Отборочное задание для соискателей на позицию Тестировщика ПО (QA jun)

Первая часть задания выполняется в Postman.

Загрузите решение в виде архива тестового файла с ответами на вопросы, скриншотами процесса и готовой коллекцией запросов Postman и пришлите ссылку на него.

**Подготовительный этап**

Авторизуйтесь на gorest.co.in и получите токен для аутентификации (http bearer tokens)

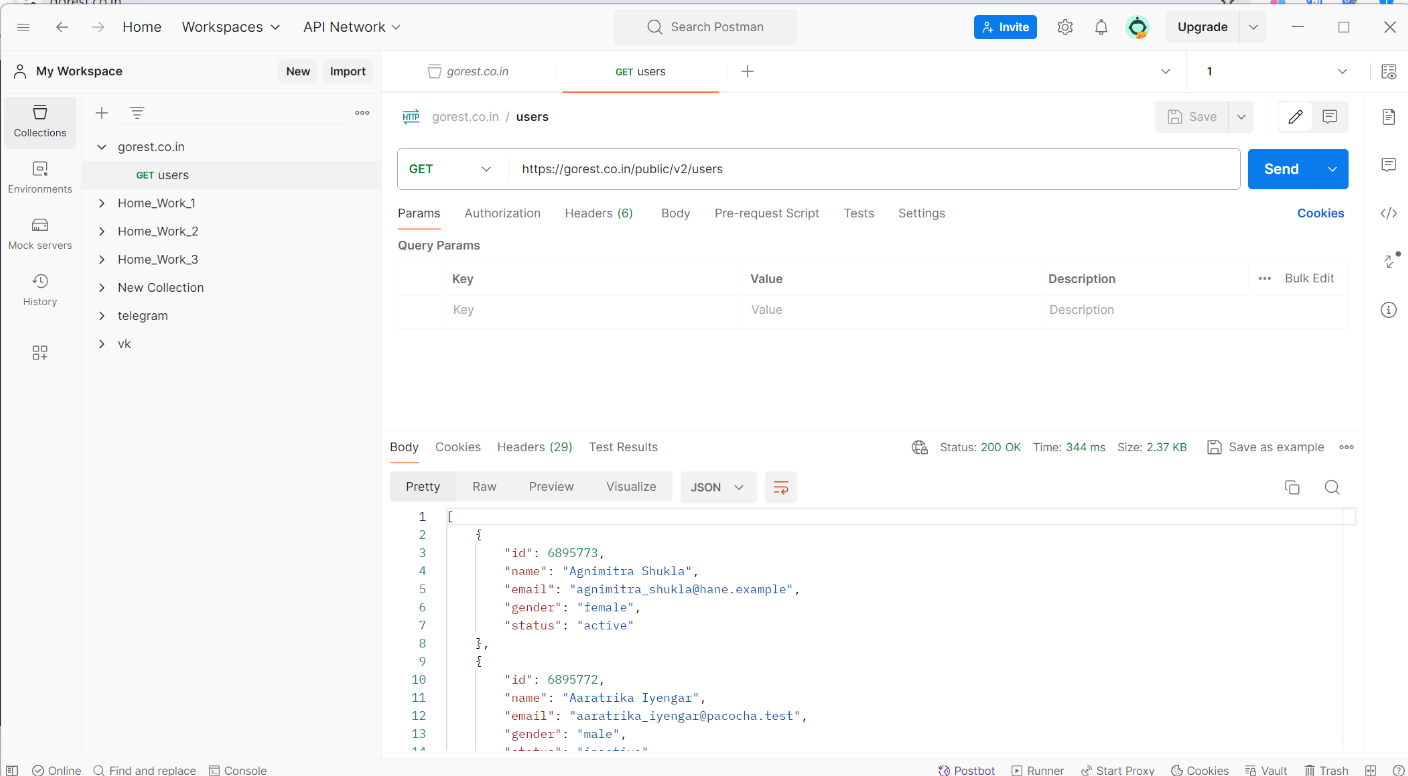
86800c26b3d46a9443419e5011b57bb4ba214d8f8906ae3eb649035e4b543f68

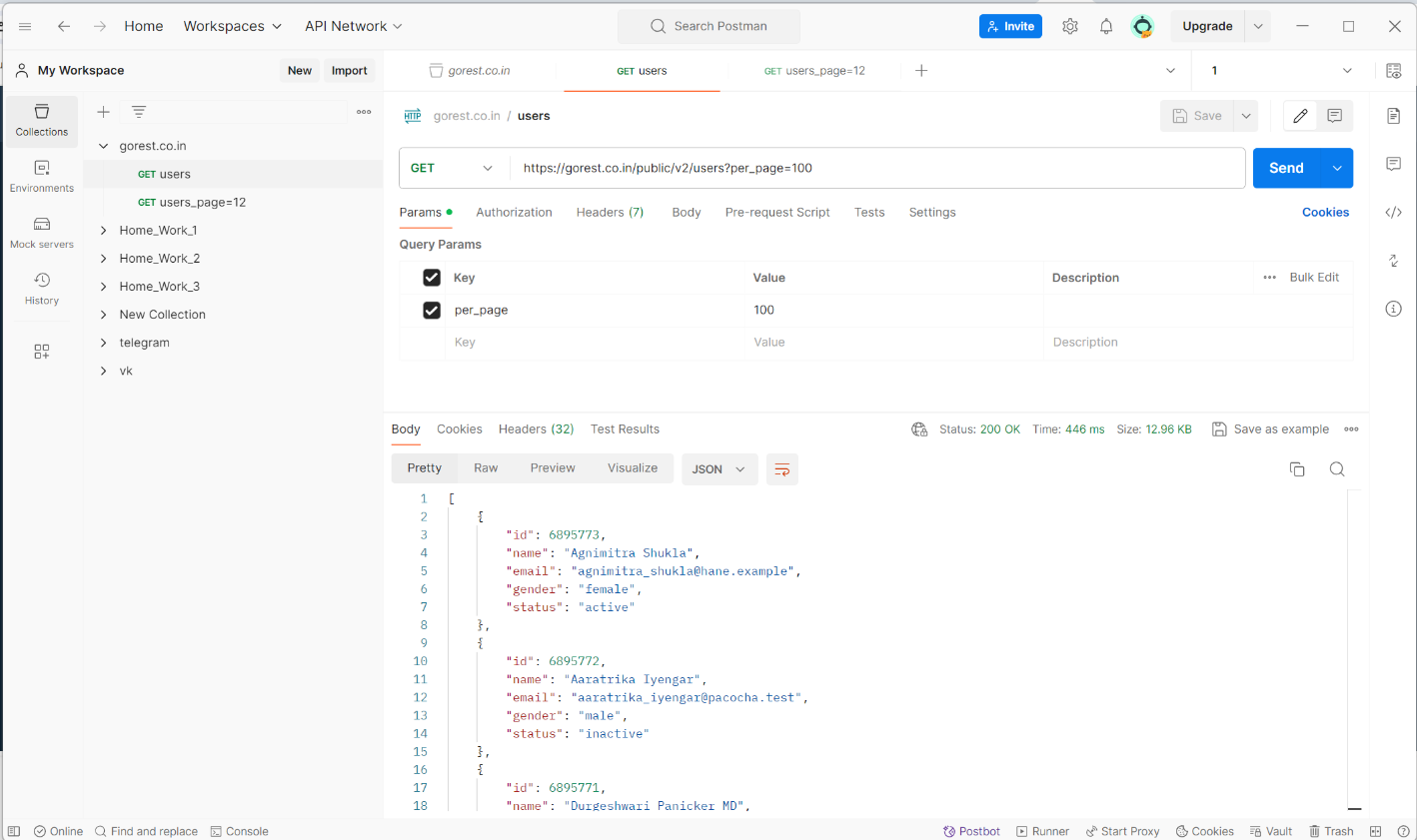
**Задание**

1. Получите список пользователей.  
Каким запросом получили список пользователей? - **https://gorest.co.in/public/v2/users**  
Какой статус ответа получили? - **200OK**

Что содержится в теле ответа? – содержится информация о 10 пользователях на 1-ой странице. Можно изменить количество пользователей на страницы, добавив параметр ?per\_page=100(максимум можно вывести 100 записей)

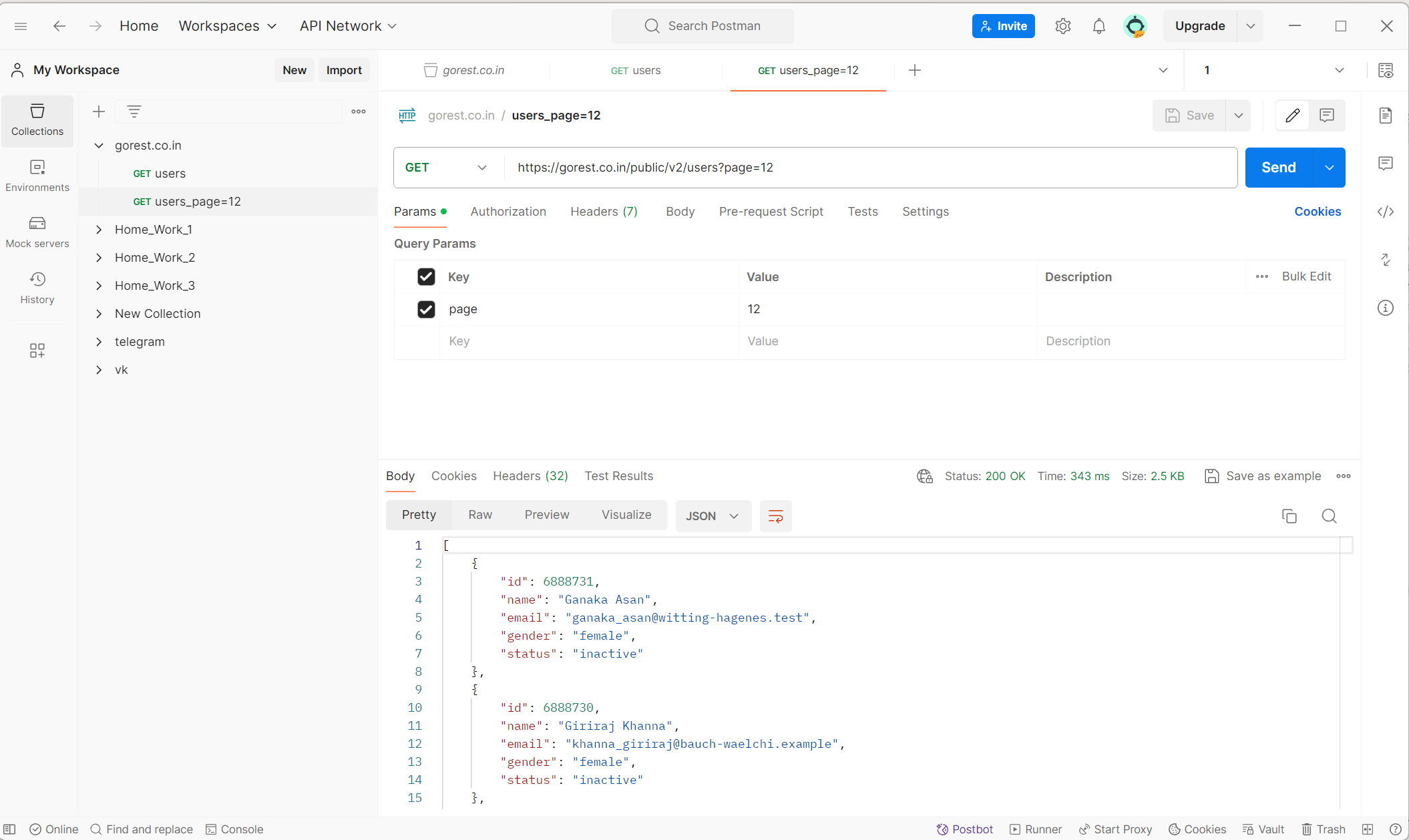
Сделайте скриншот:





2. Модифицируйте запрос и получите страницу 12. Что содержится в теле ответа? Сделайте скриншот.

В теле ответа содержится информация о 10 пользователях на 12-ой странице

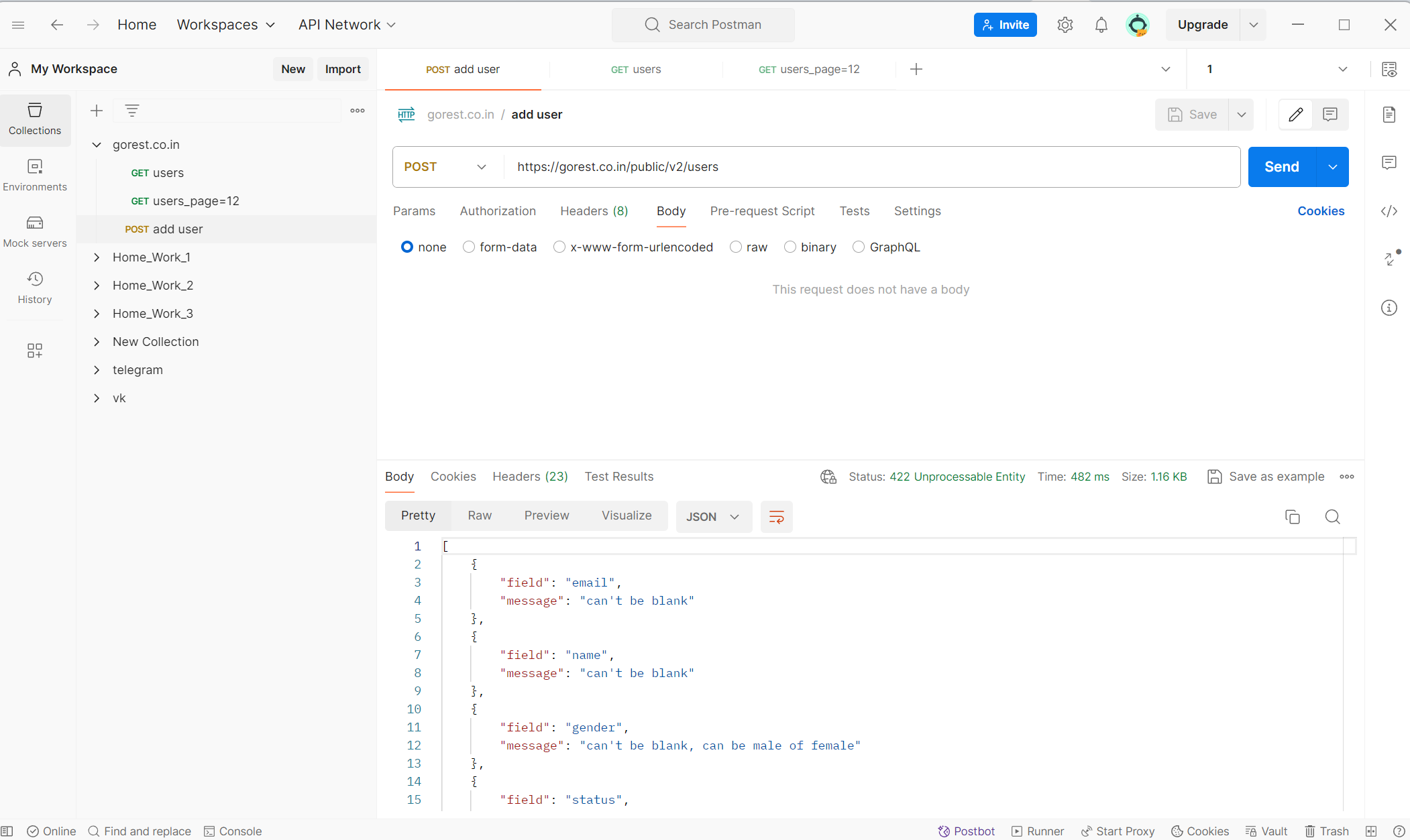


3. Какие отличия запроса в п.2 от запроса в п.1?

Использование параметра page

4. Сделайте post запрос для получения пользователя с пустым телом запроса. Какой ответ пришел и почему? Сделайте скриншот.

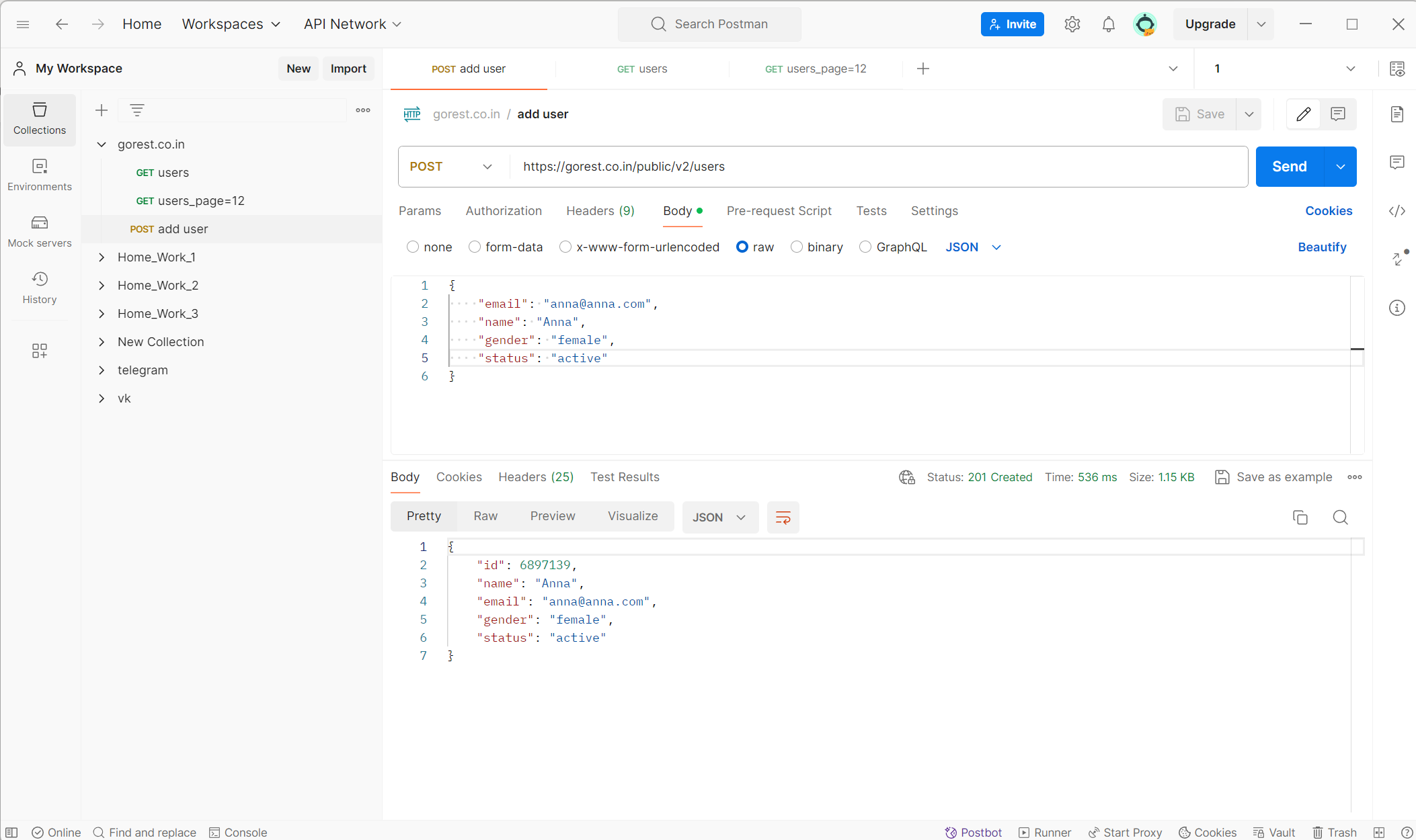
Ответ пришел 422, потому что не заполнены необходимые поля: email, name, gender, status



5. Напишите запрос для создания пользователя, используя метод raw => json. Какой ответ пришел? Сделайте скриншот.

Пришел ответ 201: A resource was successfully created in response to a POST request. The Location header contains the URL pointing to the newly created resource.

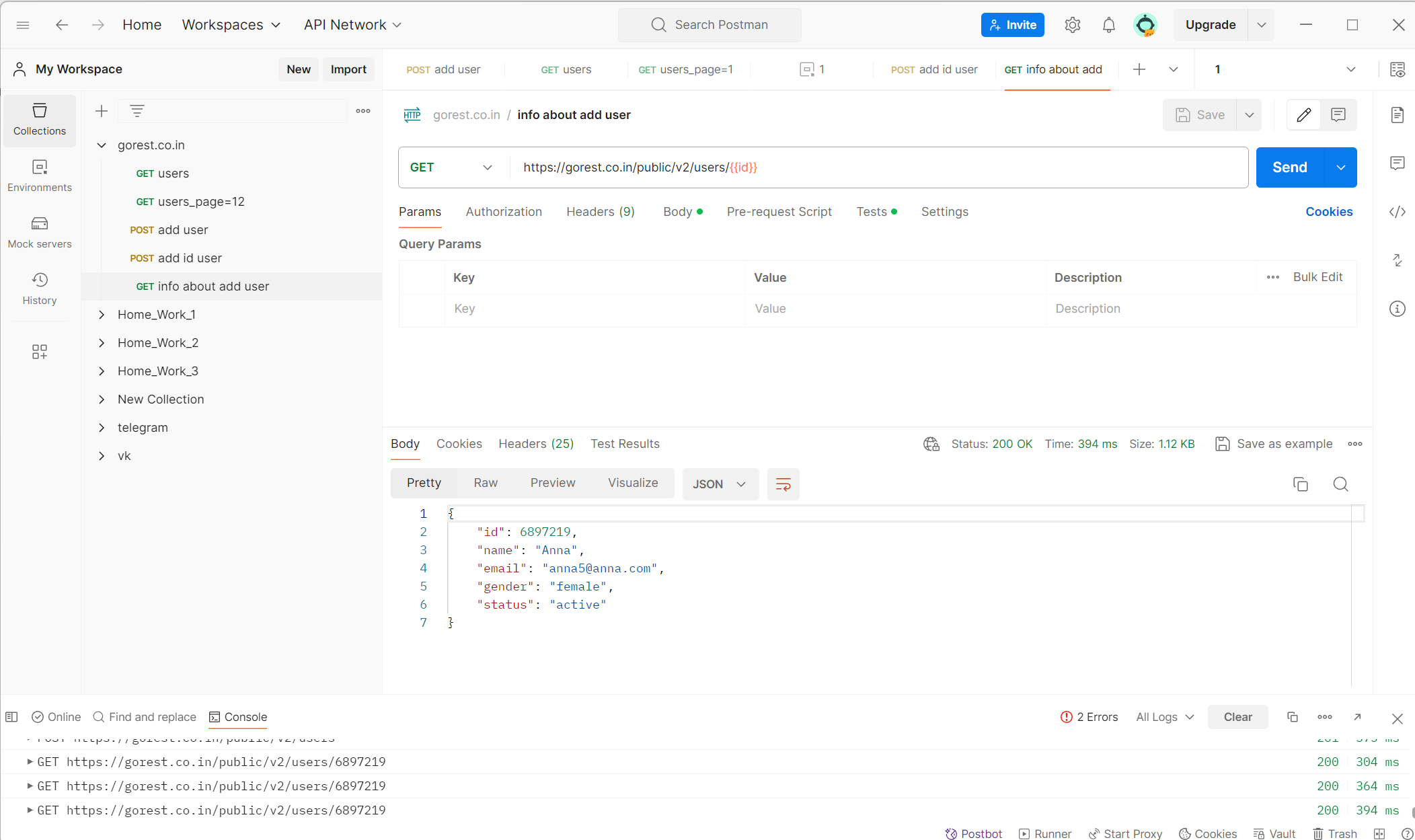
В теле ответа информация о только что добавленном пользователе.



6. Напишите запрос, возвращающий в ответе запись о созданном пользователе. Сделайте скриншот.

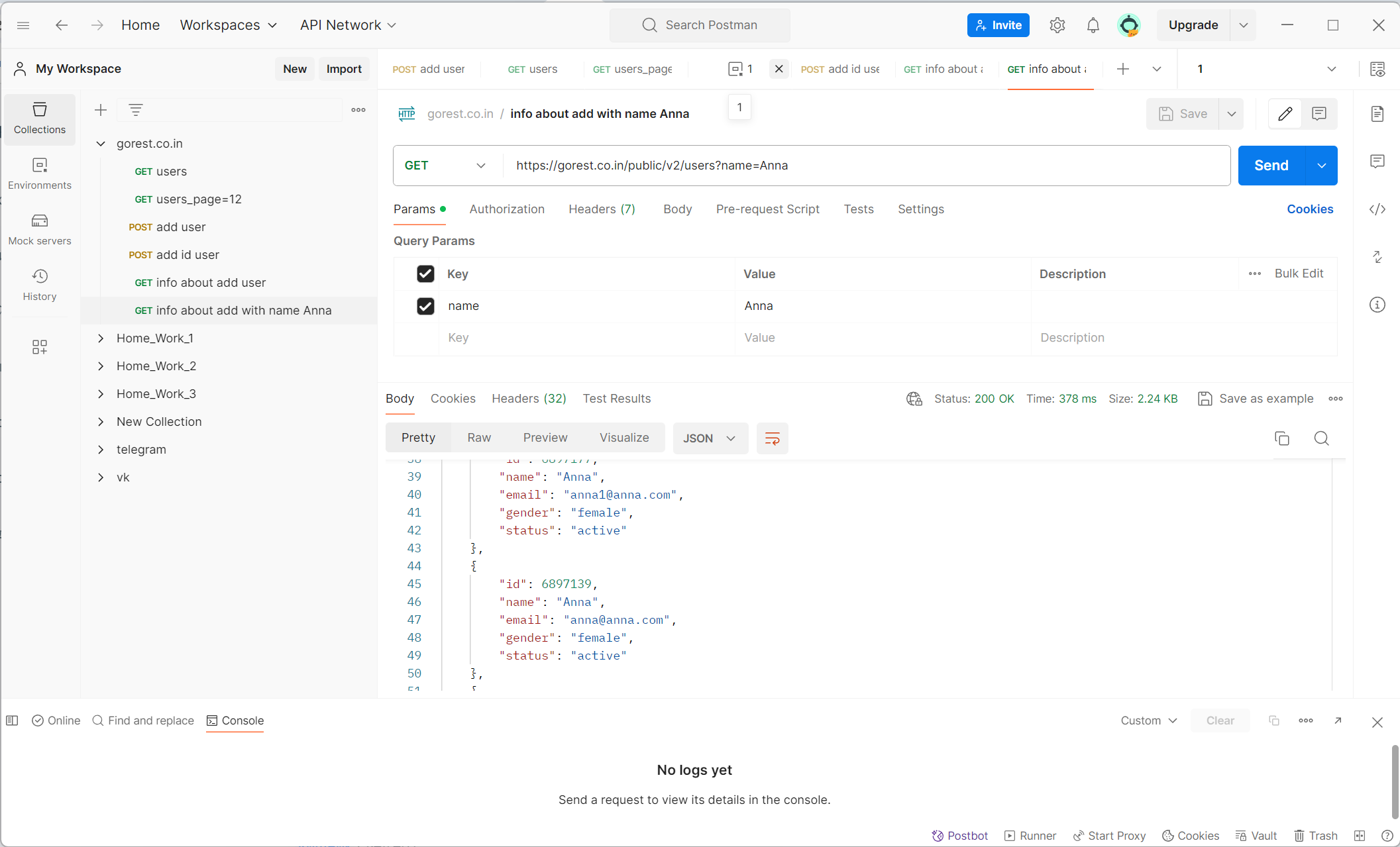
Добавляем в переменную id только что созданного пользователя.

Выводим информацию об этом пользователе:

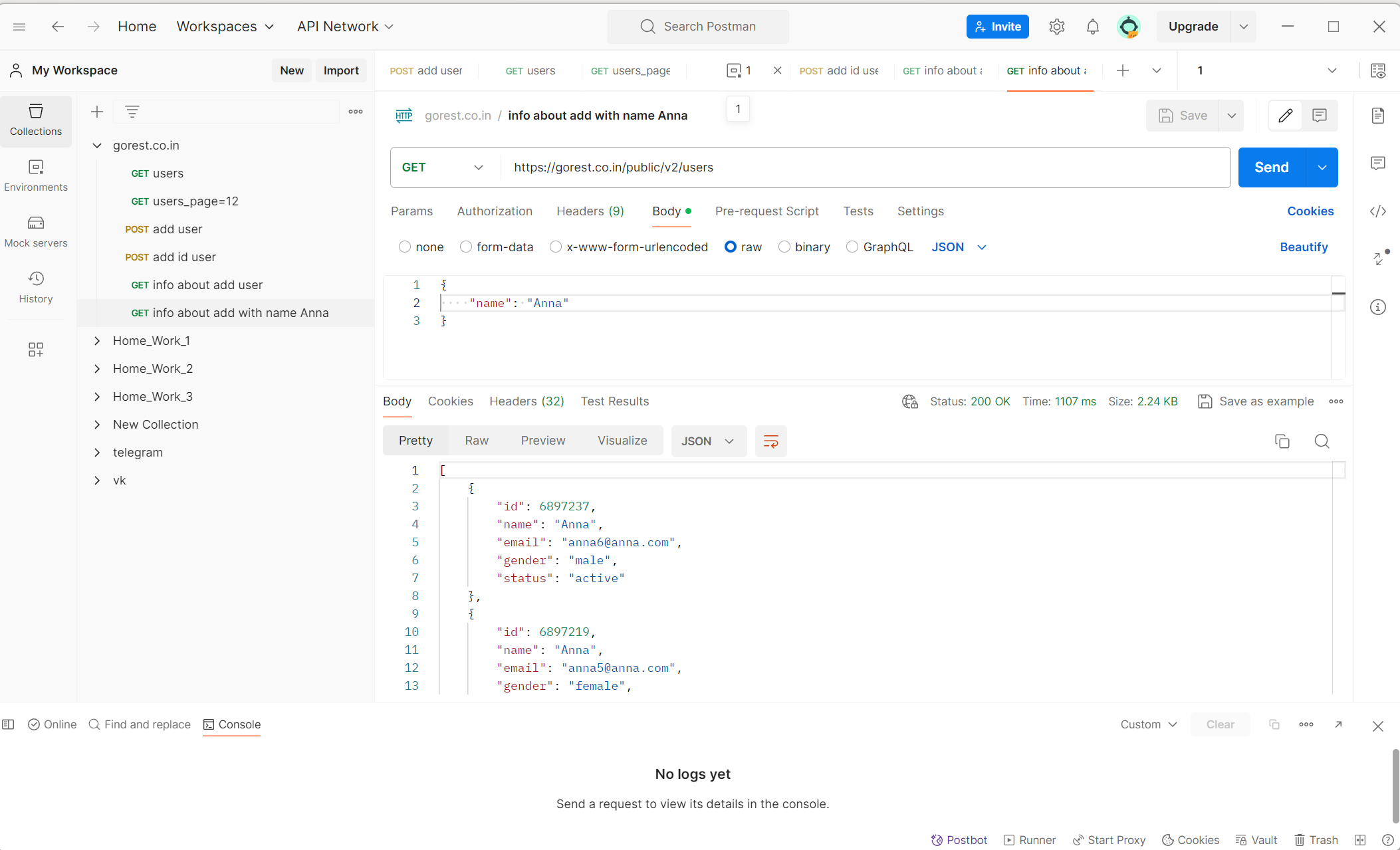


7. Создайте еще несколько пользователей с одинаковыми значениями “name”, но разными email. Получите список всех только что созданных пользователей. Сделайте скриншот.  
Какими ещё get запросами можно получить запись о пользователе?

Запросом <https://gorest.co.in/public/v2/users?name=Anna> выведется список пользователей, в поле name у которых содержится значение “Anna”

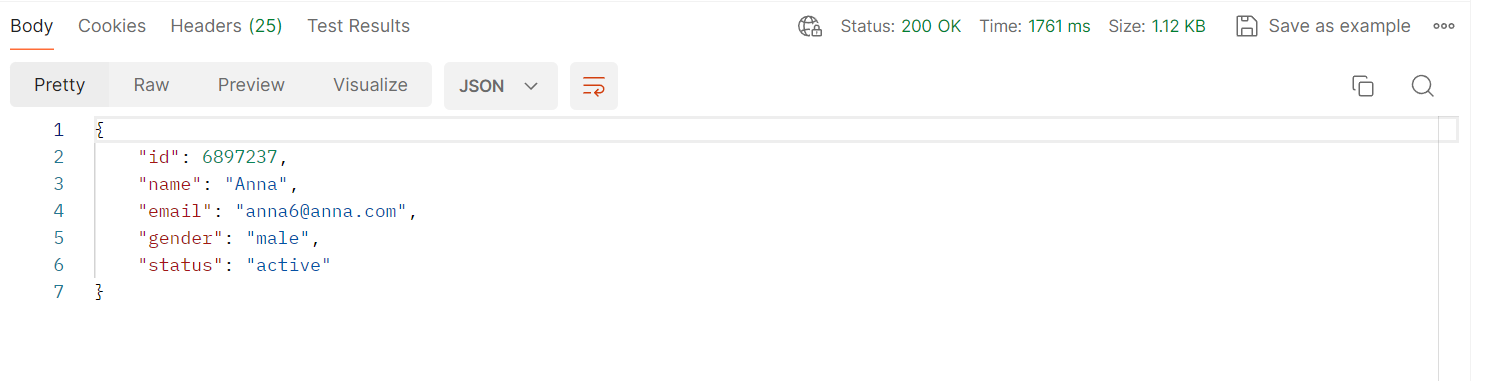


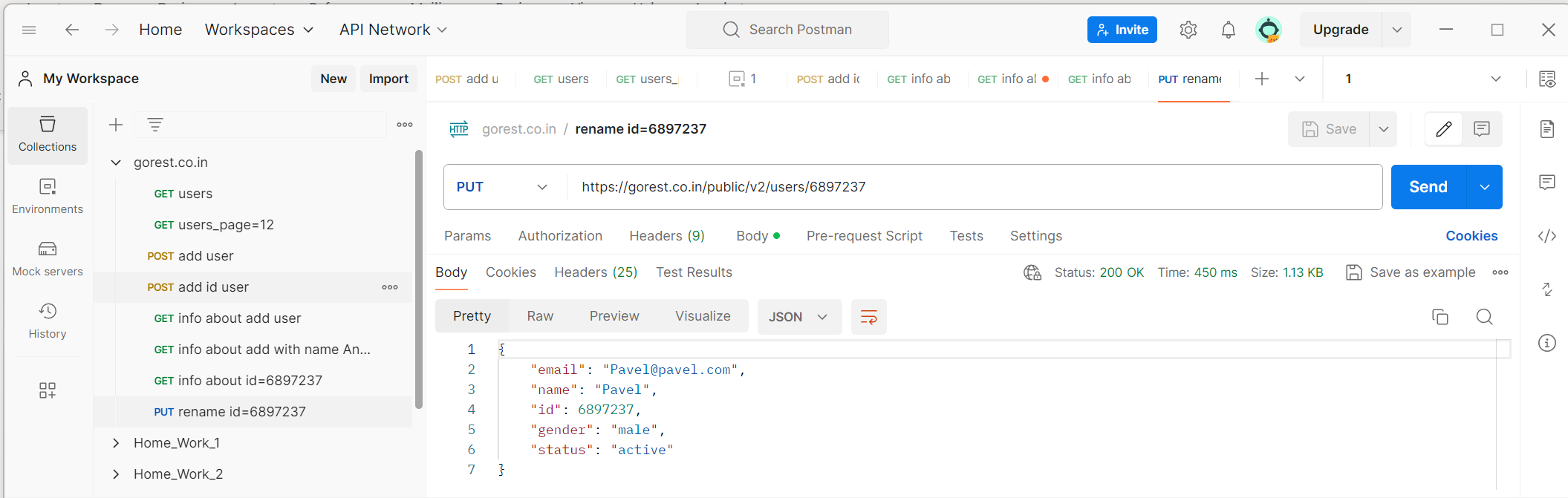
Либо можно записать параметр в body



8. Напишите запрос с методом put на изменение email и name на другое значение. Проверьте изменение get запросом. Сделайте скриншот.

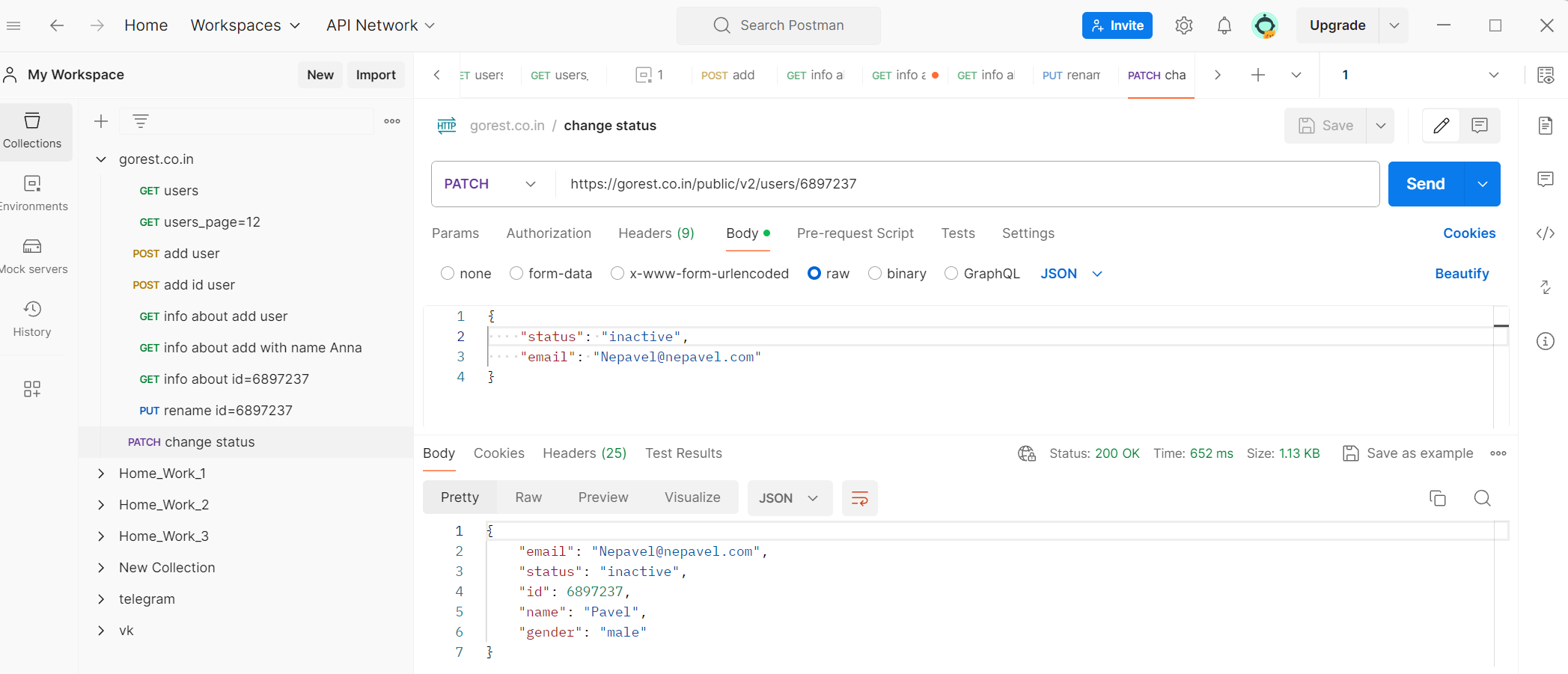
Первоначальное значение(id 6897237)





9. Чем отличаются запросы put и patch?

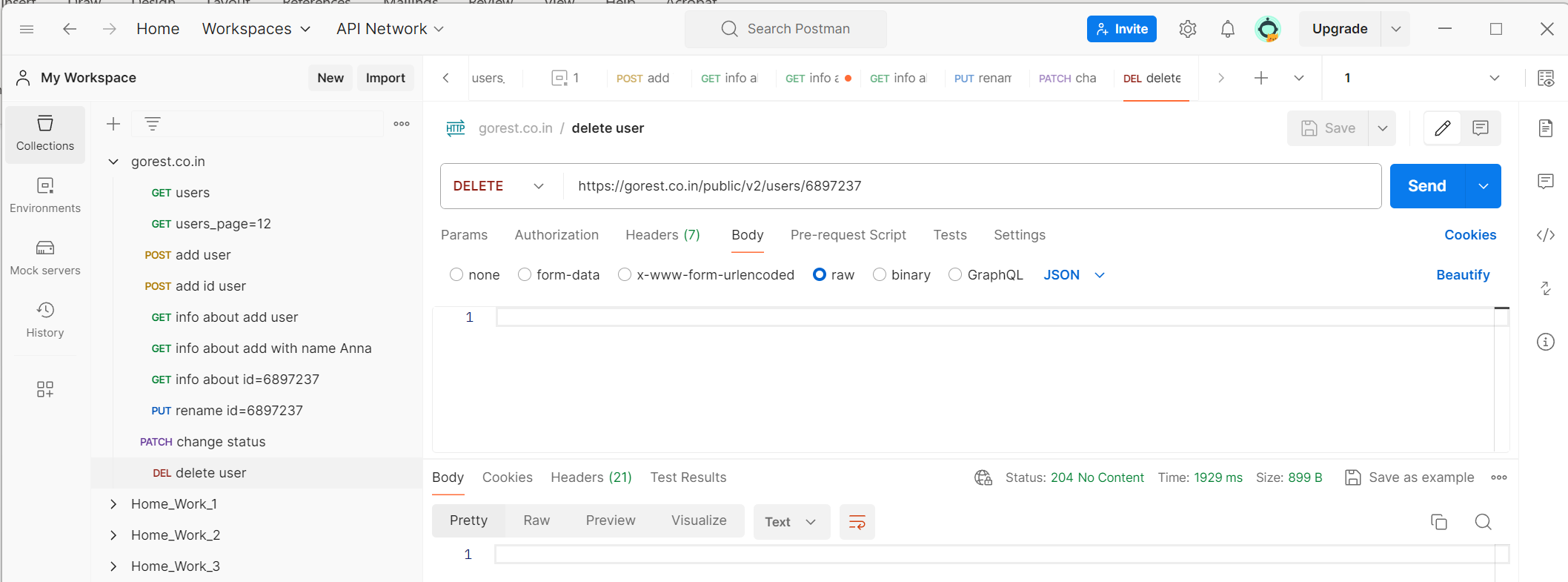
Метод PUT используется для обновления целого ресурса целиком, метод PATCH обновляет отдельные аттрибуты ресурса.  
Напишите запрос с методом patch для корректировки полей status и email. Сделайте скриншот.



В чем отличие полученного результата, от результата запроса из п.8?

В данном пример отличий нет, потому что здесь при использовании метода put ресурс переписывается не полностью, а частично.

10. Удалите пользователя, созданного в п.5. Что содержится в теле ответа? Сделайте скриншот.



В телt ответа ничего не содержится  
Выполните get запрос удаленного пользователя. Какой ответ пришел и почему? Сделайте скриншот.

Ответ – 204 - The request was handled successfully and the response contains no body content (like a DELETE request)

11. Удалите всех ранее созданных пользователей.

12. Сохраните коллекцию получившихся запросов в Postman.

**Теоретическое задание.**

Кратко ответьте на вопросы ниже.

1. **Опишите общий процесс тестирования отдельного модуля SPA приложения.**

SPA - single page application, то есть одностраничное приложение.

В первую очередь, как и при тестировании любого приложения, я бы ознакомилась с документацией и требованием к продукту. Я бы протестировала непосредственно функции в модуле(Unit тестирование/обычно их пишут программисты, но иногда и тестировщики могут покрыть их автотестами), чтобы убедиться, что они изолированно друг от друга работают корректно. Потом необходимо протестировать взаимодействие функций в модуле. Проверить, как они обрабатывают входные данные и соответствует ли выходные данные ожидаемому результату.

1. **Что должен включать репорт найденной ошибки, какую структуру должен иметь?**

Баг репорт должен включать себя:

* Название, которое должно отвечать на вопрос Что? Где? Когда?
* Шаги для воспроизведения
* Фактический результат
* Ожидаемый результат
* Вложения( скрин, видео, логи)
* Окружение
* Приоритет
* Серьезность

1. **Как будет выглядеть баг-репорт, если, к примеру, не работает электрический чайник?**

Название: не включается чайник при подключении к сети 220В. Серьезность: высокая.

Приоритет: высокий.

Окружение: питьевая вода без видимых примисей, розетка 220В.

Шаги воспроизведения: 1. налить воду в чайник от минимальной до максимальной отметки; 2. включить вилку в розетку; 3. нажать на кнопку включить;  
Фактический результат: индикатор не загорается, вода не начинает бурлить

Ожидаемый результат: через несколько минут вода начинает кипеть, кнопка включение автоматичсеки выключается

1. Составить баг репорт страниц и разделов, включая корневой по иерархии сайта <https://site.domain.ru>

1. Аттрибут lang указан “EN” в devtools

Шаги:

1.Зайти на сайт <https://site.domain.ru/>

2. Открыть devtools

Ожидаемый результат:

1. Аттрибут lang указан как “RU”

Фактический результат:

1. Аттрибут lang указан как “EN”

Окружение: Windows 11, Google Chrome Version 124.0.6367.119

1. Необходимо протестировать регистрацию нового пользователя в приложении. Расскажите, как вы будете это делать? (описать все этапы)

1. Регистрация с валидными данными

2. Попытка регистрации с невалидными данными

3. Попытка регистрации без заполнения обязательных полей

4. Отправка пустой формы

В задании не указано, как происходит подтверждение регистрации. Должно ли на почту приходить письмо или смс на указанный номер.

1. Представьте ситуацию, что у разрабатываемого приложения еще нет интерфейса, но реализован REST API. Разработчик просит вас создать какую-то сущность в базе и проверить, что она создалась с нужными параметрами. Опишите ход ваших действий в данной ситуации: что и как вы бы проверили, опираясь на имеющееся описание API, с указанием конкретики (название типов запросов и т.д.)

Используя информацию из документации, я бы создала запрос на создание сущности с учетом нужных параметров. Обычно для создания сущности используется HTTP метод POST, и в теле запроса передаются параметры новой сущности в формате JSON или форматах, указанных в документации. Вначале отправила бы запрос со всеми обязательными полями, проверила бы в базе, что запись сохранилась. При успешной создании сущности сервер должен вернуть код 201(Created).

Далее попыталась бы отправить запрос без заполнения обязательных полей, чтоб убедиться, что валидация проверяется и сущность в базе данных не создается, сервер возвращает ошибку 400(Bad Request).

Проверка содержимого ответа на запрос(метод Get). В ответе сервер олжен вернуть созданную сущность.

При необходимости я бы повторила процесс создания сущности несколько раз с разными наборами параметров, чтобы удостовериться в корректной работе системы.

1. Поле username должно быть обязательным, но оно не обязательно. Приведите пример баг-репорта, созданного для этой ошибки.

Название: Происходит создание сущности при отправке запроса с незаполненным полем username

Шаги: ввести значения во все необходимые поля, но поле username оставить пустым

Ожидаемый результат: ошибка 400 при отправке запроса, в базе данных новая сущность не создается

Фактический результат: в БД создана новая сущность

1. Можно ли передавать в json бинарные данные? Почему?

В json передаются текствовые данные, а последовательность нулей и единиц может быть передана в json как строка текста, но с условием, что программа ее переведет в бинарный вид, т.е. должна быть кодирована для преобразования бинарных данных в текстовый формат, который можно передать в JSON.

1. Написать тестовые наборы данных для поля ввода даты, которое отсеивает пользователей в возрасте до 18 лет.

Ввод даты, когда пользователю ровно 18 лет

Ввод даты, когда пользователю без одного дня 18 лет

Ввод даты, когда пользователю 18 лет и 1 день

Ввод даты, когда пользователю 35 лет

Ввод даты, когда пользователю 5 лет

Ввод даты, когда пользователю 0 лет

Ввод текущей даты

Ввод даты в будующем

Дату не вводить, поле оставить пустым

1. Как протестировать скачивание файла при недостаточном количестве места на диске.

1. Создать файл большого размера на сервере.

2. Убедиться, что на диске недостаточно свободного места для сохранения этого файла.

3. Послать запрос на сервер для скачивания файла.

4. Во время скачивания должно появиться сообщение об ошибке, указывающее на недостаточное количество свободного места на диске.

5. Проверить, что файл не был сохранен на диск из-за ошибки с недостаточным местом, проверить, что было показано сообщение об этом

6. Освободить немного места на диске и повторите попытку скачивания файла для проверки, что он успешно сохраняется после этого.

Можно использовать, например, Docker, для создания ситуации, когда не хватает места на диске или на сервере.

1. Как протестировать загрузку файла при недостаточном количестве места на сервере.

При загрузке файла большего размера, чем есть на сервере, я бы ожидала, что система покажет об этом ошибку. Следовательно, проверила бы количество фактически свободного места на сервере и размер файла, убедилась бы, что места меньше, чем размер файла и попробовала бы загрузить его.

Ожидаемый результат: показывается сообщение об ошибке, что файл не был загружен, на сервере новых файлов не появилось.

Далее я бы уменьшила файл до размеров свободного места на сервере и попыталась бы загрузить его, чтоб проверить, что загрузка работает.

1. Есть веб-страница с полями: e-mail, password и кнопкой submit. Необходимо привести примеры отрицательных тест-кейсов, по которым можно проверить эту страницу.

1. В поле email не вводить @

2. Начать заполнять поле email с @

3. В поле email два раза зиспользовать @

4. В поле email не вводить .

5. Поле email оставить пустым

6. Ввод e-mail с недопустимыми символами, например, пробелами или специальными символами, не разрешенными в адресе электронной почты.

7. В задании ничего не сказано про ограничения на поля пароль. Обычно нужно использовать от 8 до 18 символов, маленькие и большие буквы и обязательно наличие одного спец символа и одной цифры. Следовательно, буду использовать классы эквивалентности, граничные значения, и попробую ввести пароль меньше минимально разрешенного количества символов, больше разрешенного, не содержащего цифры, буквы, специальные символы

8. Поле пароль оставлю пустым

13. Имеется функционал калькулятора, доступный через веб браузер по ссылке. Он имеет только функцию делить, так сказать, MVP-версию. Диапазоны для вписывания в числитель и делитель от 0,1 до 99,9. Вывод значения происходит автоматически, потому что front-end реализован на React JS. Как вы будете тестировать этот функционал? Какие виды тестирования примените? Какие техники тест-дизайна используете?

Применю классы-эквивалентности вместе с граничными значениями:

- числа от 0,1 до 99,9

- числа меньше, чем 0,1 и больше, чем 99,9(негативные проверки)

- буквы, пробел и специальные символы(негативные проверки)

1. Проверю, что при делении 0,1 на 99 получается результат 0.001010101010101

2. При делении 1 на 99,9 получается результат 0.01001001001001

3. Проверю, что в форму для ввода числителя и делителя нельзя вписать буквы, специальные символы, и числа, меньше, чем 0,1 и больше, чем 99,9

Проверю, что результат полностью отображается на одной строке, вводимые цифры хорошо читаются и не сливаются с формой.(UI тестирование)