**Фильтрация. При попытке отфильтровать пользователей по конкретным значениями полей в ответе выводится весь список пользователей**

Условия

Стенд: https://qa-course.kontur.host/api\_kontur\_practice2/

Авторизация: Type - Basic Auth, login - admin, password - admin

Шаги

1. Выбрать метод GET, в адрес ввести запрос [**https://qa-course.kontur.host/api\_kontur\_practice2/api/users/?name=Мария**](https://qa-course.kontur.host/api_kontur_practice2/api/users/?name=%D0%9C%D0%B0%D1%80%D0%B8%D1%8F)

2. Отправить запрос

ОР: В теле ответа появятся список пользователей,значение поля “name” у которых совпадает с “Мария”

ФР: Фильтрация не срабатывает, в теле ответа выводится список всех добавленных ранее пользователей

Скриншот: <http://joxi.ru/D2Pl5ypHwkvoP2>

**Добавление пользователя. После отправки данных о пользователе к введенному значению в поле Место проживания / Location добавляется значение « Сити».**

Условия

Стенд:<https://qa-course.kontur.host/api_kontur_practice3/>

Авторизация: Type - Basic Auth, login - admin, password - admin

Тело запроса: вид raw, формат JSON

Шаги

1. Выбрать метод POST, в адрес ввести запрос <https://qa-course.kontur.host/api_kontur_practice3/api/users/create>

2. В тело ввести

{

"ItemsOfClients":

[

{

"id": "787878",

"Person":

{

"Name":"Test",

"Surname":"Test",

"Patronymic":"Test",

"DateOfBirth":"01.01.2000",

"Location":"Екатеринбург"

}

}

]

}

В поле "id": " " указать любое положительное целое числовое значение. (Например, "id": "787878")

4. Отправить запрос, получить в ответе "Created successfully"

5. Выбрать метод GET, в адрес ввести запрос [https://qa-course.kontur.host/api\_kontur\_practice1/api/users/{userId](https://qa-course.kontur.host/api_kontur_practice1/api/users/%7BuserId)}, подставив вместо [{userId](https://qa-course.kontur.host/api_kontur_practice1/api/users/%7BuserId)} значение ID введенное в Шаге 2.

ОР: В теле ответа появятся данные о созданном пользователе, совпадающие с ранее отправленными

ФР: В теле ответа появятся данные о созданном пользователе, в поле Location к ранее введенному значению добавляется значение “ Сити”.

Скриншот:

<http://joxi.ru/GrqW4BNFzkxDb2> - Добавился с Location “Екб”

<http://joxi.ru/52ax7edclbeXMm> - Стал с Location “Екб Сити”

**Дата рождения выводится в некорректном формате при вводе значения дня <13 при добавлении пользователя**

Условия

Стенд:<https://qa-course.kontur.host/api_kontur_practice3/>

Авторизация: Type - Basic Auth, login - admin, password - admin

Запрос <https://qa-course.kontur.host/api_kontur_practice3/api/users/create>

Тело запроса: вид raw, формат JSON

Шаги

1. Выбрать метод POST, в адрес ввести запрос <https://qa-course.kontur.host/api_kontur_practice3/api/users/create>

2. В тело запроса ввести

{

"ItemsOfClients":

[

{

"id": " ",

"Person":

{

"Name":"Test",

"Surname":"Test",

"Patronymic":"Test",

"DateOfBirth":"01.01.2000",

"Location":"Test"

}

}

]

}

Где ID - любое числовое значение.

Значение дня в поле DateOfBirth <13 (например, 01.01.2000).

3. Отправить запрос, получить в ответе "Created successfully".

4. Переключить метод на GET, в адрес ввести запрос [https://qa-course.kontur.host/api\_kontur\_practice1/api/users/{userId](https://qa-course.kontur.host/api_kontur_practice1/api/users/%7BuserId)}, подставив вместо [{userId](https://qa-course.kontur.host/api_kontur_practice1/api/users/%7BuserId)} значение ID введенное в Шаге 3.

ОР: В теле ответа появятся данные о созданном пользователе, совпадающие с ранее отправленными

ФР: В теле ответа появятся данные о созданном пользователе, в поле DateOfBirth дата отображается в формате Ticks.

**При добавлении пользователя с ранее занятым ID не возникает ошибки. Запрос успешно отправляется. Данные старого пользователя перезаписываются новыми.**

Условия

Стенд:<https://qa-course.kontur.host/api_kontur_practice3/>

Авторизация: Type - Basic Auth, login - admin, password - admin

Запрос <https://qa-course.kontur.host/api_kontur_practice3/api/users/create>

Тело запроса: вид raw, формат JSON

Шаги

1. Выбрать метод POST, в адрес ввести запрос <https://qa-course.kontur.host/api_kontur_practice3/api/users/create>

1. В тело запроса ввести

{

"ItemsOfClients":

[

{

"id": "99999",

"Person":

{

"Name":"Test",

"Surname":"Test",

"Patronymic":"Test",

"DateOfBirth":"01.01.2000",

"Location":"Test"

}

}

]

}

Где ID - положительное целое числовое значение (Например, 99999)

4. Отправить запрос. Отправить запрос, получить в ответе "Created successfully".

5. Не меняя значения ID в теле запроса, отправить его еще раз.

ОР: В теле ответа появятся сообщение об ошибке User already exists. UserId: \*введенное значение\*. Второй пользователь не добавится. Данные пользователя, который был добавлен первоначально, не изменятся.

ФР: В теле ответа появляются данные об успешном добавлении пользователя "Created successfully". Данные пользователя, который был добавлен первоначально, перезаписываются данными отправленными повторно.

**Редактирование пользователя. При попытке редактирования клиента в ответ на запрос PUT отображается ошибка 500 Internal Server Error**

Условия

Стенд:<https://qa-course.kontur.host/api_kontur_practice3/>

Авторизация: Type - Basic Auth, login - admin, password - admin

Запрос <https://qa-course.kontur.host/api_kontur_practice3/api/users/create>

Тело запроса: вид raw, формат JSON

Шаги

1. Выбрать метод POST, в адрес ввести запрос <https://qa-course.kontur.host/api_kontur_practice3/api/users/create>

2. В тело запроса ввести

{

"ItemsOfClients":

[

{

"id": "99999",

"Person":

{

"Name":"Test",

"Surname":"Test",

"Patronymic":"Test",

"DateOfBirth":"01.01.2000",

"Location":"Test"

}

}

]

}

Где ID - положительное целое числовое значение (Например, 99999)

4. Отправить запрос. Отправить запрос, получить в ответе "Created successfully".

5. Переключить метод на PUT. Адрес запроса поменять на [https://qa-course.kontur.host/api\_kontur\_practice1/api/users/{userId}/update](https://qa-course.kontur.host/api_kontur_practice1/api/users/%7BuserId%7D/update) ,где вместо [{userId}](https://qa-course.kontur.host/api_kontur_practice1/api/users/%7BuserId%7D/update) ввести ранее использованое значение ID.

6. Изменить данные в каком-либо поле в теле запроса. (Например, "Name":"Test" -> "Name":"Test2" )

7. Отправить запрос PUT.

ОР: Запрос успешно отправится в ответ придет сообщение об успешном обновлении данных пользователя и ответ 200.

ФР: В ответ приходит ошибка 500 Internal Server Error

**Удаление пользователя. Отправка запроса DELETE проходит успешно, но данные о пользователе, которые должны удалиться, сохраняются**

Условия

Стенд:<https://qa-course.kontur.host/api_kontur_practice3/>

Авторизация: Type - Basic Auth, login - admin, password - admin

Запрос <https://qa-course.kontur.host/api_kontur_practice3/api/users/create>

Тело запроса: вид raw, формат JSON

Шаги

1. Выбрать метод POST, в адрес ввести запрос <https://qa-course.kontur.host/api_kontur_practice3/api/users/create>

2. В тело запроса ввести

{

"ItemsOfClients":

[

{

"id": "88888",

"Person":

{

"Name":"Test",

"Surname":"Test",

"Patronymic":"Test",

"DateOfBirth":"01.01.2000",

"Location":"Test"

}

}

]

}

Где ID - положительное целое числовое значение (Например, 88888)

4. Отправить запрос. Отправить запрос, получить в ответе "Created successfully".

5. Переключить метод на DELETE. Адрес запроса поменять на [https://qa-course.kontur.host/api\_kontur\_practice3/api/users/{userId}/](https://qa-course.kontur.host/api_kontur_practice1/api/users/%7BuserId%7D/update)delete ,где вместо [{userId}](https://qa-course.kontur.host/api_kontur_practice1/api/users/%7BuserId%7D/update) ввести ранее использованое значение ID.

6. Отправить запрос DELETE, получить в ответе User deleted. UserId: \*ранее введенный ID\*

7. Переключить метод на GET. Адрес запроса поменять на [https://qa-course.kontur.host/api\_kontur\_practice3/api/users/{userId}/](https://qa-course.kontur.host/api_kontur_practice1/api/users/%7BuserId%7D/update) ,где вместо [{userId}](https://qa-course.kontur.host/api_kontur_practice1/api/users/%7BuserId%7D/update) ввести ранее использованое значение ID.

8. Отправить запрос GET.

ОР: В ответ приходит ошибка 404 и сообщение User not found. UserId: \*ранее введенный ID\*

ФР: Запрос выполняется успешно, приходит информация о пользователе

Скриншот:

<http://joxi.ru/8AnQeNyuN7WQx2> - Запрос DELETE успешно отправлен

<http://joxi.ru/KAgN4J3H5X7xz2> - Информация о пользователе и введеным для удаления ID продолжает приходить

**Есть возможность создать пользователя с типом данных в DateOfBirth / Дата рождения, отличным от типа «Дата»**

Условия

Стенд:<https://qa-course.kontur.host/api_kontur_practice3/>

Авторизация: Type - Basic Auth, login - admin, password - admin

Запрос<https://qa-course.kontur.host/api_kontur_practice3/api/users/create>

Тело запроса: вид raw, формат JSON

Шаги

1. Выбрать метод POST, в адрес ввести запрос<https://qa-course.kontur.host/api_kontur_practice3/api/users/create>

2. В тело запроса ввести

{

"ItemsOfClients":

[

{

"Person":

{

"Name":"Test",

"Surname":"Test",

"Patronymic":"Test",

"DateOfBirth":"Test",

"Location":"Test"

}

}

]

}

3. Отправитьзапрос POST.

ОР: В теле запроса появится ошибка о недопустимости ввода в поле DateOfBirth данных, несовпадающих с типом «Дата». Пользователь не добавится.

ФР: В теле ответа появляется сообщение об успешном добавлении пользователя. Данные о пользователе появляются в базе.

Скриншот: <http://joxi.ru/4AkM4ExijXYBJm>

**Есть возможность создать пользователя с типом данных в ID, отличным от типа «Число»**

Условия

Стенд:<https://qa-course.kontur.host/api_kontur_practice3/>

Авторизация: Type - Basic Auth, login - admin, password - admin

Запрос<https://qa-course.kontur.host/api_kontur_practice3/api/users/create>

Тело запроса: вид raw, формат JSON

Шаги

1. Выбрать метод POST, в адрес ввести запрос<https://qa-course.kontur.host/api_kontur_practice3/api/users/create>

2. В тело запроса ввести

{

"ItemsOfClients":

[

{

"id": "Test",

"Person":

{

"Name":"Test",

"Surname":"Test",

"Patronymic":"Test",

"DateOfBirth":"01.01.2000",

"Location":"Test"

}

}

]

}

3. Отправить запрос POST.

ОР: В теле запроса появится ошибка о недопустимости ввода в поле ID данных, несовпадающих с типом «Число». Пользователь не добавится.

ФР: В теле ответа появляется сообщение об успешном добавлении пользователя. Данные о пользователе появляются в базе.

Скриншот: <http://joxi.ru/BA08PbDi1PddYA>