**Задача:**

Протестировать функционал, задача которого создать пользователя и определить его возрастную группу.

Сервис работает по протоколу https, параметры передаются в теле сообщения в формате JSON

**Описание доступных api-запросов:**

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Апи | Тело | Ответ при успехе |
| POST | http://host:80/CreateUser | {  "Name": "String"  } | 201 ОК  {  "id": "String"  } |
| GET | http://host:80/GetUser | {  "id": "String"  } | 200 ОК  {  "Name": "String",  "Age": int32,  } |
| POST | http://host:80/SetUserAge | {  "id": "String",  "Age ": Int32  } | 200 ОК |
| GET | http://host:80/{id}/GetAgeGroupById |  | 200 ОК  {  "AgeGroup": "String"  } |

Правило определения группы (AgeGroup):

Young – age от 0 до 18

Adult – age от 19 до 100

**Примечание:**

Решение прислать в любой удобной форме.

**Решение:**

Предлагаю обратить внимание пункт «правило определения группы» я полагаю, что в требованиях к функциональности допущена ошибка в нижней границе группы возраста Adult либо в верхней границе возраста Young, так как возраст 18 лет не входит ни в одну из указанных групп (в группе Young указан возраст «до 18»).

Предлагаю обратить внимание пункт «описание доступных api-запросов» я полагаю, что здесь допущена ошибка, так как запрос метода GET не может содержать тело запроса. В связи с этим протестировать создание пользователя этим методом в данном варианте описания невозможно, так как не способа передать идентификатор пользователя (шаг 3 в тест-кейсах выполнить нельзя, создание пользователя можно проверить только через шаг 4).

**Предлагаю следующие тест-кейсы:**

Тест-кейс № 1 (позитивный): возрастной группы Young, возраст равен 0:

1. Создать пользователя
2. Задать возраст 0
3. Проверить созданного пользователя (имя и возраст)
4. Проверить, что пользователь принадлежит к группе Young

Тест-кейс № 2 (позитивный) возрастной группы Young, возраст от 1 до 17:

1. Создать пользователя
2. Задать возраст от 1 до 17
3. Проверить созданного пользователя (имя и возраст)
4. Проверить, что пользователь принадлежит к группе Young

Тест-кейс № 3 (позитивный) возрастной группы Young, возраст равен 17:

1. Создать пользователя
2. Задать возраст 17
3. Проверить созданного пользователя (имя и возраст)
4. Проверить, что пользователь принадлежит к группе Young

Тест-кейс № 4 (негативный) возрастной группы Young, возраст равен 18:

1. Создать пользователя
2. Задать возраст 18
3. Проверить созданного пользователя (имя и возраст)
4. Проверить, что пользователь не принадлежит к группе Young
5. Проверить, что пользователь не принадлежит к группе Adult

Тест-кейс № 5 (позитивный) возрастной группы Adult, возраст равен 19:

1. Создать пользователя
2. Задать возраст 19
3. Проверить созданного пользователя (имя и возраст)
4. Проверить, что пользователь принадлежит к группе Adult

Тест-кейс № 6 (позитивный) возрастной группы Adult, возраст 20-99:

1. Создать пользователя
2. Задать возраст 20-99
3. Проверить созданного пользователя (имя и возраст)
4. Проверить, что пользователь принадлежит к группе Adult

Тест-кейс № 7 (негативный) возрастной группы Adult, возраст равен 100:

1. Создать пользователя
2. Задать возраст 100
3. Проверить созданного пользователя (имя и возраст)
4. Проверить, что пользователь не принадлежит к группе Adult
5. Проверить, что пользователь не принадлежит к группе Young

Тест-кейс № 8 (негативный) добавления пользователя с отрицательным возрастом:

1. Создать пользователя
2. Задать отрицательный возраст, например, -1
3. Получить ошибку: недопустимый возраст

Тест-кейс № 9 (негативный) добавления пользователя с возрастом более 100:

1. Создать пользователя
2. Задать возраст больше 100
3. Проверить созданного пользователя (имя и возраст)
4. Получить ошибку: недопустимый возраст

Тест-кейс № 10 (негативный) добавления пользователя с пустым именем:

1. Создать пользователя с пустым именем
2. Получить ошибку: недопустимое имя

Дополнительно можно также проверить значения возраста 1, 20 и 99.