**"');  
 $('#surname').text(\""**.$str->**doctor\_surname**.**"\");  
 $('#name').text(\""**.$str->**doctor\_name**.**"\");  
 $('#patronymic').text(\""**.$str->**doctor\_patronymic**.**"\");  
 $('#spec').text(\""**.$str->**doctor\_specialization**.**"\");  
 $('#worktime').text(\""**.$str->**work\_time**.**"\");  
 $('#address').text(\""**.$adress->**branch\_adress**.**"\");  
 }  
 </script>  
 <li hover='' onclick=\"change();\"); \">"**.$str->**doctor\_surname**.**"</li>  
 "**;  
 }  
 } **else** {  
 **echo 'ERROR: There was a problem with the query.'**;  
 }  
 } **else** {  
 *// Ничего не делаем.* } *// Это queryString.*} **else** {  
 **echo 'There should be no direct access to this script!'**;  
}  
**?>**

## Файл newLayout.php

**<?php***/\*\*  
 \* Created by PhpStorm.  
 \* User: Lirveez  
 \* Date: 13.05.2018  
 \* Time: 14:55  
 \*/***echo "<span class='db'> Добавить данные</span> <br><form style=\"text-align: right\" action=\"../php/new.php\" method=\"post\">  
 <div style=\"text-align: center\"><span class=\"db\" >  
 Пожалуйста, заполните форму</span></div>  
 <div><span class=\"db\">  
 Фамилия:<input name=\"surname\" placeholder=\"Матюшин\" type=\"text\"><br>  
 </span></div>  
 <span class=\"db\">  
 Имя:<input name=\"firstname\" placeholder=\"Дмитрий\" type=\"text\"><br>  
 </span>  
 <span class=\"db\">  
 Отчество:<input name=\"patronomic\" placeholder=\"Сергеевич\" type=\"text\"><br>  
 </span>  
 <span class=\"db\">  
 Специализация:<input name=\"spec\" placeholder=\"Терапевт\" type=\"text\"><br>  
 </span>  
 <span class=\"db\">  
 Квалификация:<input name=\"skill\" placeholder=\"Высшая\" type=\"text\"><br>  
 </span>  
 <span class=\"db\">  
 Опыт работы:<input name=\"exp\" placeholder=\"10\" type=\"number\"><br>  
 </span>  
 <span class=\"db\">  
 Диплом:<input name=\"diploma\" list=\"bool\"><br>  
 <datalist id=\"bool\">  
 <option value=\"+\">  
 </option>  
 <option value=\"-\"></option>  
 </datalist>  
 </span>  
 <span class=\"db\">  
 Стажировка:<input name=\"empl\" list=\"bool\"><br>  
 </span>  
 <span class=\"db\">  
 Заработная плата:<input name=\"salary\" type=\"number\"  
 min=\"0\"><br>  
 </span>  
 <span class=\"db\">  
 Количество детей:<input name=\"kids\" type=\"number\"  
 min=\"0\"><br>  
 </span>  
 <span class=\"db\">  
 Дата рождения:<input name=\"birth\" type=\"date\"  
 min=\"2018-21-03\"><br>  
 </span>  
 <span class=\"db\">  
 Адрес:<input name=\"address\" type=\"text\"><br>  
 </span>  
 <span class=\"db\">  
 Часы работы:<input name=\"work\_time\" placeholder=\"9:00-12:00\" type=\"text\"><br>  
 </span>  
 <span class=\"db\">  
 ID отделения:<input name=\"branch\_ID\" ><br>  
 </span>  
 <div style=\"text-align: right\"><input type=\"submit\" value=\"Добавить врача\"></div>  
 </form>"**;  
**?>**

## Файл new.php

**<?php***/\*\*  
 \* Created by PhpStorm.  
 \* User: Lirveez  
 \* Date: 11.05.2018  
 \* Time: 16:04  
 \*/***require\_once 'connections.php'**;  
$link=*mysqli\_connect*($host,$user,$password,$database)  
 **or die**(**"Ошибка"**.*mysqli\_error*($link));  
*mysqli\_set\_charset*($link,**'utf8'**);  
*mysqli\_query*($link,**"SET NAMES 'utf8'"**);  
*mysqli\_query*($link,**"SET CHARACTER SET 'utf8'"**);  
*mysqli\_query*($link,**"SET SESSION collation\_connection = 'utf8\_general\_ci'"**);  
$idis = *mysqli\_query*($link,**"Select doctor\_id from doctors"**);  
**while** ($str = *mysqli\_fetch\_object*($idis)) {  
 $id = $str;  
}  
$id = $id + 1;  
$firstname = $\_POST[**'firstname'**];  
$surname = $\_POST[**'surname'**];  
$patronomic = $\_POST[**'patronomic'**];  
$spec = $\_POST[**'spec'**];  
$skill = $\_POST[**'skill'**];  
$exp = $\_POST[**'exp'**];  
$diploma = $\_POST[**'diploma'**];  
$epml = $\_POST[**'empl'**];  
$salary = $\_POST[**'salary'**];  
$kids = $\_POST[**'kids'**];  
$birth = $\_POST[**'birth'**];  
$address = $\_POST[**'address'**];  
$work\_time = $\_POST[**'work\_time'**];  
$branch\_ID = $\_POST[**'branch\_ID'**];  
$query = **"INSERT INTO doctors VALUES ('**$id**','**$surname**',  
'**$firstname**','**$patronomic**',  
'**$spec**','**$skill**','**$exp**','**$diploma**','**$epml**','**$salary**',  
'**$kids**','**$birth**',  
'**$address**','**$work\_time**','**$branch\_ID**')"**;  
$result = *mysqli\_query*($link,$query) **or die**(**"Ошибка "**.  
 *mysqli\_error*($link));  
**if**($result){  
 **echo "<script>** *alert***('Данные успешно добавлены');</script>"**;  
 **echo "<script> *location*.replace('../index.html');</script>"**;  
 *mysqli\_close*($link);  
 **exit**();  
}  
**echo "<script>** *alert***('Ошибка');</script>"**;  
**echo "<script> *location*.replace('../index.html');</script>"**;  
*mysqli\_close*($link);  
**?>**

## Файл changeLayuot.php

**<?php***/\*\*  
 \* Created by PhpStorm.  
 \* User: Lirveez  
 \* Date: 13.05.2018  
 \* Time: 15:20  
 \*/***echo "<span class='db'> Изменить данные</span> <br><style>.suggestionsBox {  
 position: relative;  
 left:40px;  
 width: 200px;  
 background-color: #ffffff;  
 border: 2px solid #ffffff;  
 color: black;  
 }  
.info{  
 text-align: left;  
 padding-left: 42%;  
 font-style: italic;  
 }  
 .suggestionList {  
 margin: 0px;  
 padding: 0px;  
 }  
  
 .suggestionList li {  
 margin: 0px 0px 3px 0px;  
 padding: 3px;  
 cursor: pointer;  
 }  
  
 .suggestionList li:hover {  
 background-color: white;  
 }</style><script>  
function lookup(inputString) {  
 if(inputString.length == 0) {  
 // Hide the suggestion box.  
 $('#suggestions').hide();  
 } else {  
 $.post(\"../php/change.php\", {queryString: \"\"+inputString+\"\"},  
 function(data){  
 if(data.length >0) {  
 $('#suggestions').show();  
 $('#autoSuggestionsList').html(data);  
  
 }  
 });  
 }  
 } // lookup  
  
function fill(thisValue) {  
 $('#inputString').val(thisValue);  
 setTimeout(\"$('#suggestions') . hide();\", 200);  
}  
  
</script><span>Введите Фамилию</span>  
 <input style=\"width: 200px\" type=\"text\" value=\"\"  
 id=\"inputString\" onkeyup=\"lookup(this.value);\" onblur=\"fill();\" /><br>  
  
  
 <div class=\"info\">  
 <div class=\"suggestionsBox\" id=\"suggestions\" style=\"display: none;\">  
 <div class=\"suggestionList\" id=\"autoSuggestionsList\">  
 </div>  
 </div>  
   
   
 </div>  
 <form style=\"text-align: right\" action=\"../php/replace.php\" method=\"post\">  
 <div style=\"text-align: center\"></div>  
 <div><span class=\"db\">  
 Фамилия:<input id=\"sur\" name=\"surname\" placeholder=\"Матюшин\" type=\"text\"><br>  
 </span></div>  
 <span class=\"db\">  
 Имя:<input id=\"name\" name=\"firstname\" placeholder=\"Дмитрий\" type=\"text\"><br>  
 </span>   
 <span class=\"db\">  
 Отчество:<input id=\"patr\" name=\"patronomic\" placeholder=\"Сергеевич\" type=\"text\"><br>  
 </span>  
 <span class=\"db\">  
 Специализация:<input id=\"spec\" name=\"spec\" placeholder=\"Терапевт\" type=\"text\"><br>  
 </span>  
 <span class=\"db\">  
 Квалификация:<input id=\"skill\" name=\"skill\" placeholder=\"Высшая\" type=\"text\"><br>  
 </span>  
 <span class=\"db\">  
 Опыт работы:<input id=\"work\" name=\"exp\" placeholder=\"10\" type=\"number\"><br>  
 </span>  
 <span class=\"db\">  
 Диплом:<input id=\"diploma\" name=\"diploma\" list=\"bool\"><br>  
 <datalist id=\"bool\">  
 <option value=\"+\">  
 </option>  
 <option value=\"-\"></option>  
 </datalist>  
 </span>  
 <span class=\"db\">  
 Стажировка:<input id=\"empl\" name=\"empl\" list=\"bool\"><br>  
 </span>  
 <span class=\"db\">  
 Заработная плата:<input id=\"salary\" name=\"salary\" type=\"number\"  
 min=\"0\"><br>  
 </span>  
 <span class=\"db\">  
 Количество детей:<input id=\"kids\" name=\"kids\" type=\"number\"  
 min=\"0\"><br>  
 </span>  
 <span class=\"db\">  
 Дата рождения:<input id=\"dateb\" name=\"birth\" type=\"date\"  
 min=\"2018-21-03\"><br>  
 </span>  
 <span class=\"db\">  
 Адрес:<input id=\"home\" name=\"address\" type=\"text\"><br>  
 </span>  
 <span class=\"db\">  
 Часы работы:<input id=\"worktime\" name=\"work\_time\" placeholder=\"9-12\" type=\"text\"><br>  
 </span>  
 <span class=\"db\">  
 ID отделения:<input id=\"branchid\" name=\"branch\_ID\" ><br>  
 </span>  
 <div style=\"text-align: right\"><input type=\"submit\" value=\"Обновить данные\">  
 </div>  
 <input style=\"visibility: hidden\" id=\"id\" name=\"doctor\_id\" type=\"number\">  
 </form>"**;  
**?>**

## Файл change.php

**<?php***/\*\*  
 \* Created by PhpStorm.  
 \* User: Lirveez  
 \* Date: 11.05.2018  
 \* Time: 16:04  
 \*/***require\_once '../php/connections.php'**;  
$link=*mysqli\_connect*($host,$user,$password,$database)  
**or die**(**"Ошибка"**.*mysqli\_error*($link));  
*mysqli\_set\_charset*($link,**'utf8'**);  
*mysqli\_query*($link,**"SET NAMES 'utf8'"**);  
*mysqli\_query*($link,**"SET CHARACTER SET 'utf8'"**);  
*mysqli\_query*($link,**"SET SESSION collation\_connection = 'utf8\_general\_ci'"**);  
  
**if**(**isset**($\_POST[**'queryString'**])) {  
 $queryString = $\_POST[**'queryString'**];  
 **if**(*strlen*($queryString) > 0) {  
  
 $query = **"SELECT** *\** **FROM doctors  
 WHERE doctor\_surname LIKE '**$queryString**%' LIMIT 10"**;  
 $result = *mysqli\_query*($link,$query);  
 **if**($result) {  
 **while** ($str = *mysqli\_fetch\_object*($result)) {  
 **echo "<script>  
 function** *change***(){  
 fill('"**.$str->**doctor\_surname**.**"');  
   
 *document*.getElementById(\"sur\").value = \""**.$str->**doctor\_surname**.**"\";  
 *document*.getElementById(\"name\").value = \""**.$str->**doctor\_name**.**"\";  
 *document*.getElementById(\"patr\").value = \""**.$str->**doctor\_patronymic**.**"\";  
 *document*.getElementById(\"spec\").value = \""**.$str->**doctor\_specialization**.**"\";  
 *document*.getElementById(\"skill\").value = \""**.$str->**doctor\_skill\_level**.**"\";  
 *document*.getElementById(\"work\").value = \""**.$str->**experience**.**"\";  
 if ('"**.$str->**diploma**.**"'=='+'){  
 *document*.getElementById(\"diploma\").value = \"+\";  
 }  
 else   
 *document*.getElementById(\"diploma\").value = \"-\";  
 if ('"**.$str->**employment\_history**.**"'=='+'){  
 *document*.getElementById(\"empl\").value = \"+\";  
 }  
 else  
 *document*.getElementById(\"empl\").value = \"-\";  
 *document*.getElementById(\"salary\").value = "**.$str->**salary**.**";  
 *document*.getElementById(\"kids\").value = \""**.$str->**amount\_of\_kids**.**"\";  
 *document*.getElementById(\"dateb\").value = \""**.$str->**date\_of\_birth**.**"\";  
 *document*.getElementById(\"home\").value = \""**.$str->**home\_address**.**"\";  
 *document*.getElementById(\"worktime\").value = \""**.$str->**work\_time**.**"\";  
 *document*.getElementById(\"branchid\").value = \""**.$str->**branch\_ID**.**"\";  
 *document*.getElementById(\"id\").value = "**.$str->**doctor\_ID**.**";  
 }  
 </script>  
 <li hover='' onclick=\"***change***();\">"**.$str->**doctor\_surname**.**"</li>  
 "**;  
 }  
 } **else** {  
 **echo 'ERROR: There was a problem with the query.'**;  
 }  
 } **else** {  
 *// Ничего не делаем.* } *// Это queryString.*} **else** {  
 **echo 'There should be no direct access to this script!'**;  
}  
**?>**

## Файл replace.php

**<?php***/\*\*  
 \* Created by PhpStorm.  
 \* User: Lirveez  
 \* Date: 11.05.2018  
 \* Time: 16:04  
 \*/***require\_once 'connections.php'**;  
$link=*mysqli\_connect*($host,$user,$password,$database)  
**or die**(**"Ошибка"**.*mysqli\_error*($link));  
*mysqli\_set\_charset*($link,**'utf8'**);  
*mysqli\_query*($link,**"SET NAMES 'utf8'"**);  
*mysqli\_query*($link,**"SET CHARACTER SET 'utf8'"**);  
*mysqli\_query*($link,**"SET SESSION collation\_connection = 'utf8\_general\_ci'"**);  
  
$firstname = $\_POST[**'firstname'**];  
$surname = $\_POST[**'surname'**];  
$patronomic = $\_POST[**'patronomic'**];  
$spec = $\_POST[**'spec'**];  
$skill = $\_POST[**'skill'**];  
$exp = $\_POST[**'exp'**];  
$diploma = $\_POST[**'diploma'**];  
$epml = $\_POST[**'empl'**];  
$salary = $\_POST[**'salary'**];  
$kids = $\_POST[**'kids'**];  
$birth = $\_POST[**'birth'**];  
$address = $\_POST[**'address'**];  
$work\_time = $\_POST[**'work\_time'**];  
$branch\_ID = $\_POST[**'branch\_ID'**];  
$id = $\_POST[**'doctor\_id'**];  
$query = **"UPDATE doctors SET  
 doctor\_surname = '**$surname**',  
 doctor\_name ='**$firstname**',  
 doctor\_patronymic ='**$patronomic**',  
 doctor\_specialization ='**$spec**',  
 doctor\_skill\_level ='**$skill**',  
 experience ='**$exp**',  
 diploma ='**$diploma**',  
 employment\_history ='**$epml**',  
 salary ='**$salary**',  
 amount\_of\_kids ='**$kids**',  
 date\_of\_birth ='**$birth**',  
 home\_address ='**$address**',  
 work\_time ='**$work\_time**',  
 branch\_ID ='**$branch\_ID**'  
WHERE doctor\_id ='**$id**' "**;  
$result = *mysqli\_query*($link,$query) **or die**(**"Ошибка "**.  
 *mysqli\_error*($link));  
**if**($result){  
 **echo "<script>** *alert***('Данные успешно обновлены');</script>"**;  
 **echo "<script> *location*.replace('../index.html');</script>"**;  
 *mysqli\_close*($link);  
 **exit**();  
}  
*mysqli\_close*($link);  
**?>**

## Файл deleteLayout.php

**<?php***/\*  
\*  
 \* Created by PhpStorm.  
 \* User: Lirveez  
 \* Date: 11.05.2018  
 \* Time: 18:35  
 \*/***echo "<span class='db'> Удалить данные</span> <br><style>.suggestionsBox {  
 position: relative;  
 left:40px;  
 width: 200px;  
 background-color: #ffffff;  
 border: 2px solid #ffffff;  
 color: black;  
 }  
.info{  
 text-align: left;  
 padding-left: 42%;  
 font-style: italic;  
 }  
 .suggestionList {  
 margin: 0px;  
 padding: 0px;  
 }  
  
 .suggestionList li {  
 margin: 0px 0px 3px 0px;  
 padding: 3px;  
 cursor: pointer;  
 }  
  
 .suggestionList li:hover {  
 background-color: white;  
 }</style><script>  
function** *lookup***(inputString) {  
 if(inputString.length == 0) {  
 // Hide the suggestion box.  
 $('#suggestions').hide();  
 } else {  
 $.post(\"../php/deletesearch.php\", {queryString: \"\"+inputString+\"\"},  
 function(data){  
 if(data.length >0) {  
 $('#suggestions').show();  
 $('#autoSuggestionsList').html(data);  
  
 }  
 });  
 }  
 }   
  
function** *fill***(thisValue) {  
 $('#inputString').val(thisValue);** *setTimeout***(\"$('#suggestions').hide();\", 200);  
}  
  
</script><span>Введите Фамилию</span>  
 <input style=\"width: 200px\" type=\"text\" value=\"\"  
 id=\"inputString\" onkeyup=\"***lookup***(this.value);\" onblur=\"***fill***();\" /><br>  
  
  
 <div class=\"info\">  
 <div class=\"suggestionsBox\" id=\"suggestions\" style=\"display: none;\">  
 <div class=\"suggestionList\" id=\"autoSuggestionsList\">  
 </div>  
 </div>  
   
 </div>"**;  
**?>**

## Файл deleteSearch.php

**<?php  
require\_once '../php/connections.php'**;  
$link=*mysqli\_connect*($host,$user,$password,$database)  
**or die**(**"Ошибка"**.*mysqli\_error*($link));  
*mysqli\_set\_charset*($link,**'utf8'**);  
*mysqli\_query*($link,**"SET NAMES 'utf8'"**);  
*mysqli\_query*($link,**"SET CHARACTER SET 'utf8'"**);  
*mysqli\_query*($link,**"SET SESSION collation\_connection = 'utf8\_general\_ci'"**);  
  
**if**(**isset**($\_POST[**'queryString'**])) {  
 $queryString = $\_POST[**'queryString'**];  
 **if**(*strlen*($queryString) > 0) {  
  
 $query = **"SELECT doctor\_surname,doctor\_name,doctor\_patronymic,doctor\_specialization,work\_time,branch\_ID   
FROM doctors  
 WHERE doctor\_surname LIKE '**$queryString**%' LIMIT 10"**;  
 $result = *mysqli\_query*($link,$query);  
 **if**($result) {  
 *// Пока в полученном ресурсе есть результирующая информация проходимся по всему ресурсу вытягивая обьект - результат .* **while** ($str = *mysqli\_fetch\_object*($result)) {  
 **echo "<script>  
 function** *change***(){  
 fill('"**.$str->**doctor\_surname**.**"');  
 $.ajax({  
 url: \"../php/delete.php\", //url страницы (action\_ajax\_form.php)  
 data: \"surname="**.$str->**doctor\_surname**.**"\",  
 type: \"POST\",  
 success: function(response) { //Данные отправлены успешно  
 result = response;  
 $('#resultd').html(result);  
 },  
 error: function(response) { // Данные не отправлены  
 $('#resultd').html('Ошибка. Данные не отправлены.');  
 }  
 });** *alert***(\"Данные удалены\");  
 }  
 </script>  
 <li hover='' onclick=\"***change***();\">"**.$str->**doctor\_surname**.**"</li>  
 <div id=\"resultd\"></div>  
 "**;  
 }  
  
 } **else** {  
 **echo 'ERROR: There was a problem with the query.'**;  
 }  
 } **else** {  
 *// Ничего не делаем.* } *// Это queryString.*} **else** {  
 **echo 'There should be no direct access to this script!'**;  
}  
**?>**

## Файл delete.php

**<?php***/\*\*  
 \* Created by PhpStorm.  
 \* User: Lirveez  
 \* Date: 11.05.2018  
 \* Time: 16:04  
 \*/***require\_once 'connections.php'**;  
$link=*mysqli\_connect*($host,$user,$password,$database)  
**or die**(**"Ошибка"**.*mysqli\_error*($link));  
*mysqli\_set\_charset*($link,**'utf8'**);  
*mysqli\_query*($link,**"SET NAMES 'utf8'"**);  
*mysqli\_query*($link,**"SET CHARACTER SET 'utf8'"**);  
*mysqli\_query*($link,**"SET SESSION collation\_connection = 'utf8\_general\_ci'"**);  
$idis = *mysqli\_query*($link,**"Select doctor\_id from doctors"**);  
  
$surname = $\_POST[**'surname'**];  
$query = **"DELETE FROM doctors WHERE doctor\_surname LIKE '**$surname**'"**;  
*mysqli\_query*($link,$query);  
$result = *mysqli\_query*($link,$query) **or die**(**"Ошибка "**.  
 *mysqli\_error*($link));  
**if**($result){  
 **echo "<span> Успешно</span>"**;  
 *mysqli\_close*($link);  
 **exit**();  
}  
**echo "<script>** *alert***('Ошибка');</script>"**;  
**echo "<script> *location*.replace('../index.html');</script>"**;  
*mysqli\_close*($link);  
**?>**

## Файл index.html

<!DOCTYPE **html**>  
<**html lang="ru"**>  
<**head**>  
  
 <**meta charset="UTF-8"**>  
 <**style**>  
  
 **li**{  
 **margin-bottom**: 20**px**;  
 }  
  
 **li**:**hover**{  
 **cursor**: **pointer**;  
 }  
 **body**{  
 **background**: **blue**;  
 }  
 **#result\_form**{  
 **text-align**: **center**;  
 }  
 </**style**>  
  
 <**title**>Матюшин Дмитрий</**title**><**meta http-equiv="X-UA-Compatible" content="IE=edge"**>  
 <**link rel="stylesheet" href="css/main.css"**>  
 <**link rel="stylesheet" href="css/jquery.modal.css"**>  
 <**script language="JavaScript" type="text/javascript"  
 src="https://ajax.googleapis.com/ajax/libs/jquery/2.0.3/jquery.min.js"**>  
 </**script**>  
 <**script src="js/ajax.js"**></**script**>  
</**head**>  
  
<**body style="background**: **#659CD8"**>  
<**ul class="menu-main"**>  
 <**li**><**a onclick="***allElements*()**"**>Все записи</**a**></**li**>  
 <**li**><**a onclick="***searchElement*()**"**>Вывод одной записи</**a**></**li**>  
 <**li**><**a onclick="***changeElement*()**"** >Изменить запись</**a**>  
 <**li**><**a onclick="***del*()**"** >Удалить запись</**a**></**li**>  
 <**li**><**a onclick="***newElement*()**"** >Добавить запись</**a**>  
  
</**ul**>  
  
 <**div id="result\_form"**>  
  
 </**div**>  
</**body**>  
</**html**>

## Файл currency.html

<meta http-equiv="Content-Type" content="text/html;charset=UTF8">  
<html>  
<head>  
 <title>Курс валют</title>  
 <style>  
 .body{  
 background: grey;}  
 .txt{  
 margin-top: 30%;  
 text-align: center;  
  
 }  
 .title{  
 font-size: 30px;  
 font-weight: bold;  
 }  
 p{  
 font-size: 20px;  
 }  
 </style>  
 <script>  
 const socket = new WebSocket("ws://127.0.0.1:159/event/");  
 const currencies = {'BTC':'Биткоин','ETH':'Эфир','EUR':'Евро','LTC':'Лайткоин','USD':'Доллар'};  
 socket.onmessage = function(event) {  
 console.log("Получены данные " + event.data);  
 var data = JSON.parse(event.data);  
 for (val in data)  
 {  
 document.getElementById(val).innerHTML = currencies[val]+": "+data[val]+" ₽";  
 }  
 };  
 </script>  
</head>  
<body class="body">  
<div class="txt">  
 <p class="title">Текущий курс</p>  
 <p id = 'EUR'>Евро: </p>  
 <p id = 'USD'>Доллар: </p>  
 <p id = 'BTC'>Биткоин: </p>  
 <p id = 'LTC'>Лайткоин: </p>  
 <p id = 'ETH'>Эфир: </p>  
</div>  
  
</body>  
</html>

## Файл CurrencyListner.java

import java.util.Map;  
  
public interface CurrencyListener {  
 void onChangeCurrencies(Map<String, Double> map);  
}

## Файл CurrencyObserver.java

import com.alibaba.fastjson.JSON;  
import com.alibaba.fastjson.JSONObject;  
  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStreamReader;  
import java.net.HttpURLConnection;  
import java.net.URL;  
import java.util.\*;  
import java.util.stream.Collectors;  
  
public class CurrencyObserver implements Runnable {  
 public static final int *DELAY* = 5000;  
 private static final CurrencyObserver *instance* = new CurrencyObserver();  
  
 private final List<CurrencyListener> listeners = Collections.*synchronizedList*(new LinkedList<>());  
 private final Map<String, Double> currencies = new TreeMap<>();  
 private CurrencyObserver()  
 {  
 Thread t = new Thread(this);  
 t.start();  
 }  
 public void addListener(CurrencyListener l)  
 {  
 this.listeners.add(l);  
 l.onChangeCurrencies(currencies);  
 }  
 public void removeListener(CurrencyListener l)  
 {  
 this.listeners.remove(l);  
 }  
  
  
 private String getCurrencies() throws IOException {  
 String url = "https://min-api.cryptocompare.com/data/pricemulti?fsyms=BTC,ETH,EUR,USD,LTC&tsyms=RUR";  
  
 URL obj = new URL(url);  
 HttpURLConnection connection = (HttpURLConnection) obj.openConnection();  
  
 connection.setRequestMethod("GET");  
  
 BufferedReader in = new BufferedReader(new InputStreamReader(connection.getInputStream()));  
 String inputLine;  
 StringBuilder response = new StringBuilder();  
  
 while ((inputLine = in.readLine()) != null) {  
 response.append(inputLine);  
 }  
  
 in.close();  
 return response.toString();  
 }  
 @Override  
 public void run() {  
 while (true)  
 {  
 try {  
 String json = getCurrencies();  
 Map<?, ?> JSONMap = JSON.*parseObject*(json, Map.class);  
 Map<String, Double> map =  
 JSONMap.entrySet().stream().map(entry ->  
 {  
 String key = entry.getKey().toString();  
 Double value = ((JSONObject) entry.getValue()).getDoubleValue("RUR");  
 return new AbstractMap.SimpleEntry<>(key, value);  
 }).filter(e -> !e.getValue().equals(currencies.get(e.getKey()))).  
 collect(Collectors.*toMap*(Map.Entry::getKey, Map.Entry::getValue));  
 if (!map.isEmpty()) {  
 listeners.forEach(currencyListener -> currencyListener.onChangeCurrencies(map));  
 map.forEach(currencies::put);  
 }  
  
 Thread.*sleep*(*DELAY*);  
 } catch (IOException|InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 public static CurrencyObserver getInstance() {  
 return *instance*;  
 }  
}

## Файл EventServlet.java

import org.eclipse.jetty.websocket.servlet.WebSocketServlet;  
import org.eclipse.jetty.websocket.servlet.WebSocketServletFactory;  
  
public class EventServlet extends WebSocketServlet {  
 @Override  
 public void configure(WebSocketServletFactory webSocketServletFactory) {  
 webSocketServletFactory.register(EventSocket.class);  
 }  
}

## Файл EventSocket.java

import com.alibaba.fastjson.JSON;  
import org.eclipse.jetty.websocket.api.Session;  
import org.eclipse.jetty.websocket.api.WebSocketAdapter;  
  
import java.io.IOException;  
import java.util.Map;  
  
public class EventSocket extends WebSocketAdapter implements CurrencyListener {  
  
 @Override  
 public void onWebSocketConnect(Session session) {  
 super.onWebSocketConnect(session);  
 CurrencyObserver.*getInstance*().addListener(this);  
 }  
  
 @Override  
 public void onWebSocketClose(int statusCode, String reason) {  
 super.onWebSocketClose(statusCode, reason);  
 CurrencyObserver.*getInstance*().removeListener(this);  
 }  
  
 @Override  
 public void onChangeCurrencies(Map<String, Double> map) {  
 try {  
 this.getSession().getRemote().sendString(JSON.*toJSONString*(map));  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

## Файл MainServlet.java

import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.FileInputStream;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.OutputStream;  
import java.util.Scanner;  
  
public class MainServlet extends HttpServlet {  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 resp.setContentType("text/html");  
 resp.setStatus(HttpServletResponse.*SC\_OK*);  
 InputStream stream = new FileInputStream("src/main/java/currency.html");  
 *copyStream*(stream, resp.getOutputStream());  
 }  
 public static void copyStream(InputStream input, OutputStream output)  
 throws IOException  
 {  
 StringBuilder builder = new StringBuilder();  
 Scanner scanner = new Scanner(input);  
 while (scanner.hasNextLine())  
 {  
 builder.append(scanner.nextLine());  
 builder.append("\n");  
 }  
 output.write(builder.toString().getBytes());  
 }  
}

## Файл Launcher.java

import org.eclipse.jetty.server.Server;  
import org.eclipse.jetty.server.ServerConnector;  
import org.eclipse.jetty.servlet.ServletContextHandler;  
public class Launcher {  
  
 public static void main(String[] args) throws Exception {  
 Server server = new Server();  
 ServerConnector connector = new ServerConnector(server);  
 connector.setPort(159);  
 server.addConnector(connector);  
 ServletContextHandler contextHandler = new ServletContextHandler(ServletContextHandler.*SESSIONS*);  
 contextHandler.setContextPath("/");  
 server.setHandler(contextHandler);  
 contextHandler.addServlet(EventServlet.class,"/event/");  
 contextHandler.addServlet(MainServlet.class, "/");  
 server.start();  
  
 server.join();  
 }  
}