

# Тестовое задание junior python developer

**Время на выполнение:** 4-6 часов

**Стек:** Python3, Asyncio, MongoDB, любая асинхронная библиотека для телеграм бота

**Приоритетным является решение через построение запросов к mongodb**

## Описание задачи:

Вашей задачей в рамках этого тестового задания будет написание алгоритма агрегации статистических данных о зарплатах сотрудников компании по временным промежуткам. Ссылка на скачивание коллекции со статистическими данными, которую необходимо использовать при выполнении задания, находится в конце документа.

На обычном языке пример задачи выглядит следующим образом:

“Необходимо посчитать суммы всех выплат с {28.02.2022} по {31.03.2022}, единица группировки - {день}”.

## Ваш алгоритм должен принимать на вход:

1. Дату и время старта агрегации в ISO формате (далее dt\_from)
2. Дату и время окончания агрегации в ISO формате (далее dt\_upto)
3. Тип агрегации (далее group\_type). Типы агрегации могут быть следующие: hour, day, month. То есть группировка всех данных за час, день, неделю, месяц.

**Пример входных данных:**

```
{  
  "dt_from": "2022-09-01T00:00:00",  
  "dt_upto": "2022-12-31T23:59:00",  
  "group_type": "month"  
}
```

**Комментарий к входным данным:** вам необходимо агрегировать выплаты с 1 сентября 2022 года по 31 декабря 2022 года, тип агрегации по месяцу

**На выходе ваш алгоритм формирует ответ содержащий:**

1. Агрегированный массив данных (далее dataset)
2. Подписи к значениям агрегированного массива данных в ISO формате (далее labels)

**Пример ответа:**

```
{"dataset": [5906586, 5515874, 5889803, 6092634], "labels":  
["2022-09-01T00:00:00", "2022-10-01T00:00:00",  
"2022-11-01T00:00:00", "2022-12-01T00:00:00"]}
```

**Комментарий к ответу:** в нулевом элементе датасета содержится сумма всех выплат за сентябрь, в первом элементе сумма всех выплат за октябрь и т.д. В лейблах подписи соответственно элементам датасета.

**Несколько примеров для проверки корректности работы вашего алгоритма на нашей коллекции данных:**

**Входные данные 1**

```
{  
  "dt_from": "2022-09-01T00:00:00",  
  "dt_upto": "2022-12-31T23:59:00",  
  "group_type": "month"  
}
```

**Ответ 1**

```
{"dataset": [5906586, 5515874, 5889803, 6092634],  
"labels": ["2022-09-01T00:00:00", "2022-10-01T00:00:00",  
"2022-11-01T00:00:00", "2022-12-01T00:00:00"]}
```

**Входные данные 2**

```
{  
  "dt_from": "2022-10-01T00:00:00",  
  "dt_upto": "2022-11-30T23:59:00",  
  "group_type": "day"  
}
```

## Ответ 2

```
{"dataset": [0, 0, 0, 195028, 190610, 193448, 203057, 208605, 191361, 186224, 181561, 195264, 213854, 194070, 208372, 184966, 196745, 185221, 196197, 200647, 196755, 221695, 189114, 204853, 194652, 188096, 215141, 185000, 206936, 200164, 188238, 195279, 191601, 201722, 207361, 184391, 203336, 205045, 202717, 182251, 185631, 186703, 193604, 204879, 201341, 202654, 183856, 207001, 204274, 204119, 188486, 191392, 184199, 202045, 193454, 198738, 205226, 188764, 191233, 193167, 205334], "labels": ["2022-10-01T00:00:00", "2022-10-02T00:00:00", "2022-10-03T00:00:00", "2022-10-04T00:00:00", "2022-10-05T00:00:00", "2022-10-06T00:00:00", "2022-10-07T00:00:00", "2022-10-08T00:00:00", "2022-10-09T00:00:00", "2022-10-10T00:00:00", "2022-10-11T00:00:00", "2022-10-12T00:00:00", "2022-10-13T00:00:00", "2022-10-14T00:00:00", "2022-10-15T00:00:00", "2022-10-16T00:00:00", "2022-10-17T00:00:00", "2022-10-18T00:00:00", "2022-10-19T00:00:00", "2022-10-20T00:00:00", "2022-10-21T00:00:00", "2022-10-22T00:00:00", "2022-10-23T00:00:00", "2022-10-24T00:00:00", "2022-10-25T00:00:00", "2022-10-26T00:00:00", "2022-10-27T00:00:00", "2022-10-28T00:00:00", "2022-10-29T00:00:00", "2022-10-30T00:00:00", "2022-10-31T00:00:00", "2022-11-01T00:00:00", "2022-11-02T00:00:00", "2022-11-03T00:00:00", "2022-11-04T00:00:00", "2022-11-05T00:00:00", "2022-11-06T00:00:00", "2022-11-07T00:00:00", "2022-11-08T00:00:00", "2022-11-09T00:00:00", "2022-11-10T00:00:00", "2022-11-11T00:00:00", "2022-11-12T00:00:00", "2022-11-13T00:00:00", "2022-11-14T00:00:00", "2022-11-15T00:00:00", "2022-11-16T00:00:00", "2022-11-17T00:00:00", "2022-11-18T00:00:00", "2022-11-19T00:00:00", "2022-11-20T00:00:00", "2022-11-21T00:00:00", "2022-11-22T00:00:00", "2022-11-23T00:00:00", "2022-11-24T00:00:00", "2022-11-25T00:00:00", "2022-11-26T00:00:00", "2022-11-27T00:00:00", "2022-11-28T00:00:00", "2022-11-29T00:00:00", "2022-11-30T00:00:00"]}]
```

### Входные данные 3

```
{  
  "dt_from": "2022-02-01T00:00:00",  
  "dt_upto": "2022-02-02T00:00:00",  
  "group_type": "hour"  
}
```

### Ответ 3

```
{"dataset": [8177, 8407, 4868, 7706, 8353, 7143, 6062, 11800, 4077, 8820, 4788,  
11045, 13048, 2729, 4038, 9888,  
7490, 11644, 11232, 12177, 2741, 5341, 8730, 4718, 0],  
"labels": ["2022-02-01T00:00:00", "2022-02-01T01:00:00",  
"2022-02-01T02:00:00", "2022-02-01T03:00:00",  
"2022-02-01T04:00:00", "2022-02-01T05:00:00", "2022-02-01T06:00:00",  
"2022-02-01T07:00:00",  
"2022-02-01T08:00:00", "2022-02-01T09:00:00", "2022-02-01T10:00:00",  
"2022-02-01T11:00:00",  
"2022-02-01T12:00:00", "2022-02-01T13:00:00", "2022-02-01T14:00:00",  
"2022-02-01T15:00:00",  
"2022-02-01T16:00:00", "2022-02-01T17:00:00", "2022-02-01T18:00:00",  
"2022-02-01T19:00:00",  
"2022-02-01T20:00:00", "2022-02-01T21:00:00", "2022-02-01T22:00:00",  
"2022-02-01T23:00:00",  
"2022-02-02T00:00:00"]}
```

После разработки алгоритма агрегации, вам необходимо создать телеграм бота, который будет принимать от пользователей текстовые сообщения содержащие JSON с входными данными и отдавать агрегированные данные в ответ. Посмотрите @rlt\_testtaskexample\_bot - в таком формате должен работать и ваш бот.

### Как происходит проверка тестового задания?

После выполнения тестового задания вам необходимо прислать нашему боту @rlttesttask\_validator никнейм вашего телеграм бота. Наш бот автоматически обратится к вашему боту и прокинет несколько тестовых запросов, если агрегированные ответы вашего бота окажутся верными - вы прошли тестовое задание.

**Внимание! Чтобы бот начал проверку вашего задания вам необходимо сообщить в чате на hh.ru ваш telegram id для**

**выдачи доступа, его можно узнать, написав в @rlttesttask\_validator**

**Внимание 2! Код вашего решения должен быть размещен в публичном репозитории на GitHub. После прохождения валидации, наш бот запросит ссылку на репозиторий, для того чтобы мы провели код-ревью**

**Внимание 3! Отсчет времени на выполнение начинается с первого коммита**

По любым вопросам пишите в вакансии на hh.ru

Коллекция со статистическими данными -  
<https://drive.google.com/file/d/1pcNm2TAtXHO4Jlad9dkzpbNc4q7NoYkx/view?usp=sharing>