

Тестовое задание Python + Django

Нужно написать сервис, который будет интегрироваться с внешними учетными системами, которые предоставляют данные. В частности нужно провести одну из интеграций (+ возможно предусмотреть относительно несложное добавление интеграции аналогичных данных из других внешних систем).

Итак, Дано:

Внешняя система 1С

Пример запроса на получение списка сотрудников:

POST <http://176.192.70.122:90/fitness nfc mobile/hs/nfc mobile/v1>

Basic auth

login: FitnessKit

password: vY0xodyg

Body:

```
{
  "Request_id": "e1477272-88d1-4acc-8e03-7008cdedc81e",
  "ClubId": "59115d1e-9052-11eb-810c-6eae8b56243b",
  "Method": "GetSpecialistList",
  "Parameters": {
    "ServiceId": ""
  }
}
```

Request_id – хардкодится для данного метода (Method)

ClubId – изменяемый параметр, он разный для разных копий учетной системы.

Такие параметры как ClubId, login, password, ссылка должны быть настраиваемыми.

Что нужно сделать:

Реализовать АПИ запрос

GET /team/get_employees

Который бы получал данные из внешней системы, преобразовывал их и выдавал в ответ.

Пример ответа можно получить, если отправить запрос на уже реализованный аналогичный сервис.

<https://test.fitnesskit-admin.ru/team/get/1/>

Необязательно делать все поля, главное, чтобы у сотрудников в ответе были:

id (идентификатор сотрудника)

name (имя сотрудника)

last_name (фамилия сотрудника)

phone (телефон сотрудника)

image_url (аватар сотрудника)

В случае отсутствия какого-либо из полей присылать пустую строку в этом месте.

В качестве решения тестового задания нужно прислать:

- 1) Ссылку на админку задеплоенного решения
- 2) Логин и пароль к админке
- 3) Ссылку на АПИ запрос на список сотрудников
- 4) Ссылку на код (желательно гинхаб)

Несколько важных замечаний:

- 1) Если эталонный запрос <https://test.fitnesskit-admin.ru/team/get/1/> отвечает нормально, значит 1С функционирует в штатном режиме. Если у вас не получается получить от неё нормальный ответ, то скорее всего вы как-то не так отправляете запрос, смотрите внимательнее на пример.
- 2) Мы не будем смотреть только код, всегда сначала будет тыкаться в задеплоенное решение.
- 3) **САМОЕ ВАЖНОЕ ЗАМЕЧАНИЕ.** Нужно использовать **django 3+**, и обработка АПИ запроса должна быть полностью асинхронной.