

# Тестовое задание PHP Developer

Необходимо решить описанных ниже два задания, результат залить в публичный репозиторий и в README файле в качестве описания указать решение [Задания 1](#), а так-же инструкцию и описание по [Заданию 2](#).

**Важно:** написать подробную инструкцию по развертыванию и тестированию.

Перед тем, как приступить к выполнению задания, необходимо ознакомиться с описанием и задать вопросы. Также необходимо написать когда по планам готовы приступить к выполнению.

## Задание 1

### Mysql

Имеется три таблицы:

#### users

id	first_name	last_name	age
1	Ivan	Ivanov	18
2	Marina	Ivanova	14

#### books

id	name	author
1	Romeo and Juliet	William Shakespeare
2	War and Peace	Leo Tolstoy

#### user\_books

id	user_id	book_id
1	1	2
2	2	1

Необходимо написать запрос выборки данных из представленных таблиц, который найдет и выведет всех посетителей библиотеки, возраст которых попадает в диапазон от 7 и до 17 лет, а также которые взяли только две книги и все книги, которые у них на руках, одного и того же автора.

**Формат вывода:**

**ID, Name (first\_name last\_name), Author, Books (Book 1, Book 2, ...)**

**1; Ivan Ivanov; Leo Tolstoy; Book 1, Book 2**

## Задание 2

### PHP

Необходимо реализовать JSON API сервис на языке **php** 8 (можно использовать любой **php framework**) для работы с курсами обмена валют для биткоина (**BTC**). Реализовать необходимо с помощью **Docker**.

Сервис для получения текущих курсов валют: <https://blockchain.info/ticker>

Все методы API будут доступны только после авторизации, т.е. все методы должны быть по умолчанию не доступны и отдавать ошибку авторизации.

Для авторизации будет использоваться фиксированный токен (64 символа включающих в себя a-z A-Z 0-9 а так-же символы - и \_ ), передавать его будем в заголовках запросов. Тип **Authorization: Bearer**.

#### Формат запросов:

<your\_domain>/api/v1?method=<method\_name>&<parameter>=<value>

Формат ответа API: **JSON** (все ответы при любых сценариях должны иметь **JSON формат**)

**Все значения курса обмена должны считаться учитывая нашу комиссию = 2%**

---

API должен иметь 2 метода:

- 1) **rates**: Получение всех курсов с учетом комиссии = 2% (GET запрос) в формате:

```
{
  "status": "success",
  "code": 200,
  "data": {
    "USD" : <rate>,
    ...
  }
}
```

В случае ошибки:

```
{
  "status": "error",
  "code": 403,
  "message": "Invalid token"
}
```

Сортировка от меньшего курса к большему курсу.

В качестве параметров может передаваться интересующая валюта, в формате USD,RUB,EUR и тп В этом случае, отдаем указанные в качестве параметра currency значения.

- 2) **convert**: Запрос на обмен валюты с учетом комиссии = 2%. POST запрос с параметрами:

currency\_from: USD

currency\_to: BTC

value: 1.00

или в обратную сторону

currency\_from: BTC

currency\_to: USD

value: 1.00

В случае успешного запроса, отдаем:

```
{
  "status": "success",
  "code": 200,
  "data": {
    "currency_from": BTC,
    "currency_to": USD,
    "value": 1.00,
    "converted_value": 1.00,
    "rate": 1.00,
  }
}
```

В случае ошибки:

```
{
  "status": "error",
  "code": 403,
  "message": "Invalid token"
}
```

Важно, минимальный обмен равен 0,01 валюты from

Например: USD = 0.01 меняется на 0.0000005556 (считаем до 10 знаков)

Если идет обмен из BTC в USD - округляем до 0.01