Яндекс Контест

Пробный контест Архив соревнований Настройки компиляторов Значения ошибок Команды

Контест для Weekend offer backend

② 30 май 2021, 12:05:21 старт: 30 май 2021, 11:05:30 финиш: 30 май 2021, 13:05:30 до финиша: 00:59:49

Объявления жюри Завершить

А. Камни и украшения. [0

перекладывать... [1 балл]

D. Сортировка чисел [1

Е. Ограничение запросов

баллов]

балл] (1)

балл]

[2 балла] (2)

(1)

В. Разбиение на

интервалы дат [1

С. Опять JSON'ы

Задачи Посылки

С. Опять JSON'ы перекладывать... [1 балл]



Всем хотя бы раз в жизни приходилось перекладывать JSON. Вот и для нового проекта под названием «Единое хранилище» необходимо переложить магазинные фиды. Для размещения на Яндекс.Маркете магазины передают товары из своего ассортимент в JSON-файлах. Одно товарное предложение описывается так:

```
"offer_id": <string>,
"market_sku": <int>,
"price": <int>
```

где $offer_id$ — уникальный среди всех фидов магазина идентификатор предложения, $market_sku$ — идентификатор товара на Яндекс.Маркете, price — стоимость товара.

Весь фид магазина представляет собой JSON и выглядит так: {
 "offers": [<offer>, <offer>, ...]

Вас попросили написать программу, которая объединит все фиды одного магазина в единый фид и отсортирует товары в порядке неубывания их стоимостей, а при их равенстве — по $offer_id$.

Формат ввода

В первой строке входных данных содержится целое число n — количество фидов магазина ($1 \leq n \leq 200$). Следующие n строк содержат по одному магазинному фиду на строку. Гарантируется, что строка — валидный JSON и удовлетворяет формату фида. В одном фиде не больше 200 товарных предложений. $offer_id$ состоит из строчных и заглавных букв латинского алфавита и цифр, $1 \leq |offer_id| \leq 10$, $1 \leq market_sku \leq 2^{31} - 1$, $1 \leq price \leq 10^6$.

Формат вывода

Выходной поток должен содержать один JSON-документ, удовлетворяющий формату товарного фида. Количество строк в выходном файле и табуляция не имеют значения.

Пример

Примечания

Для решений на языке **python** доступны все стандартные библиотеки, включая json.

Для решений на языке **Java** доступна библиотека json-simple версии 1.1.1. Соответствующие import'ы могут выглядеть так:

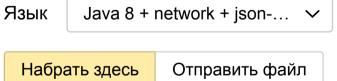
```
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;
import org.json.simple.parser.ParseException;
```

Для решений на языке **C++** доступна библиотека nlohmann/json v3.8.0. Соответствующие include выглядят так:

#include "json.hpp"

Для решений на **golang** доступны все стандартные пакеты, включая encoding/json, sort и другие.

Для решений на **C#** доступны библиотеки System.Text.Json и Newtonsoft.Json. Соответствующие using могут выглядеть так: using Newtonsoft.Json; using System.Text.Json;



```
1 import java.io.BufferedReader;
  2 import java.io.IOException;
  3 import java.io.InputStreamReader;
    import java.util.*;
    import java.util.stream.Collectors;
    import org.json.simple.JSONArray;
  8 import org.json.simple.JSONObject;
  9 import org.json.simple.JSONValue;
 10 import org.json.simple.parser.JSONParser;
 import org.json.simple.parser.ParseException;
 12
 13 public class CTask {
 14
 15
        @SuppressWarnings("unchecked")
 16
        public static void main(String[] args) throws IOException, ParseException {
 17
            BufferedReader r = new BufferedReader(new InputStreamReader(System.in));
 18
            JSONParser parser = new JSONParser();
 19
            List<Offer> input = new ArrayList<>();
 20
 21
            int feedCount = Integer.parseInt(r.readLine());
 22
 23
            for (int i = 0; i < feedCount; ++i) {</pre>
 24
                JSONObject jsonFeed = (JSONObject) parser.parse(r.readLine());
 25
                JSONArray jsonFeedOffers = (JSONArray) jsonFeed.get("offers");
 26
                jsonFeedOffers.forEach(offer -> {
 27
                    JSONObject offerJson = (JSONObject) offer;
 28
                    input.add(new Offer(String.valueOf(offerJson.get("offer_id")),
 29
                            Integer.parseInt(String.valueOf(offerJson.get("market_sku"))
 30
                            Integer.parseInt(String.valueOf(offerJson.get("price")))));
 31
                });
 32
 33
            Feed result = new Feed(new ArrayList<>());
 34
            result.getOffers().addAll(input.stream()
 35
                     .sorted((a, b) -> new OfferComparator().compare(a, b))
                     .collect(Collectors.toList()));
 36
 37
            System.out.println(JSONValue.toJSONString(result.toJSONObject()));
38
```

Java 8 + network + json-simple

OK

Отправить і осталось 99 попыток
Предыдущая
Время посылки ID Задача Компилятор

С

51725110

 Тип посылки
 Время
 Память
 Тест
 Баллы

 1.376s
 205.23Mb
 1
 отчёт

Следующая

30 май 2021, 11:09:47