

## **Задача 3**

На картинке (скачать можно по ссылке -

[https://drive.google.com/file/d/1fi9JLRvO6bhS1uJJ2PUX8ZTOq0Eq\\_At0/view?usp=drive\\_link](https://drive.google.com/file/d/1fi9JLRvO6bhS1uJJ2PUX8ZTOq0Eq_At0/view?usp=drive_link)) изображен интерфейс регистрации Пользователя (как успешный экран создания Пользователя, так и варианты с ошибками).

### **Задание 3.1:**

Требуется описать REST API интерфейс на стороне БЭК сервиса, который должен вызываться фронтеном при нажатии на кнопку “Register”. Описание должно включать:

- http метод + URL
- входные параметры (наименование, тип, обязательность, ограничения)
- выходные параметры при успешном ответе (наименование, тип, обязательность, ограничения)
- выходные параметры при ответе с ошибкой (наименование, тип, обязательность, ограничения)
- описание ошибок - коды ответов (главное верно указать клиентские или серверные) + текст сообщения
- пример запроса
- пример ответа (для успешного сценария и сценария с ошибкой).

Требований к формату нет (может быть в виде таблицы или просто текстом).

Допускается описание в формате openAPI спецификации. Полученный yaml приложить в виде текста к решению или в виде отдельного файла (обязательно не забыть открыть доступ).

### **Задание 3.2**

Описать подробный пошаговый алгоритм, создания Пользователя на стороне БЭК сервиса при вызове, получившегося в Задачи 3.1 метода. Алгоритм должен подробно описывать какие проверки требуется выполнять БЭК сервису после получения запроса на создание нового Пользователя (при нажатии кнопки “Register”), прежде чем создать Пользователя в БД. Описание может быть в виде текста или диаграммы (например, activity)/блок-схемы.

## Задание №3

### Задание 3.1

#### 1. HTTP метод + URL

POST - метод для передачи данных на сервер

/api/v1/users/register - URL для создания нового пользователя

#### 2. Входные параметры

Наименование	Тип	Обязательность	Ограничения
First Name	String	Обязательный	1-20 символов
Last Name	String	Обязательный	1-20 символов
UserName	String	Обязательный	1-20 символов
Password	String	Обязательный	Минимум 8 символов 1 заглавная буква 1 цифра 1 специальный символ
CaptchaToken	String	Обязательный	

#### 3. Выходные параметры при успешном ответе

Наименование	Тип	Обязательность	Описание
UserName	String	Да	Логин пользователя
Message	String	Да	Сообщение об успешной регистрации

#### **4. Выходные параметры при ответе с ошибкой**

Наименование	Тип	Обязательность	Описание
ErrorCode	String	Да	Код ошибки
ErrorMessage	String	Да	Описание ошибки

#### **5. Описание ошибок - Коды ответов**

Код ошибки	Тип	Текст сообщения
409	Клиентская	Пароль должен содержать по крайней мере один символ, одну цифру (0-9), одну заглавную букву (A-Z), одну строчную букву (a-z), один специальный символ и пароль должен содержать более восьми символов.
409	Клиентская	Пользователь уже существует.
409	Клиентская	Пожалуйста пройдите капчу для регистрации.
400	Клиентская	Пожалуйста заполните все обязательные поля.
500	Серверная	Ошибка на строны сервера. Пожалуйста повторите запрос через некоторое время.

#### **6. Пример запроса**

```
{  
    "FirstName": "Ivan",  
    "LastName": "Ivanov",  
    "Username": "ivan",  
    "Password": "Qwerty!1",  
    "CaptchaToken": "123"  
}
```

## **7. Примеры ответов**

**Для успешного сценария (код 201)**

```
{  
    "id": "12345",  
    "username": "ivan",  
    "message": "User registered successfully"  
}
```

**Для сценария с ошибкой**

```
{  
    "errorCode": "USER_ALREADY_EXISTS",  
    "message": "User already exists"  
}
```

## **Задание 3.2**

### **1. Сервис получил пост запрос**

Проверяется, содержит ли запрос тело в формате JSON

### **2. Проверка всех обязательных полей**

Сервис проверяет, заполнены ли все обязательные поля.

Если какое-нибудь поле не заполнено, сервис отправляет ошибку с кодом 400 и сообщение с описанием ошибки.

### **3. Проверка правильности личных данных**

Сервис проверяет что:

- строка FirstName, LastName UserName не пустые и содержат 1-20 символов
- пароль имеет минимум 8 символов, содержит как минимум 1 заглавную букву, 1 строчную букву, 1 цифру и 1 специальный символ

Если какие то данные не проходят проверку, сервис возвращает ошибку с соответствующим кодом.

### **4. Проверка Captcha**

Сервис отправляет полученный в запросе CaptchaToken во внешний сервис проверки Captcha

Если Captcha отклонена, сервис возвращает ошибку с соответствующим кодом.

### **5. Проверка уникальности пользователя.**

Сервис выполняет запрос username в БД. Если пользователь с таким username существует, то сервис возвращает ошибку с соответствующим кодом.

### **6. Если пользователь не существует, то сервис создает нового пользователя в БД.**

Если новая запись в БД по какой-нибудь причине не сохраняется, то сервис возвращает ошибку с кодом 500.

### **7. Сервис шифрует пароль пользователя**

### **8. Сервис формирует ответ и отправляет пользователю сообщение об успешной регистрации с кодом 201**