

Тестовое задание: iOS developer

Задание: создать iOS приложение для управления пользователями, которые хранятся на сервере.

Мы оцениваем стиль кода, его читаемость, соответствие гайдлайнам, наличие и качество документации. Большое внимание уделяем внешнему виду, дизайну и деталям.

Использование библиотек приветствуется, ограничений нет. Задание должно быть выполнено с использованием последней стабильной версии Swift.

Сервер

Сервер развёрнут по следующему адресу: <https://frogogo-test.herokuapp.com>.

Список пользователей

Первый экран приложения. Необходимо отобразить список всех пользователей с сервера. Отображаются следующие данные: имя и фамилия в заголовке ячейки, аватар в левой части (если имеется), email в подзаголовке.

Для получения пользователей необходимо отправить GET запрос на сервер.

| GET /users.json

```
[
  {
    "id":1,
    "first_name":"Foo",
    "last_name":"Bar",
    "email":"foo@bar.com",
    "avatar_url":"",
    "created_at":"2016-09-20T14:44:33.573Z"
  },
  {
    "id":2,
    "first_name":"Baz",
```

```
"last_name": "Qux",  
"email": "baz@qux.com",  
"avatar_url": "http://www.fillmurray.com/200/200",  
"created_at": "2016-09-20T15:33:25.536Z"  
}  
]
```

Создание пользователя

На экране списка пользователей должна быть кнопка "+", при нажатии на которую открывается экран с формой для создания пользователя.

В форме имеются следующие поля: имя, фамилия, почта. Поля должны проверяться на валидность: наличие имени и фамилии, правильный формат почты.

После нажатия на кнопку "Создать" отправляется POST запрос на сервер в формате JSON с данными о пользователе.

| POST /users.json

```
{  
  "user": {  
    "first_name": "Foo",  
    "last_name": "Bar",  
    "email": "foo@bar.com",  
    "avatar_url": ""  
  }  
}
```

- Успешное выполнение запроса: 201 Created
- При наличии ошибок: 422 Unprocessable Entity с описанием ошибок в теле респонса.

Редактирование пользователя

При нажатии на ячейку пользователя открывается окно для редактирования пользователя. Окно идентично тому, что используется при создании, однако все поля заполнены имеющимися значениями.

PATCH /users/1.json

```
{
  "user": {
    "first_name": "Foo",
    "last_name": "Bar",
    "email": "foo@bar.com",
    "avatar_url": ""
  }
}
```

- Успешное выполнение запроса: 200 OK
- При наличии ошибок: 422 Unprocessable Entity с описанием ошибок в теле респонса.

Результат

Загрузить на GitHub, ссылку на репозиторий отправить на [почту](#). Пожалуйста, не присылайте репозитории, где только два-три коммита, или проекты без какой-либо документации и README – такие работы будут сразу отклоняться. По вопросам обращаться в [Telegram](#).