# Задание

## 1. Получите список пользователей. Каким запросом получили список пользователей? Какой статус ответа получили? Что содержится в теле ответа? Сделайте скриншот.

Запрос: GET <https://gorest.co.in/public/v2/users>

Статус ответа: 200

Тело ответа:

[

{

"id": 6823682,

"name": "Harinarayan Iyengar",

"email": "iyengar\_harinarayan@larkin.example",

"gender": "male",

"status": "inactive"

},

{

"id": 6823681,

"name": "Bhramar Kaur",

"email": "kaur\_bhramar@becker.test",

"gender": "female",

"status": "inactive"

},

{

"id": 6823680,

"name": "Bhoopati Mahajan",

"email": "mahajan\_bhoopati@homenick.test",

"gender": "male",

"status": "inactive"

},

{

"id": 6823678,

"name": "Ekaksh Nambeesan",

"email": "ekaksh\_nambeesan@bruen-wilderman.example",

"gender": "male",

"status": "active"

},

{

"id": 6823677,

"name": "Balgopal Dwivedi",

"email": "dwivedi\_balgopal@kulas-spinka.example",

"gender": "male",

"status": "inactive"

},

{

"id": 6823676,

"name": "Harita Achari",

"email": "achari\_harita@frami.test",

"gender": "male",

"status": "active"

},

{

"id": 6823675,

"name": "Gandharva Somayaji",

"email": "gandharva\_somayaji@champlin.example",

"gender": "female",

"status": "active"

},

{

"id": 6823674,

"name": "Deeptendu Mehra V",

"email": "v\_mehra\_deeptendu@windler.test",

"gender": "female",

"status": "active"

},

{

"id": 6823673,

"name": "Sen. Chandramauli Menon",

"email": "menon\_chandramauli\_sen@willms.example",

"gender": "male",

"status": "inactive"

},

{

"id": 6823672,

"name": "Kamala Namboothiri",

"email": "namboothiri\_kamala@paucek.example",

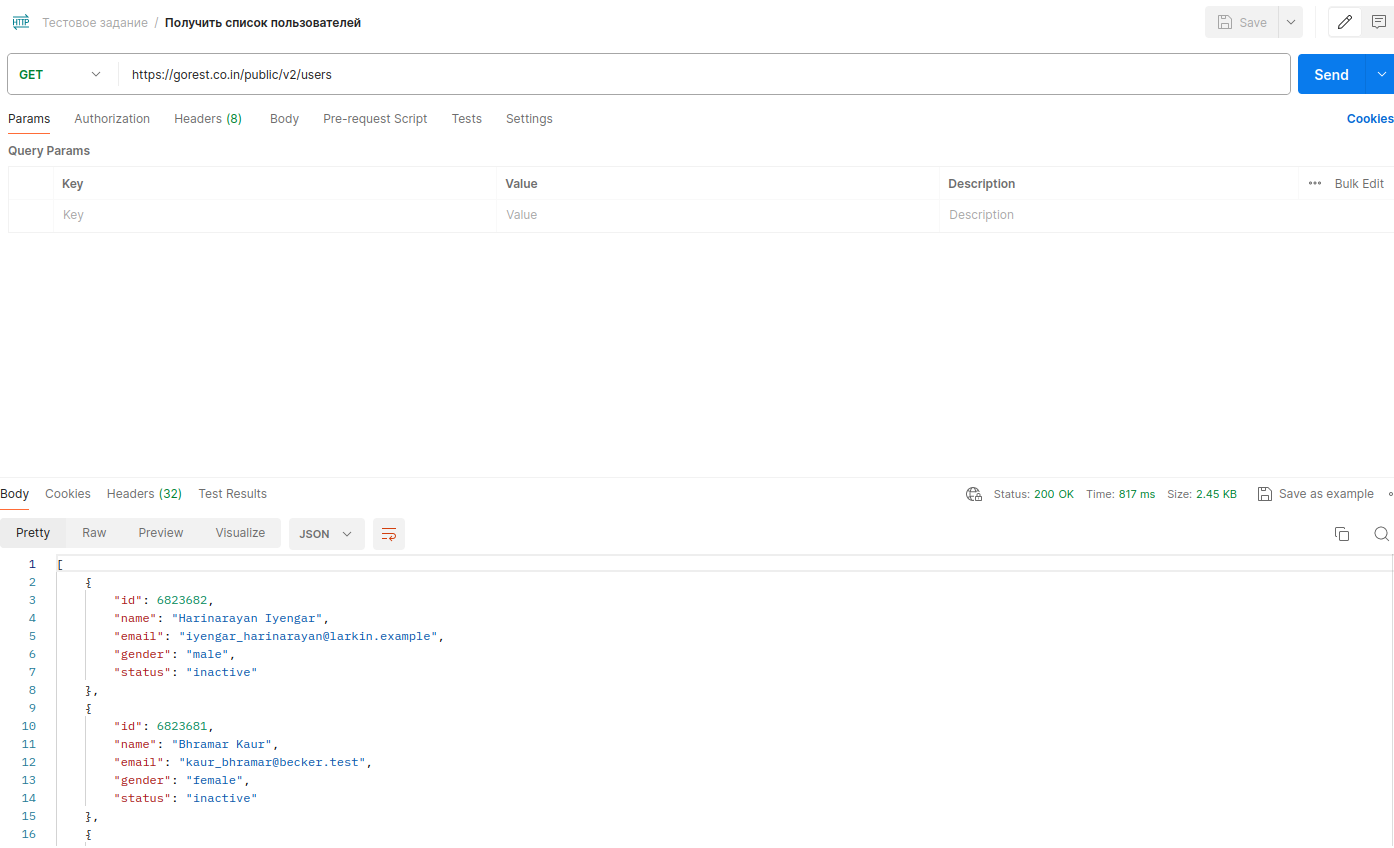
"gender": "male",

"status": "inactive"

}

]

Скриншот:

****

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

Тело ответа:

[

{

"id": 6823485,

"name": "Bakula Pothuvaal",

"email": "bakula\_pothuvaal@dooley.test",

"gender": "female",

"status": "inactive"

},

{

"id": 6823484,

"name": "Vrund Bharadwaj",

"email": "vrund\_bharadwaj@daugherty-kutch.example",

"gender": "male",

"status": "inactive"

},

{

"id": 6823483,

"name": "Amb. Ujjawal Prajapat",

"email": "amb\_prajapat\_ujjawal@heidenreich.test",

"gender": "male",

"status": "inactive"

},

{

"id": 6823482,

"name": "Akshaj Nair",

"email": "akshaj\_nair@denesik-feest.test",

"gender": "female",

"status": "active"

},

{

"id": 6821483,

"name": "Ms. Jay Nayar",

"email": "nayar\_jay\_ms@rath-mcclure.example",

"gender": "female",

"status": "active"

},

{

"id": 6821482,

"name": "Vidhya Mukhopadhyay",

"email": "vidhya\_mukhopadhyay@abshire-jerde.example",

"gender": "male",

"status": "active"

},

{

"id": 6821481,

"name": "Vidhya Khatri",

"email": "vidhya\_khatri@johnson.example",

"gender": "female",

"status": "active"

},

{

"id": 6821480,

"name": "Sen. Chandrabhaga Ganaka",

"email": "chandrabhaga\_sen\_ganaka@hegmann.example",

"gender": "female",

"status": "active"

},

{

"id": 6821479,

"name": "Prem Khan",

"email": "prem\_khan@langworth.test",

"gender": "male",

"status": "inactive"

},

{

"id": 6821478,

"name": "Tara Mishra",

"email": "mishra\_tara@douglas.test",

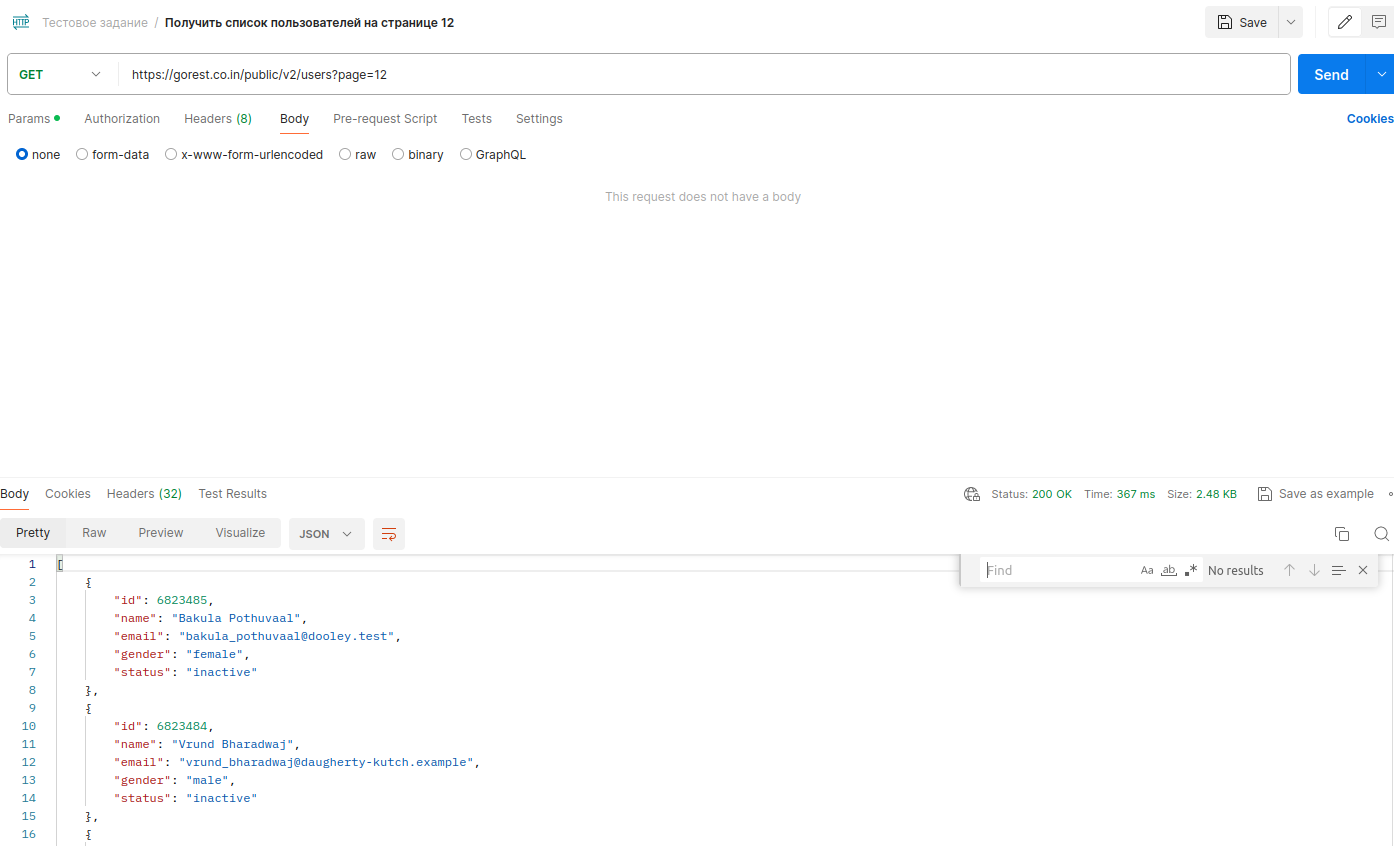
"gender": "male",

"status": "inactive"

}

]

Скриншот:



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

Запрос в п.2 содержит параметр page равный 12.

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

Ответ запроса:

[

{

"field": "email",

"message": "can't be blank"

},

{

"field": "name",

"message": "can't be blank"

},

{

"field": "gender",

"message": "can't be blank, can be male of female"

},

{

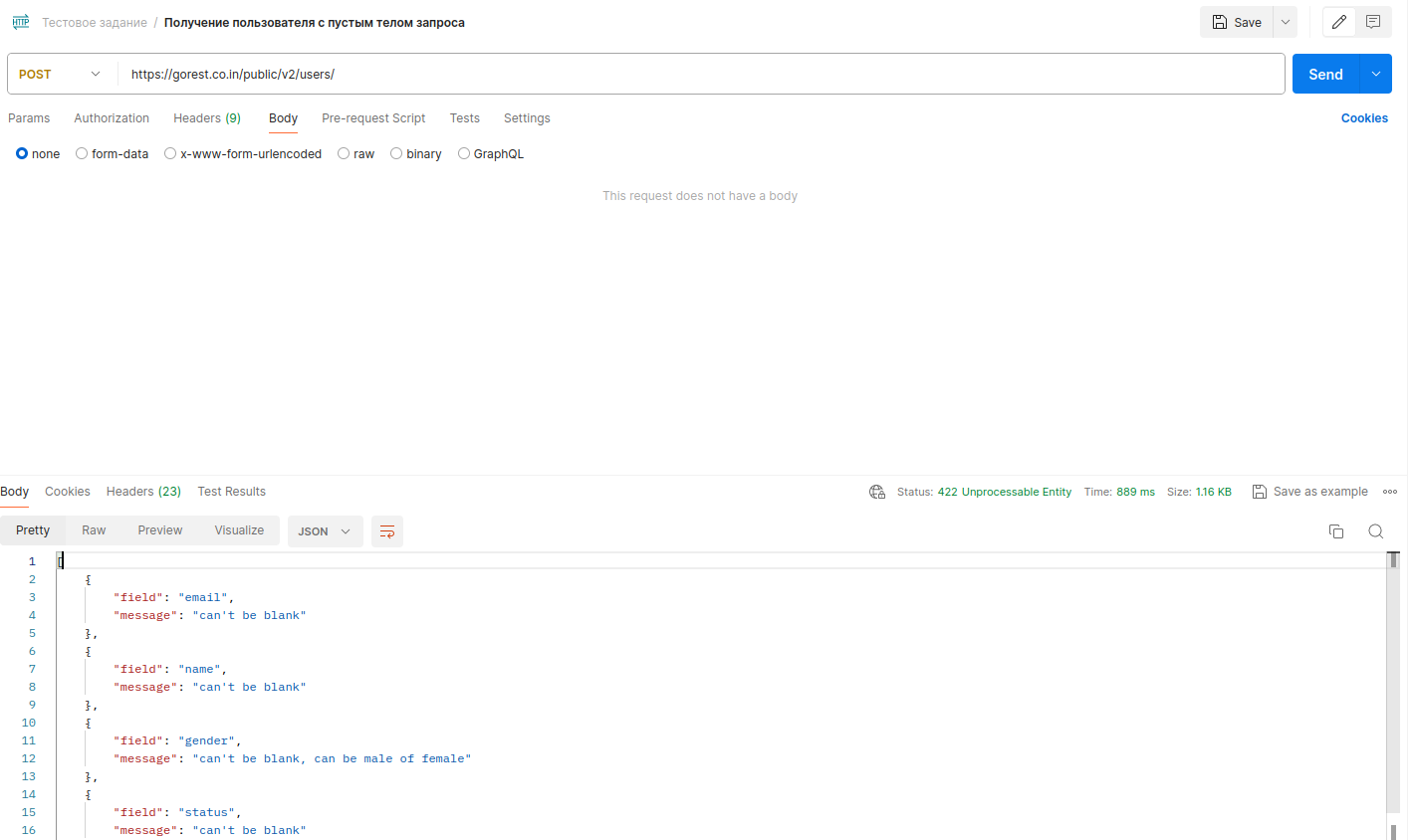
"field": "status",

"message": "can't be blank"

}

]

Скриншот:

****

Как можно понять из статуса ответа (422), сервер не смог обработать запрос, так как не указаны необходимые параметры тела запроса: email, name, gender и status.

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

Ответ запроса:

{

"id": 6825578,

"name": "Ignacio.Wuckert",

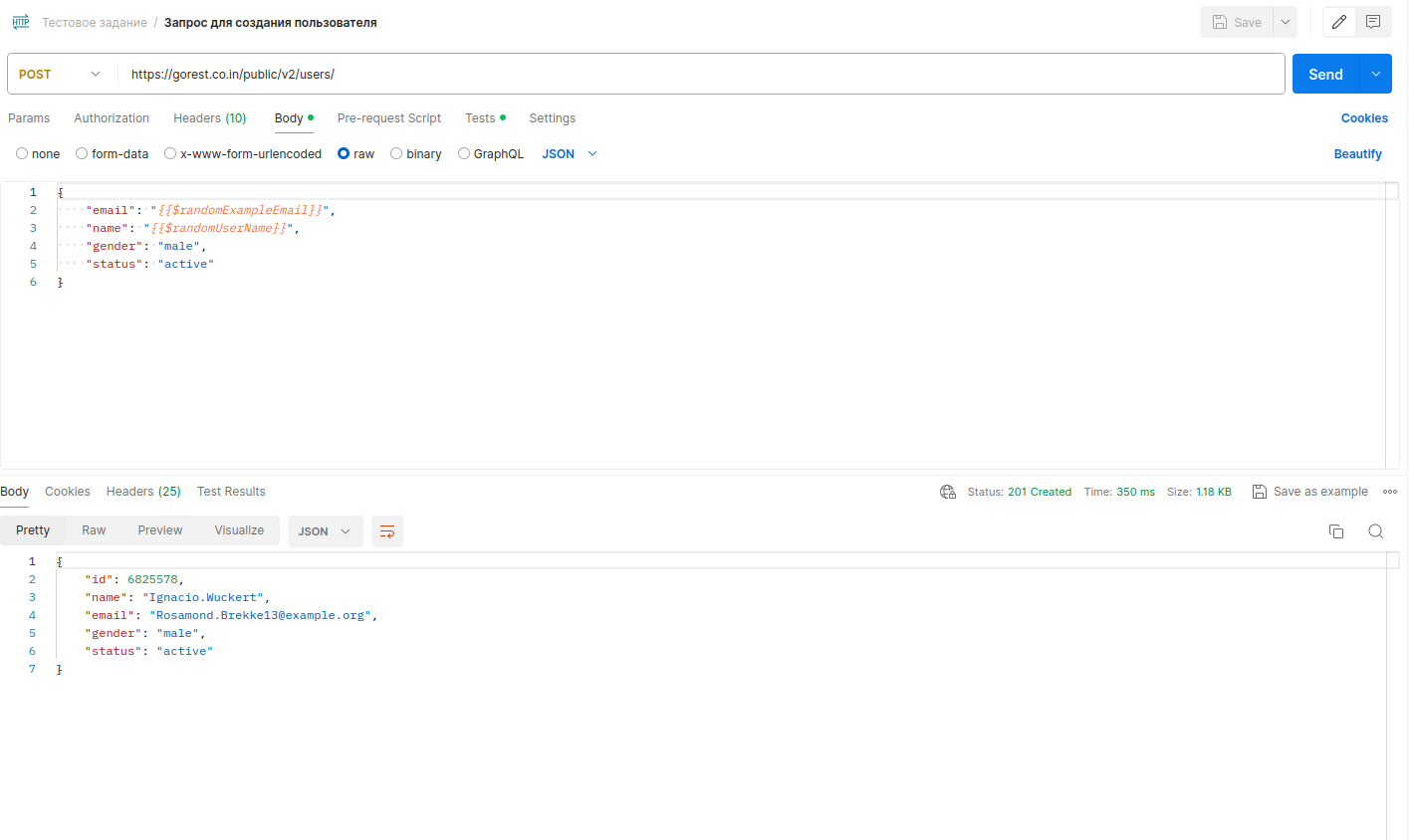
"email": "Rosamond.Brekke13@example.org",

"gender": "male",

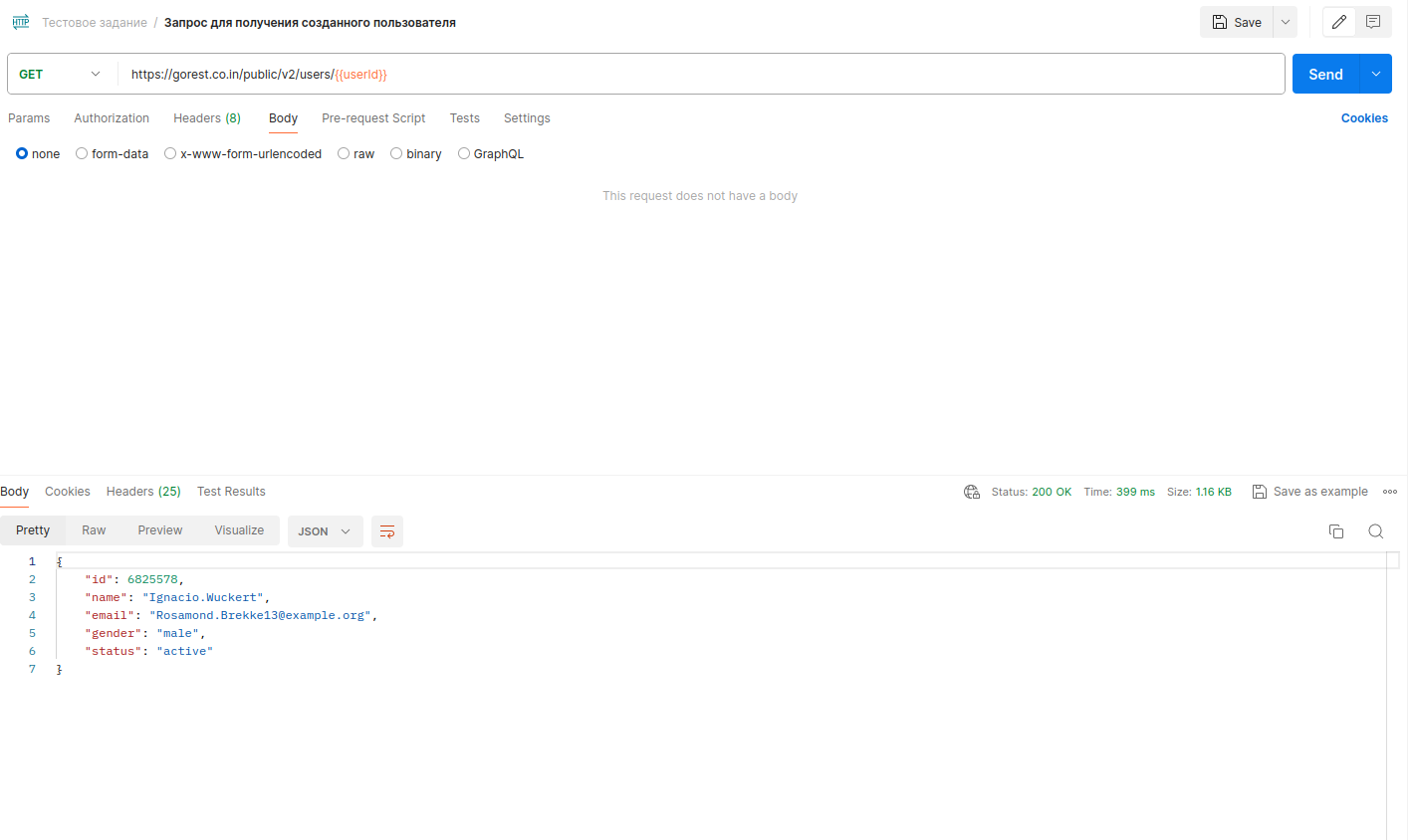
"status": "active"

}

Скриншот:

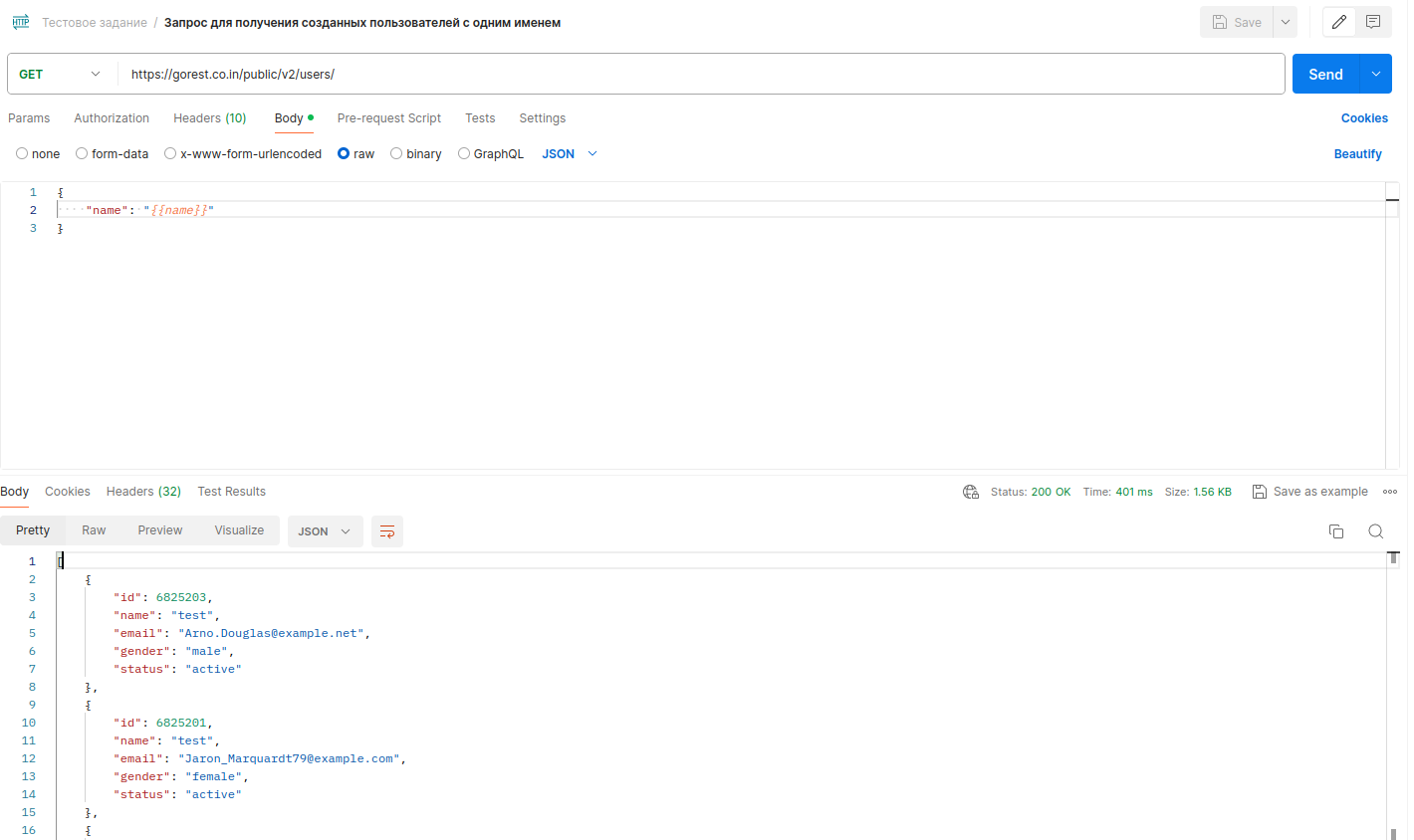
****

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

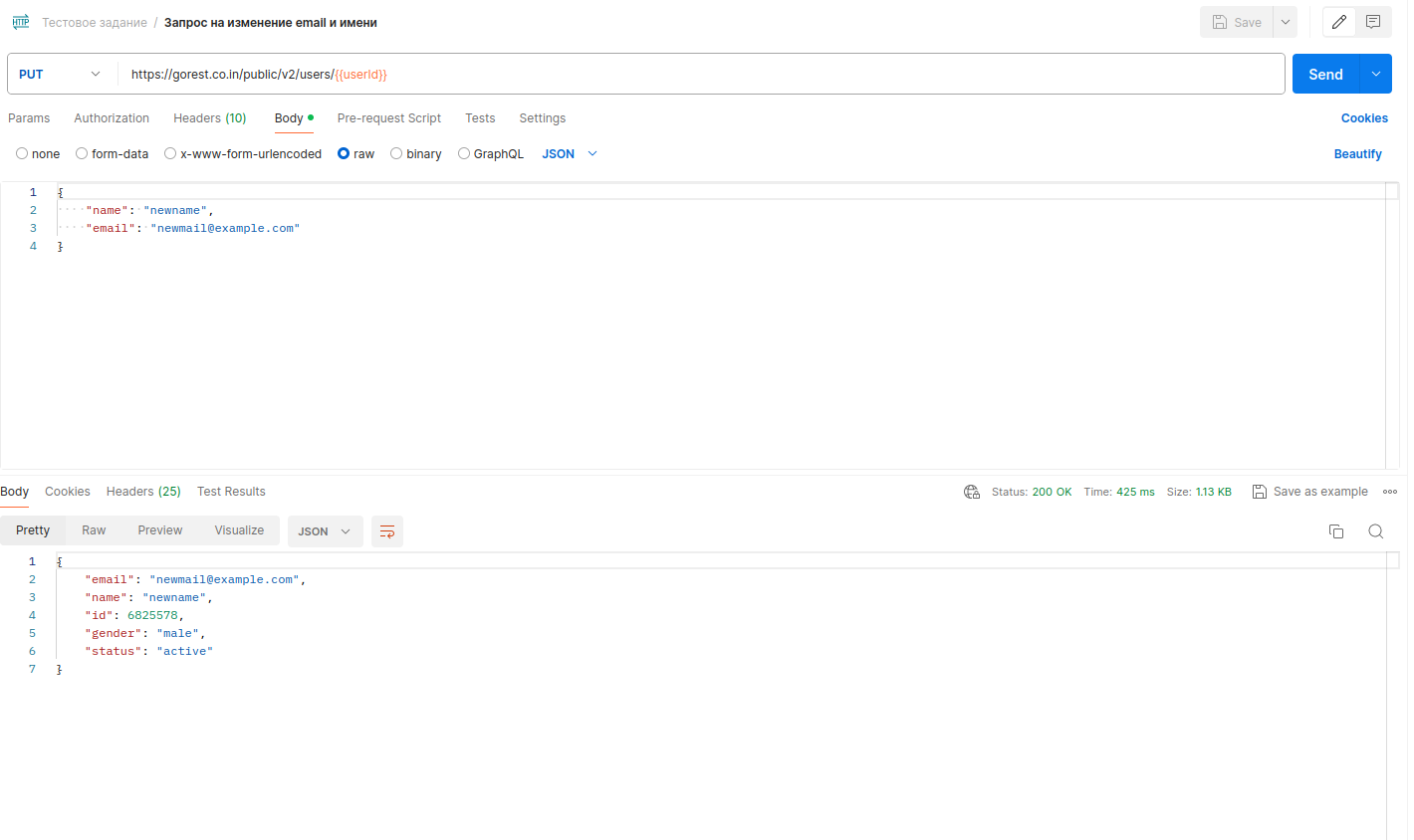
****

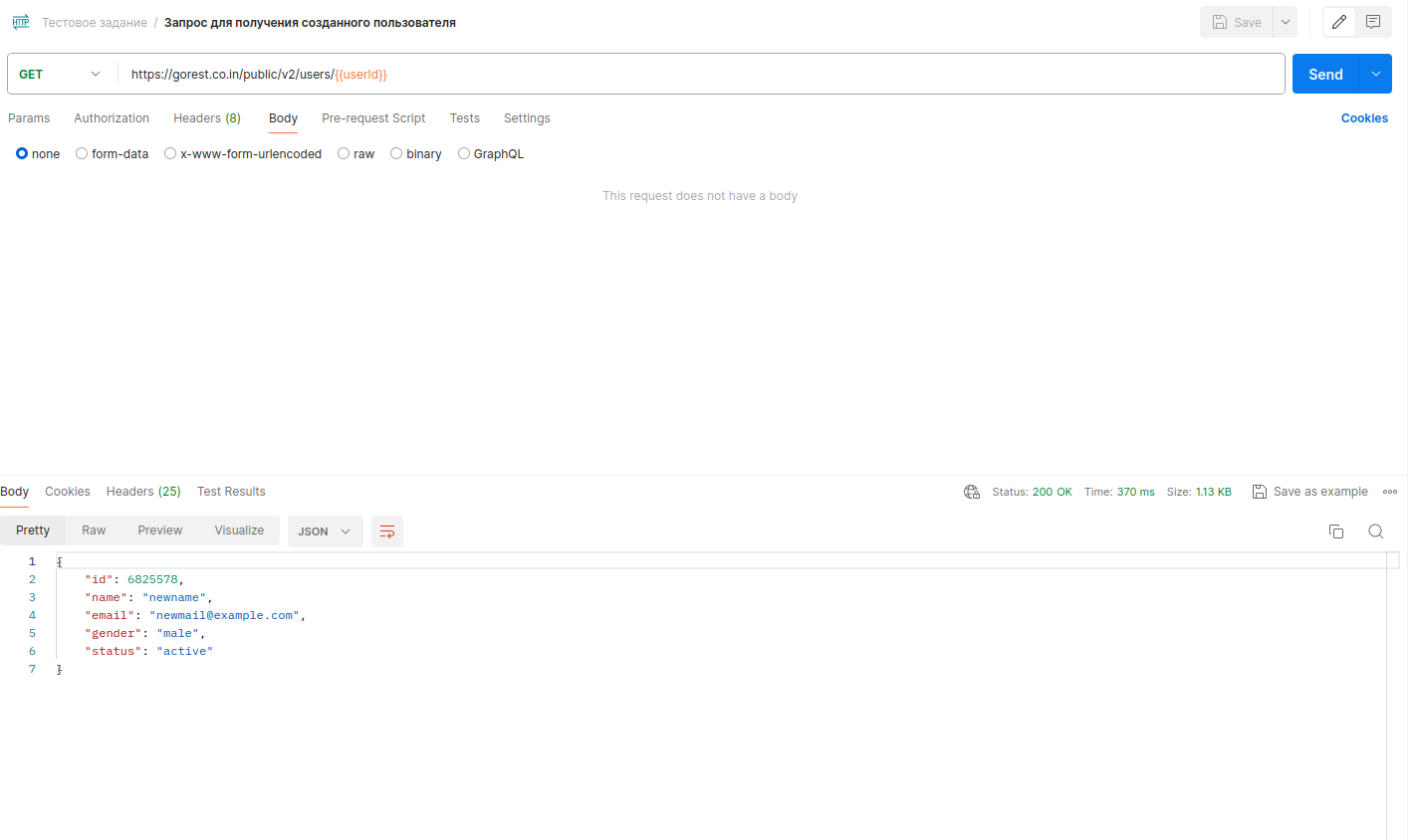
## 7. Создайте еще несколько пользователей с одинаковыми значениями “name”, но разными email. Получите список всех только что созданных пользователей. Сделайте скриншот. Какими ещё get запросами можно получить запись о пользователе?

Скриншот:

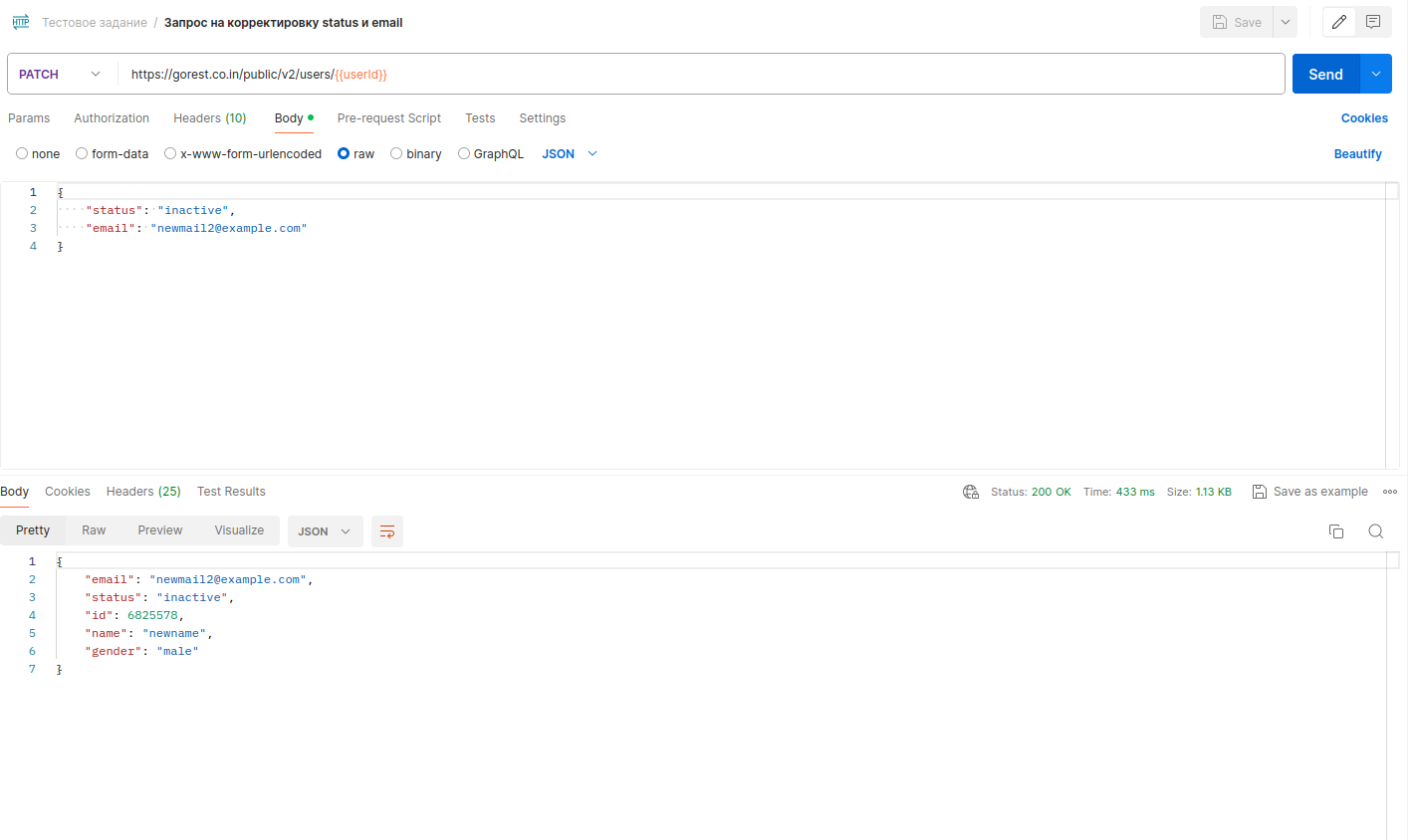
Запись о пользователе можно получить также путем передачи любого из этих параметров в тело запроса: id, name, email, gender, status. Также можно использовать совокупности параметров с целью фильтрации, например выбрать всех активных пользователей-женщин.

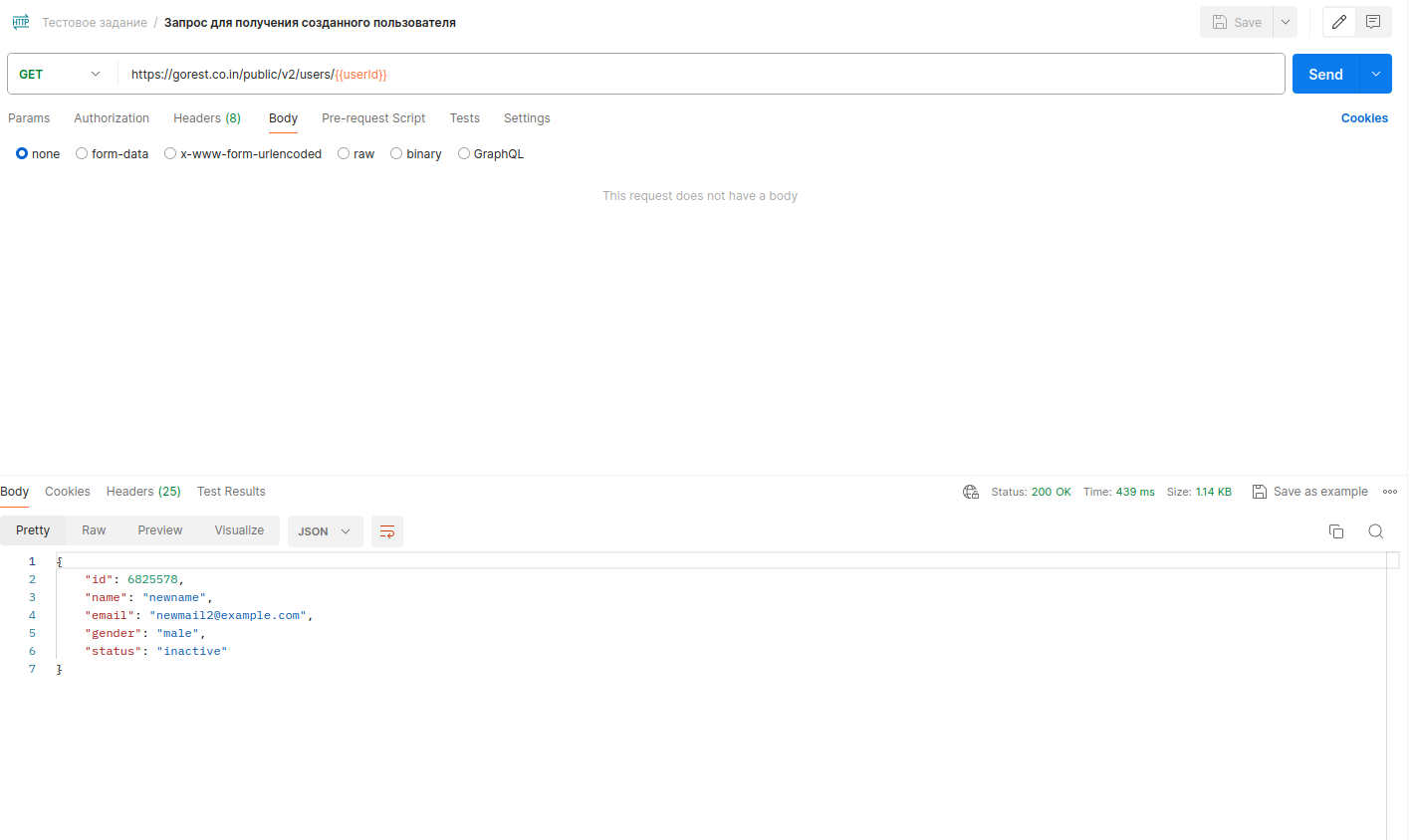
## 8. Напишите запрос с методом put на изменение email и name на другое значение. Проверьте изменение get запросом. Сделайте скриншот.

****

****

## 9. Чем отличаются запросы put и patch? Напишите запрос с методом patch для корректировки полей status и email. Сделайте скриншот. В чем отличие полученного результата, от результата запроса из п.8?

****

Запрос PUT обновляет объект целиком, основываясь на данных в теле запроса.

Запрос PATCH обновляет только те поля, которые указаны в теле запроса.

Результаты запросов отличаются значениями полей email и status. В результате запроса из п.8:

"email": "newmail@example.com",

"status": "active"

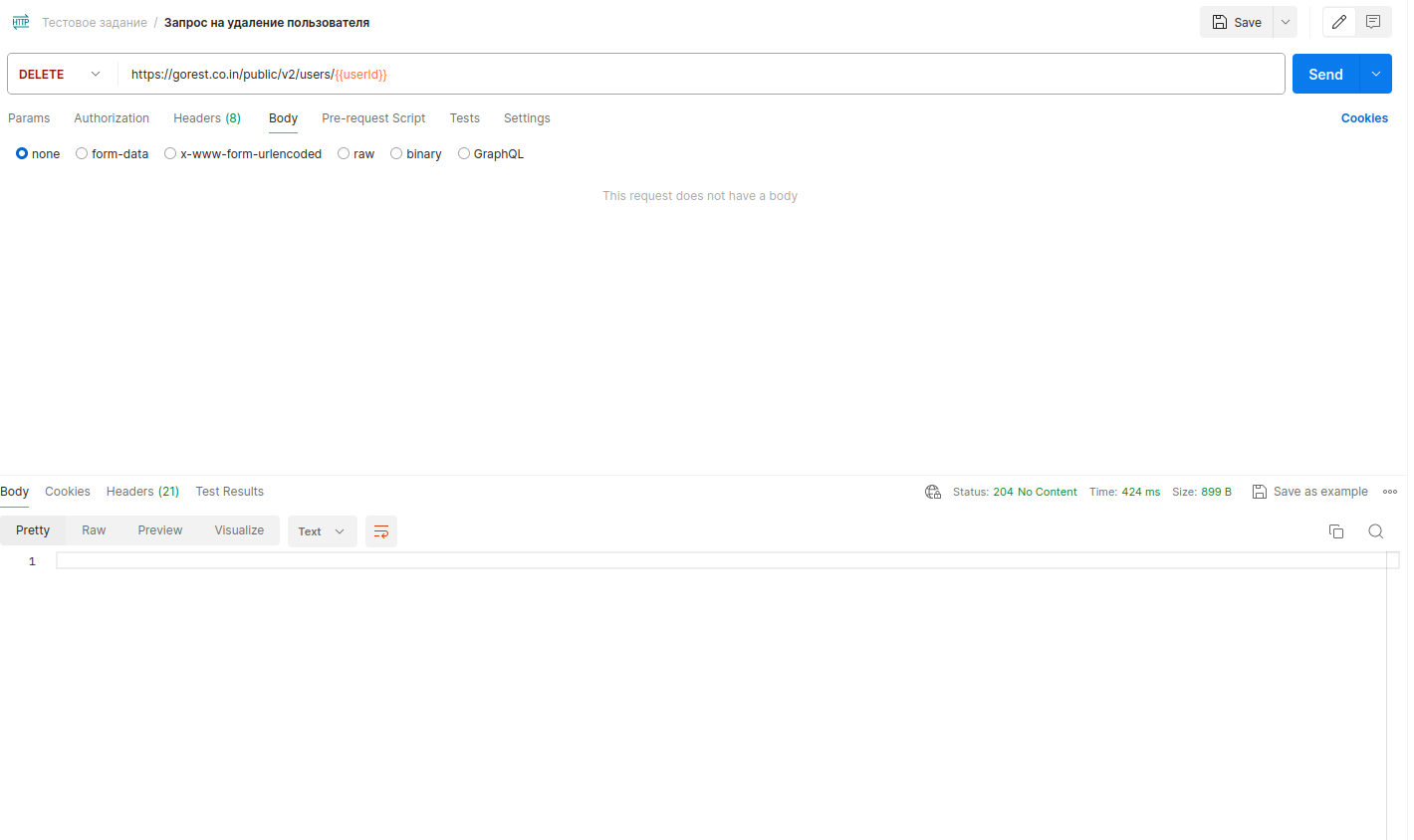
В текущем результате:

"email": "newmail2@example.com",

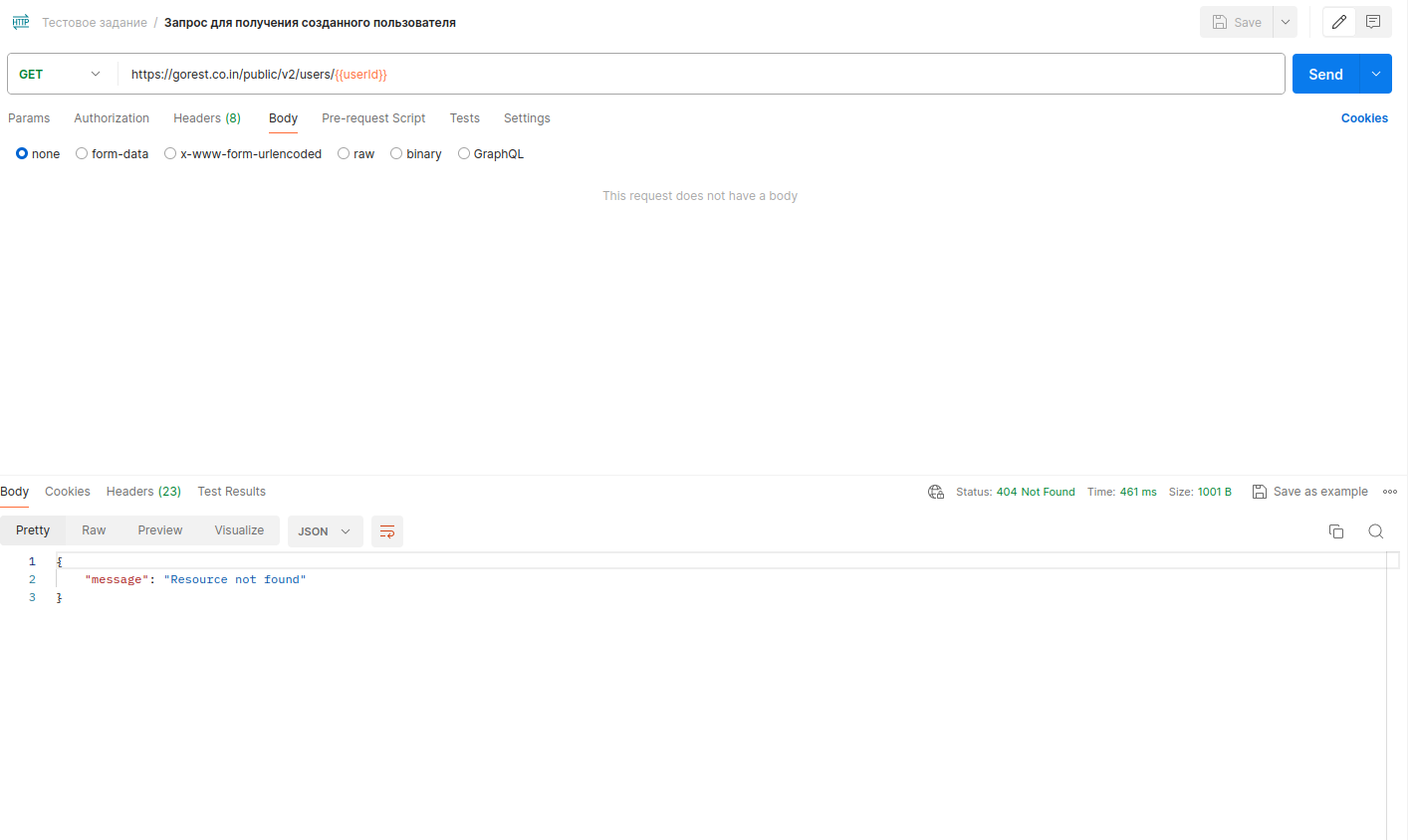
"status": "inactive",

## 10. Удалите пользователя, созданного в п.5. Что содержится в теле ответа? Сделайте скриншот. Выполните get запрос удаленного пользователя. Какой ответ пришел и почему? Сделайте скриншот.

Скриншот запроса удаления пользователя:

Тело ответа пусто.

Скриншот get запроса удаленного пользователя:

Ответ пришел следующий:

{

"message": "Resource not found"

}

Это означает, что объекта с заданным идентификатором не существует на сервере, так как он был удален через использование предыдущего запроса DELETE.

## 11. Удалите всех ранее созданных пользователей.

## 12. Сохраните коллекцию получившихся запросов в Postman.

# Теоретическое задание.

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

Базовый алгоритм тестирования модуля SPA приложения может быть сведен к двум этапам:

1. Составление тейст-кейсов, учитывающих разнообразные варианты взаимодействия с модулем.

2. Прохождение тейст-кейсов и фиксация обнаруженных ошибок в баг-репортах.

## 2. Что должен включать репорт найденной ошибки, какую структуру должен иметь?

Как правило, в репорт заносятся следующие данные:

1. Идентификатор репорта

2. Описание ошибки (ожидаемый / фактический результат), с прикрепленными скриншотами / видео воспроизведения ошибки при возможности

3. Устройство (среда, браузер), на котором была обнаружена ошибка

4. Дата обнаружения ошибки

5. Серьезность ошибки – насколько критично влияние ошибки на работоспособность системы

6. Приоритет – насколько срочно требуется исправление ошибки

Помимо этого репорт ошибки должен содержать в себе описание шагов, следуя которым пользователь может воспроизвести ошибку.

Репорт может быть представлен в виде текстового документа, электронной таблицы или быть зафиксирован в системе отслеживания ошибок.

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

Пример.

**ID Репорта: Report01**

|  |  |
| --- | --- |
| **Описание** | Не работает электрический чайник |
| **Ожидаемый результат** | После нажатия кнопки включения индикатор работы чайника должен загореться синим цветом |
| **Фактический результат** | После нажатия кнопки включения ничего не происходит |
| **Дата обнаружения** | 06.04.2024 |
| **Серьезность** | Критический |
| **Приоритет** | Высокий |
| **Статус** | Обнаружен |
| **Условие воспроизведения** | Чайник должен быть подключен к розетке модели 220V Type C |

**Шаг 1**

|  |  |
| --- | --- |
| **Действие** | Нажать кнопку включения чайника |
| **Ожидаемый результат** | Индикатор работы чайника не горит |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

## 4. Составить баг репорт страниц и разделов, включая корневой по иерархии сайта https://site.domain.ru

**ID Репорта: SD0001**

|  |  |
| --- | --- |
| **Описание** | Форма образной связи не принимает электронный адрес с доменом на кириллице за правильный e-mail адрес |
| **Ожидаемый результат** | После нажатия кнопки «Отправить сообщение» при ввведеном электронном адресе с доменом на кириллице появляется сообщение об успешной отправке обращения |
| **Фактический результат** | После нажатия кнопки «Отправить сообщение» при ввведеном электронном адресе с доменом на кириллице присутствует сообщение об ошибке «Ваш адрес e-mail не является правильным E-Mail адресом» |
| **Устройство обнаружения** | PC / Ubuntu 22.04.4 LTS / Firefox 123.0 |
| **Дата обнаружения** | 07.04.2024 |
| **Серьезность** | Незначительный |
| **Приоритет** | Низкий |
| **Статус** | Обнаружен |

**Шаг 1**

|  |  |
| --- | --- |
| **Действие** | Открыть URL в браузере |
| **Вводимое значение** | https://site.domain.ru/contact-us |
| **Ожидаемый результат** | Появилась форма для обратной связи |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

**Шаг 2**

|  |  |
| --- | --- |
| **Действие** | Ввести указанные данные |
| **Вводимое значение** | Имя: имя  Ваш адрес e-mail: help@мсз.рф  Тема обращения: тема  Текст обращения: текст |
| **Ожидаемый результат** | Данные успешно введены |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

**Шаг 3**

|  |  |
| --- | --- |
| **Действие** | Ввести код проверки |
| **Вводимое значение** | <код проверки, показанный на изображении капчи> |
| **Ожидаемый результат** | Код проверки введен |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

**Шаг 4**

|  |  |
| --- | --- |
| **Действие** | Нажать кнопку «Отправить сообщение» |
| **Вводимое значение** | - |
| **Ожидаемый результат** | Вывод сообщения об ошибке «Ваш адрес e-mail не является правильным E-Mail адресом» |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

**ID Репорта: SD0002**

|  |  |
| --- | --- |
| **Описание** | Переход к разделу «Справочная информация» осуществляется по устаревшей ссылке |
| **Ожидаемый результат** | После нажатия гиперссылки «Справочная информация» на главной странице осуществляется переход по ссылке https://site.domain.ru/info |
| **Фактический результат** | После нажатия гиперссылки «Справочная информация» на главной странице осуществляется переход по ссылке https://site.domain.ru/news/3 |
| **Устройство обнаружения** | PC / Ubuntu 22.04.4 LTS / Firefox 123.0 |
| **Дата обнаружения** | 07.04.2024 |
| **Серьезность** | Существенный |
| **Приоритет** | Средний |
| **Статус** | Обнаружен |

**Шаг 1**

|  |  |
| --- | --- |
| **Действие** | Открыть URL в браузере |
| **Вводимое значение** | https://site.domain.ru/ |
| **Ожидаемый результат** | Открыта главная страница сайта |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

**Шаг 2**

|  |  |
| --- | --- |
| **Действие** | Нажать на гиперссылку «Справочная информация», которая расположена внизу главной страницы |
| **Вводимое значение** | - |
| **Ожидаемый результат** | Открыта страница <https://site.domain.ru/news/3> |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

**ID Репорта: SD0003**

|  |  |
| --- | --- |
| **Описание** | На указанных страницах не работает должным образом переход на главную страницу сайта по ссылке:  https://site.domain.ru/news/1  https://site.domain.ru/news/3  <https://site.domain.ru/sign-up> |
| **Ожидаемый результат** | После нажатия гиперссылки «Главная» осуществляется переход на главную страницу сайта [https://site.domain.ru/](https://site.domain.ru/info) |
| **Фактический результат** | После нажатия гиперссылки «Главная» осуществляется переход на текущую страницу (пустой атрибут href) |
| **Устройство обнаружения** | PC / Ubuntu 22.04.4 LTS / Firefox 123.0 |
| **Дата обнаружения** | 07.04.2024 |
| **Серьезность** | Незначительный |
| **Приоритет** | Низкий |
| **Статус** | Обнаружен |

**Шаг 1**

|  |  |
| --- | --- |
| **Действие** | Открыть страницы в браузере |
| **Вводимое значение** | <https://site.domain.ru/news/1>  <https://site.domain.ru/news/3>  <https://site.domain.ru/sign-up> |
| **Ожидаемый результат** | Страницы открыты |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

**Шаг 2**

|  |  |
| --- | --- |
| **Действие** | Для каждой из страниц нажать на ссылку «Главная», расположенной в верхнем меню |
| **Вводимое значение** | - |
| **Ожидаемый результат** | Каждая из открытых страниц открывается повторно |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

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

Допустим, что форма регистрации включает в себя поля email, пароль, а также дополнительные поля, для указания имени, номера телефона, даты рождения, и. т. п. В таком случае алгоритм тестирования системы регистрации может быть описан следующим образом:

1. Сформировать тестовые наборы данных для положительных и негативных тест-кейсов.

2. Создать положительные тест-кейсы, такие как:

1. Ввод только необходимых данных (email, пароль / номер телефона);
2. Ввод всех данных.

3. Создать негативные тест-кейсы, такие как:

1. Отсутствие ввода необходимых данных (одно / несколько / все поля);
2. Неверный ввод какого-либо поля (email, пароль / номер телефона).

4. Протестировать систему на основе сделанных тест-кейсов и опубликовать отчеты об ошибках, если таковые будут обнаружены.

## 6. 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.)

Для тестирования создания сущности я бы использовал следующие запросы к API:

1. GET запрос на получение данных о сущностях в базе, чтобы ознакомиться с исходным состоянием базы данных;

2. POST / PUT запрос на создание новой сущности с телом запроса, в котором описаны нужные параметры;

3. GET запрос на получение данных о новой сущности, чтобы удостовериться, что сущность действительно была создана;

4. GET запрос на получение данных о сущностях в базе после создания новой сущности, чтобы посмотреть, как изменилась база данных после добавления сущности.

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

**ID Репорта: Bug01**

|  |  |
| --- | --- |
| **Описание** | Форма не проверяет наличие заполненного поля username |
| **Ожидаемый результат** | После нажатия кнопки «Отправить» выводится ошибка «Поле username не может быть пустым» |
| **Фактический результат** | Полсе нажатия кнопки «Отправить» переход на страницу с сообщением «Ваши данные были успешно отправлены» |
| **Устройство обнаружения** | PC / Ubuntu 22.04.4 LTS / Firefox 123.0 |
| **Дата обнаружения** | 07.04.2024 |
| **Серьезность** | Критический |
| **Приоритет** | Высокий |
| **Статус** | Обнаружен |

**Шаг 1**

|  |  |
| --- | --- |
| **Действие** | Открыть URL в браузере |
| **Вводимое значение** | https://test.com/form |
| **Ожидаемый результат** | Открыта форма для ввода данных |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

**Шаг 2**

|  |  |
| --- | --- |
| **Действие** | Нажать на кнопку «Отправить» |
| **Вводимое значение** | - |
| **Ожидаемый результат** | Переход на страницу с сообщением «Ваши данные были успешно отправлены» |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

## 8. Можно ли передавать в json бинарные данные? Почему?

Можно, если эти данные представлены в текстовом формате, таком как Base64. Тем не менее, этого делать не рекомендуется по следующим причинам:

1. Скорость обработки JSON резко падает ввиду увеличения объема файла JSON за счет внесения массивных Base64 данных.

2. Удобночитаемость JSON конфига сильно снижается. Формат JSON теряет такие преимущества, как компактность и минималистичность данных.

В качестве альтернативы в JSON намного предпочтительнее использовать URI ссылки на источники бинарных данных.

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

Это могут быть следующие наборы данных:

1. Позитивные:

1. (Текущая дата - 18 лет)
2. (Текущая дата – 18 лет 0 месяцев 1 день)
3. (Текущая дата – 18 лет 1 месяц 0 дней)

2. Негативные:

1. (Текущая дата – 17 лет 11 месяцев 30 дней)
2. (Текущая дата – 17 лет 11 месяцев 0 дней)

## 10. Как протестировать скачивание файла при недостаточном количестве места на диске.

Для этого необходимо создать виртуальный жесткий диск с ограниченным объемом, который меньше, чем размер файла. Затем указать путь сохранения для скачиваемого файла на созданный виртуальный жесткий диск и наблюдать за дальнейшим поведением.

## 11. Как протестировать загрузку файла при недостаточном количестве места на сервере.

Для этого нужно ограничить объем места на сервере. Это может быть достигнуто установления дисковых квот, что позволит ограничить дисковое пространство для заданных пользователей, либо путем создания хранилищ данных с огранниченным доступным местом. После ограничения количества места на сервере необходимо загрузить файл и наблюдать за дальнейшим поведением системы.

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

* 1. **ID Кейса: auth01**

|  |  |
| --- | --- |
| **Описание** | Авторизация с некорректным паролем |
| **Тип** | Негативный |
| **Приоритет** | Высокий |
| **Условия** | Наличие аккуаунта в системе со следующими данными:  e-mail: test@example.com  password: correctpassword |
| **Статус** |  |

**Шаг 1**

|  |  |
| --- | --- |
| **Действие** | Открыть URL в браузере |
| **Вводимое значение** | https://test.com/auth |
| **Ожидаемый результат** | Появилась форма авторизации |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

**Шаг 2**

|  |  |
| --- | --- |
| **Действие** | Ввести указанные данные |
| **Вводимое значение** | e-mail: test@example.com  password: incorrectpassword |
| **Ожидаемый результат** | Данные введены |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

**Шаг 3**

|  |  |
| --- | --- |
| **Действие** | Нажать на кнопку «submit» |
| **Вводимое значение** | - |
| **Ожидаемый результат** | Появление ошибки «Неверный пароль» |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

* 1. **ID Кейса: auth02**

|  |  |
| --- | --- |
| **Описание** | Авторизация с некорректным почтовым адресом |
| **Тип** | Негативный |
| **Приоритет** | Высокий |
| **Условия** | Отсутствие аккуаунта в системе со следующими данными:  e-mail: [nonexistent@example.com](mailto:nonexistent@example.com) |
| **Статус** |  |

**Шаг 1**

|  |  |
| --- | --- |
| **Действие** | Открыть URL в браузере |
| **Вводимое значение** | https://test.com/auth |
| **Ожидаемый результат** | Появилась форма авторизации |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

**Шаг 2**

|  |  |
| --- | --- |
| **Действие** | Ввести указанные данные |
| **Вводимое значение** | e-mail: [nonexistent@example.com](mailto:nonexistent@example.com)  password: correctpassword |
| **Ожидаемый результат** | Данные введены |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

**Шаг 3**

|  |  |
| --- | --- |
| **Действие** | Нажать на кнопку «submit» |
| **Вводимое значение** | - |
| **Ожидаемый результат** | Появление ошибки «Аккаунта с данным email не существует» |
| **Фактический результат** |  |
| **Устройство проверки** |  |
| **Дата проверки** |  |
| **Статус проверки** |  |
| **Комментарий** |  |

## 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?

Я буду тестировать функционал калькулятора следующим образом:

1. Сформирую следующие наборы значений для каждого параметра, основываясь на технике анализа граничных значений:

1. Для позитивных тестов: 0.1, 5, 99.9
2. Для негативных тестов: 0, 0.09, 100, -5

2. Воспользуюсь методом попарного анализа и сформирую следующую таблицу для формирования позитивных тестов.

|  |  |  |
| --- | --- | --- |
| **числитель** | **делитель** | **результат** |
| 0,1 | 0,1 | 1 |
| 0,1 | 5 | 0,02 |
| 0,1 | 99,9 | 0,001001001001001 |
| 5 | 5 | 1 |
| 5 | 99,9 | 0,05005005005005 |
| 5 | 0,1 | 50 |
| 99,9 | 99,9 | 1 |
| 99,9 | 0,1 | 999 |
| 99,9 | 5 | 0,001001001001001 |

3. Затем сформирую следующую таблицу для негативных тестов, где каждое негативное значение встречается один раз на месте числителя и делителя совместно с позитивным значением:

|  |  |  |
| --- | --- | --- |
| **числитель** | **делитель** | **результат** |
| 0 | 5 | ошибка |
| 5 | 0 | ошибка |
| 0.09 | 5 | ошибка |
| 5 | 0.09 | ошибка |
| 100 | 5 | ошибка |
| 5 | 100 | ошибка |
| -5 | 5 | ошибка |
| 5 | -5 | ошибка |
| пустое значение | 5 | ошибка |
| 5 | пустое значение | ошибка |

4. Так как в результате получилось довольно значительное количество тестовых данных, то я бы попытался автоматизировать проведение тестирования с помощью инструмента Selenium WebDriver, путем написания соответствующих Python скриптов. Таким образом, на этом шаге завершается этап функционального тестирования приложения.

5. Так как помимо логики, приложение содержит графический интерфейс, то следует провести и UI-тестирования. Для этого я бы отобрал из тестовых наборов, используемых на предыдущих шагах, группы значений, которые дают результат, содержащий большое количество чисел. Основываясь на отобранных группах, я бы проверил отображения результатов вычислений калькулятора в различных браузерах, включая и мобильные варианты (при помощи DevTools).