

Тестовое задание для QA Automation

Подающий надежды молодой программист Джо, работающий в компании Noodle, участвует в проекте по разработке новой социальной сети NoodlePlus. Проект горит, конкуренты не дремлют, писать надо быстро. Джо не задумываясь берется за дело. Проходит пара дней, и сердце проекта готово и бьется. По крайней мере так думает менеджер проекта Сара. Так случилось, что вы работаете в этой компании тестировщиком. Вы уважаете Джо, но после вчерашнего похода в бар знаете, что тот держится на пике Балмера не очень долго, и после этого становится склонен к сомнительным авантюрам. Таким образом, когда Сара приходит к вам с просьбой протестировать продукт, вы понимаете, что будущее проекта на ваших плечах, и только вы можете спасти грядущее IPO. Вы получаете от Джо уже готовое приложение. Джо сообщает вам, что для его запуска вам потребуется ubuntu 18.04+. Запускать его надо таким образом:

```
./tester 127.0.0.1 4000
```

Здесь 127.0.0.1 - адрес локалхоста, а 4000 - некий свободный порт в системе. Более того, Джо даже был так любезен, что приложил скрипт которым якобы его потестировал. Посмотрев на него вы сразу понимаете, что приложение общается через вебсокет при помощи json, занимает указанный хост и порт в системе, а также делает вывод о том как качественно и продуманно Джо подошел к тестированию. Скрипт выглядит так:

```
#!/usr/bin/python3

import websockets
import asyncio
import json
import sys

async def do_smth():
    uri = "ws://127.0.0.1:4000"
    async with websockets.connect(uri) as ws:
        await ws.send('{"method": "delete", "id": "2341514214", "phone": "2128507"}')
        repl = await ws.recv()
        print (json.loads(repl))

asyncio.get_event_loop().run_until_complete(do_smth())
```

Впрочем, вы не питаете особых иллюзий на тему качества продукта, и намереваетесь все перепроверить, найти как можно больше багов пока это не попало в продакшн. Также вы подозреваете, что в будущем проект будет меняться, потому хотите покрыть все API автоматическими тестами на языке python3. Кроме этого, Сара сказала что за каждый найденный баг (с автоматическим кейсом который его воспроизводит) вы будете получать +1 к карме в ее глазах. Она считает багами следующее: 1. Несоответствия заявленному API 2. Ошибки приводящие к падению приложения 3. Ошибки, представляющие из себя потенциальный вектор атаки на приложение

API

Запросы

add

Если юзера нет в базе, добавить нового юзера, иначе вернуть ошибку.

Запрос: - id: string, идентификатор запроса - method: add - name: string, Имя юзера - surname: string, Фамилия юзера - phone: string, unique, primary key, Телефон юзера - age: integer, Возраст юзера

Пример запроса:

```
{
  "id": "sfda-11231-123-adfa",
  "method": "add",
  "name": "Chuck",
  "surname": "Dorris",
  "phone": "2128507",
  "age": 100500
}
```

Ответ: - status: success | failure - method: add - id: id запроса

Пример ответа:

```
{
  "id": "sfda-11231-123-adfa",
  "method": "add",
  "status": "success"
}
```

delete

Удалить из базы юзера по номеру телефона

Запрос: - id: string, идентификатор запроса - method: delete - phone: string, unique, primary key, Телефон юзера

Ответ: - status: success | failure - method: delete - id: id запроса

update

Если юзер есть в базе, обновить юзера, иначе вернуть ошибку.

Запрос: - id: string, идентификатор запроса - method: update - name: string, Имя юзера - surname: string, Фамилия юзера - phone: string, unique, primary key, Телефон юзера - age: integer, Возраст юзера

Ответ: - status: success | failure - method: delete - id: id запроса

select

Получить из базы одного (по номеру телефона) или несколько юзеров (по имени или фамилии). Можно выбирать по телефону (он уникален, один юзер), либо по фамилии или имени (по отдельности, все юзеры которые совпадут).

Запрос: - id: string, идентификатор запроса - method: select - name | phone | surname: string, Один из критериев поиска. Должен отдавать всех пользователей соответствующих критерию.

Пример запроса:

```
{
  "id": "123412-adf-13213",
  "method": "select",
  "surname": "Obama"
}
```

Ответ: - status: success | failure - method: select - id: id запроса - users: массив пользователей

Пример ответа:

```
{
  "id": "123412-adf-13213",
  "method": "select",
  "status": "success",
  "users": [
    {
      "name": "Michele",
      "surname": "Obama",
      "age": 18,
      "phone": "+131231241"
    },
    {
      "name": "Barack",
      "surname": "Obama",
      "age": 24,
      "phone": "+131314135"
    }
  ]
}
```

Ошибки

В случае ошибки возвращается: - status: failure - reason: сообщение об ошибке - id: id запроса