## Web-приложение для определения заполненных форм.

**По поводу сроков выполнения: тестовые задания принимаются до тех пор, пока открыта вакансия.**

**Результат лучше всего присылать ссылкой на репозиторий Github**

В базе данных хранится список шаблонов форм.

Шаблон формы, это структура, которая задается уникальным набором полей, с указанием их типов.

Пример шаблона формы:

| {  "name": "Form template name",  "field\_name\_1": "email",  "field\_name\_2": "phone" } |
| --- |

Всего должно поддерживаться четыре типа данных полей:

1. email
2. телефон
3. дата
4. текст.

Все типы кроме текста должны поддерживать валидацию. Телефон передается в стандартном формате **+7 xxx xxx xx xx**, дата передается в формате **DD.MM.YYYY** или **YYYY-MM-DD**.

Имя шаблона формы задается в свободной форме, например *MyForm* или *Order Form*.

Имена полей также задаются в свободной форме (желательно осмысленно), например *user\_name, order\_date* или *lead\_email.*

На вход по урлу **/get\_form** POST запросом передаются данные такого вида:

**f\_name1=value1&f\_name2=value2**

В ответ нужно вернуть имя шаблона формы, если она была найдена.

Чтобы найти подходящий шаблон нужно выбрать тот, поля которого совпали с полями в присланной форме. Совпадающими считаются поля, **у которых совпали имя и тип значения**. Полей в пришедшей форме может быть больше чем в шаблоне, в этом случае шаблон все равно будет считаться подходящим. Самое главное, чтобы все поля шаблона присутствовали в форме.

Если подходящей формы не нашлось, вернуть ответ в следующем формате

| {  f\_name1: FIELD\_TYPE,  f\_name2: FIELD\_TYPE } |
| --- |

где FIELD\_TYPE это тип поля, выбранный на основе правил валидации, проверка правил должна производиться в следующем порядке дата, телефон, email, текст.

В качестве базы данных рекомендуем использовать [tinyDB](http://tinydb.readthedocs.io/en/latest/intro.html), вместе с исходниками задания должен поставляться файл с тестовой базой, содержащей шаблоны форм. Но если сможете поднять и использовать контейнер [Docker](https://www.docker.com/) с [MongoDB](https://www.mongodb.com/) - это будет отличное решение, однако оно может отнять у вас много времени и не является обязательным.

Также в комплекте должен быть скрипт, который совершает тестовые запросы. Если окружение приложения подразумевает что-то выходящее за рамки virtualenv, то все должно быть упаковано в Docker контейнеры или таким способом, чтобы не приходилось ставить дополнительные пакеты и утилиты на машине. Все необходимые действия для настройки и запуска приложения должны находится в файле README.

Версия Python остается на ваш выбор. Мы рекомендуем использовать версию 3.6 и выше.

**Входные данные для веб-приложения:**Список полей со значениями в теле POST запроса.

**Выходные данные:**

Имя наиболее подходящей данному списку полей формы, при отсутствии совпадений с известными формами произвести типизацию полей на лету и вернуть список полей с их типами.