**Итоговая аттестация**

**Отчет о тестировании функционала**

“Запрос ленты с постами и авторизация”

(выполняется в файле)

## **Информация о проекте.**

Необходимо протестировать функционал – Авторизация тестового пользователя, Лента своих постов и Лента чужих постов.

Описание функционала доступно по ссылке: [Итоговое задание\_Продуктовые требования (Ручка запроса опубликованных постов)](https://docs.google.com/document/d/19IDnFaUDtxTRbvBo-EN5maXb2bkNlmYEBmtJb8A58aE/edit?usp=sharing)

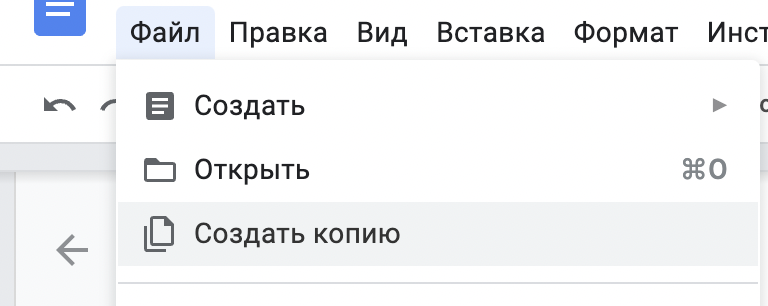
**⚠️На данный момент у вас уже должен быть доступ на тестовый стенд, после прохождения курсов Тестирование АПИ и Тестирование веб-приложений.**

**Если у вас все же нет его,** [**https://test-stand.gb.ru/login**](https://test-stand.gb.ru/login) **необходимо обратиться к преподавателю / наставнику или ревьюверу курса, тебе выдадут учетную запись, с помощью которой сможешь создавать своих тестовых пользователей.**

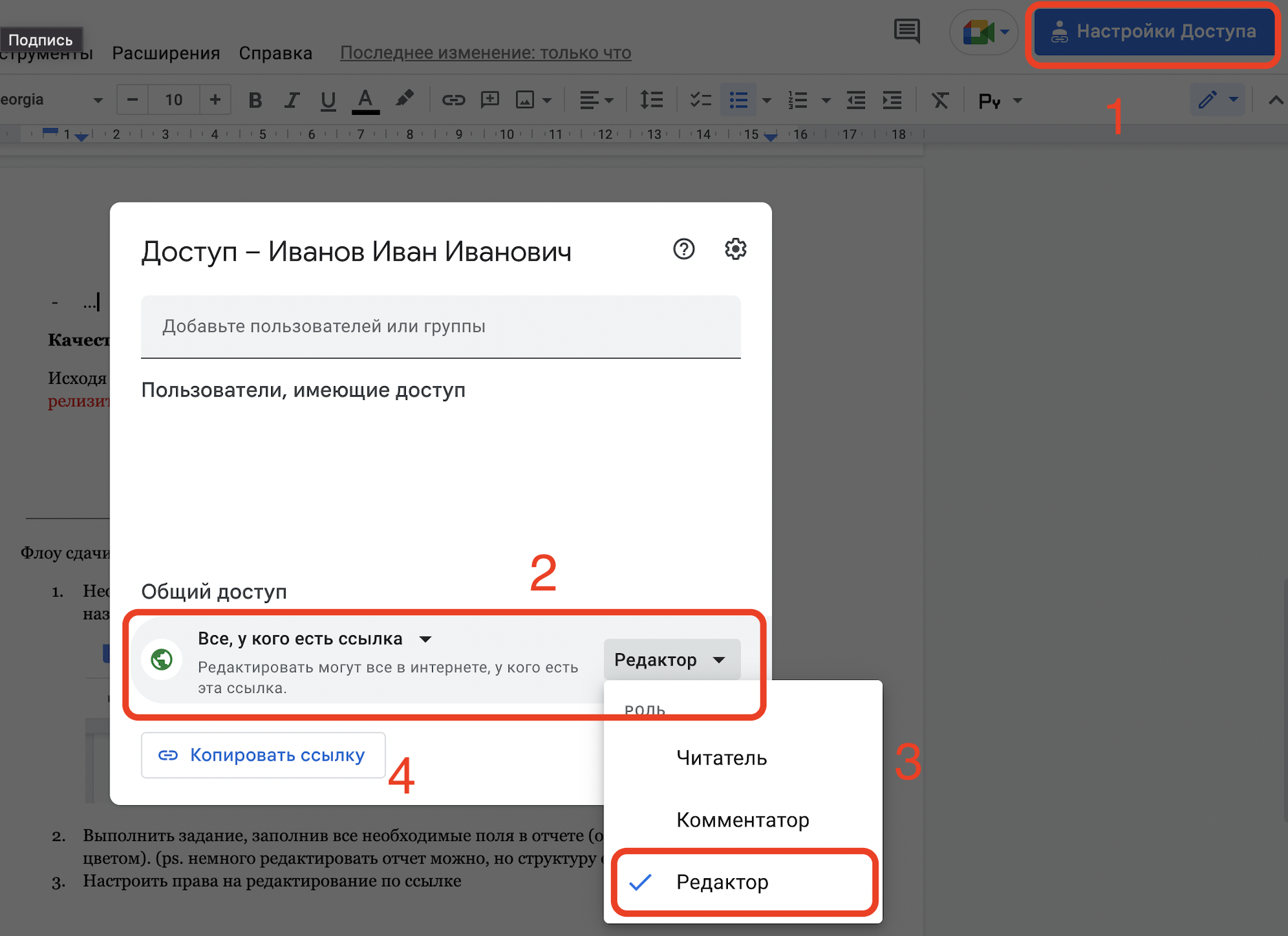
**В качестве ознакомления можешь использовать логин: autotest пароль: 4956318935. Но необходимо получить свою персональную учетную запись, чтобы настроить свое собственное окружение!**

## **Флоу сдачи итогового проекта**

1. Необходимо создать копию этого отчета себе на гугл диск, выбрав “Создать копию” и назовите файл: ФИО



1. Выполнить задание, заполнив все необходимые поля в отчете (обозначены красным цветом). (ps. немного редактировать отчет можно, но структуру оставить как есть).
2. Настроить **права на редактирование** по ссылке.



1. После того, как клонировали ссылку, убедитесь, что по ней открывается ваша работа через режим инкогнито, для этого, например в Chrome откройте “Новое окно в режиме инкогнито”.
2. Если открывается ссылка, то значит и у преподавателя откроется. Можете смело прикреплять на платформе ссылку на отчет этот.
3. Дополнительно сохранить документ как pdf и прикрепить в окошко сдачи.

## **Задание**

**Команда тестировщиков.**

В таблице указан список специалистов, задействованных в тестировании проекта, с указанием закрепленного за каждым участником тестируемого функционала.

| № | ФИО | Должность | Функционал |
| --- | --- | --- | --- |
| 1 | Тимофеев Дмитрий | Тестировщик | Выдача постов |

## **Описание процесса тестирования.**

В качестве тестовой документации были составлены чек-листы. Выбор был сделан в пользу них, так как в данном случае для тестирования нет необходимости продумывать сложные сценарии и описывать ожидаемый результат в зависимости от проделанных действий - все проверки достаточно просты для понимания и реализации; также составление чек-листа и проверка по нему помогут ускорить процесс тестирования по сравнению с вариантом подготовки тест-кейсов. Тестовая документация находится по адресу:

[Тестирование: итоговая аттестация - WEB (Тимофеев Д. А.)](https://docs.google.com/spreadsheets/d/1E4nQOoHSdFN0YuSaWvZtz4l-qHjXVYyKx252Vn2DJn8/edit?gid=0#gid=0)

[Тестирование: итоговая аттестация - API (Тимофеев Д. А.)](https://docs.google.com/spreadsheets/d/1G3WBtq4C_Lly2M8rlLHTL20WCZxpJBV5S4uRj42ruNA/edit?gid=0#gid=0)

## **Этап 1. Начнем с тестирования WEB интерфейса.**

### Задание 1. Тестирование

Необходимо провести тестирование по составленным чек-листам. Результат прогона тест-рана изложен ниже.

Страница авторизации:

*passed - 8*

*failed - 8*

*skipped - 2*

В результате анализа требований к странице авторизации и тестирования возник ряд вопросов.

**1)** В требованиях есть противоречие, связанное с возвратом кода ошибки 404. В случае отсутствия интернет-соединения система попытается отправить запрос, *но ответ получен не будет, следовательно и кода ошибки получено не будет*. Судя по требованиям, эту ситуацию должен обработать фронт, вызвав на странице авторизации ошибку ниже поля Пароль.

В то же время если текст об отсутствии интернет соединения по задумке заказчика или разработчика должен отображаться в случае возврата от сервера ответа с кодом 404, то это будет *вводить в заблуждение пользователя*, т.к. по факту интернет есть, но текст ошибки противоречит этому условию.

**2)** Также возникает вопрос об используемых символах в поле пароль. Какие символы могут приниматься полем на странице авторизации? Если судить по текущей реализации платформы, то логично предположить, что поле должно принимать цифры и латинские символы (т.к. система при создании пользователя генерирует пароль, состоящий из латинских символов и цифр).

Но что должно происходить при использовании, например, спец.символов?

Это уточнение необходимо, т.к. система может некорректно обрабатывать некоторые символы или же вовсе их не принимать

**3)** В самом задании изображение полей Логин и Пароль, а также кнопки LOGIN отличается от макета (отражено в комментариях к проверкам по отображению)

**4)** Невозможно понять в текущей реализации - работает ли ФЛК полей на фронте в принципе, - т.к. сообщение для полей в случае ошибки не отображаются

**5)** Необходимо более детальная информация, чтобы понять, как можно вызвать ошибки отличные от ошибок с кодом 401 и 404

Лента своих постов:

*passed - 23*

*failed - 10*

*skipped - 0*

*blocked - 16*

Лента чужих постов:

*passed - 21*

*failed - 9*

*skipped - 0*

*blocked - 21*

Макеты ленты постов в фигме и миро отличаются от макета, который показан в требованиях. При этом не сказано явно, что какой-то из функционал устарел. Поэтому в чек-листе составлены проверки с целью максимально покрыть требования. В результате заведены баг, в том числе и блокеры. При *необходимости* **продукт-менеджер** или **скрам-мастер** могут перевести эти баги в неактуальные, соответственно и в чек-листе можно будет сделать корректировки.

Также стоит отметить, что в требованиях есть противоречие. На фронте указано, что сортировка по умолчанию идет от новых к старым постам, т.е. DESC. Однако в требованиях к API сортировка по дефолту ASC, т.е. от старых к новым. Возможно на фронте это учитывается и происходит отображение постов в обратном порядке, но если так, то это по сути двойная работа. В любом случае требуется уточнение по документации.

### Задание 2. Статистика по дефектам.

Заведенные баг-репорты можно найти по следующему адресу:[Тестирование: промежуточная аттестация (Тимофеев Д. А.)](https://docs.google.com/spreadsheets/d/1HC210rynQG7Aghd4evb7PXKx3XT9Yi9m-9kufdUk0cQ/edit?gid=152632312#gid=152632312)

Сводка по багам на странице авторизации :

* 7 – имеют статус *major*.
* 2 – имеют статус critical.

**Стоит обратить внимание на баг №3 для страницы авторизации. Он показывает, что, возможно, ФЛК не настроен в системе вообще**. Ведь можно, используя админскую учетку, создать пользователя с недопустимыми символами в логине, например, с нижним подчеркиванием (это тоже баг):

Login: user\_polar\_wolf\_590033

PW: 2e593e01ee

А затем авторизоваться под этим юзером. Т.е. система проигнорировала ФЛК, либо его нет.

*В свою очередь в требованиях для создания аккаунтов через админскую учетку ограничение для логина обозначено лишь по длине и уникальности*. https://gbcdn.mrgcdn.ru/uploads/asset/5817788/attachment/f384f31d1bd62b3fc75a2aa50540a71d.pdf

Сводка по багам на странице своих постов :

* 3 - minor
* 3 - major
* 3 - blocker

Сводка по багам на странице чужих постов :

* 1 - minor
* 3 - major
* 4 - blocker

## **Этап 2. Ручное тестирование API**

### Задание 1. Понимание задание

Вам необходимо еще раз ознакомиться с ТЗ на раздел “Запрос ленты с постами”: [Итоговое задание\_ Ручка запроса опубликованных постов](https://docs.google.com/document/d/19IDnFaUDtxTRbvBo-EN5maXb2bkNlmYEBmtJb8A58aE/edit?usp=sharing). После ознакомления необходимо ответить на вопросы:

1. для чего необходим данный функционал: т.к. мы работаем с клиент-серверным приложением, то у него есть бэк и фронт; соответственно есть API, обращаясь к которому приложение на стороне клиента (фронт) будет взаимодействовать с сервером (бэком). Т.е. API - это контракт, по которому происходит взаимодействие клиентской части приложения с серверной.
2. укажите URL на запрос ленты с постами своего пользователя:

https://test-stand.gb.ru/api/posts?sort=createdAt&order=[ASC|DESC]&page=[0|1|2….]

1. а какие query параметры еще передаются для запроса в пункте 2, перечислите их:

sort, order, page

1. укажите URL на запрос ленты с постами чужого пользователя:

https://test-stand.gb.ru/api/posts?owner=notMe&sort=createdAt&order=[ASC|DESC]&page=[0|1|2….]

1. а какие query параметры еще передаются в пункте 4, перечислите их:

owner, sort, order, page

1. а для того, чтобы выполнить запрос на получение постов (своих/чужих) в ленте, что необходимо сделать:

Нужно предварительно отправить запрос POST <https://test-stand.gb.ru/gateway/login>, передав в теле запроса логин и пароль пользователя (поля “username” и “password”), а из ответа нам нужно будет вытащить “token” пользователя. Этот токен мы будем передавать в заголовке “X-Auth-Token” запроса GET: https://test-stand.gb.ru/api/posts

1. верно, необходимо передать X-Auth-Token в Header, а чтобы получить X-Auth-Token, какие логин и пароль необходимо использовать:

Login: user\_polar\_wolf\_590033

PW: 2e593e01ee

### Задание 2. Создание коллекции с запросами в Postman

## Необходимо провести тестирование по составленным чек-листам. Чек-листы можно найти по ссылке:

[Тестирование: итоговая аттестация - API (Тимофеев Д. А.)](https://docs.google.com/spreadsheets/d/1G3WBtq4C_Lly2M8rlLHTL20WCZxpJBV5S4uRj42ruNA/edit?gid=1643944151#gid=1643944151)

## Коллекция с запросами в Postman доступна по следующему адресу: https://drive.google.com/file/d/11YSex20Zq5mKcbxhNAVmoAk8VkDHIX-f/view?usp=drive\_link

Как указывалось ранее, в требованиях есть противоречие. На фронте указано, что сортировка по умолчанию идет от новых к старым постам, т.е. DESC. Однако в требованиях к API сортировка по дефолту ASC, т.е. от старых к новым. Возможно на фронте это учитывается и происходит отображение постов в обратном порядке, но если так, то это по сути двойная работа и лишние сложности. В любом случае требуется уточнение по документации.

Также требуется уточнение по ТЗ, что же все-таки должно приходить в случае запроса нулевой страницы. Судя по всему, должен приходить пустой ответ, т.к. данной страницы не существует.

Также требуется уточнение по документации - сколько постов должно приходить в ответе на одной странице при запросе своих постов? На данный момент принимаем за истину такое же значение, которое указано в требованиях для одной страницы своих постов на фронте - 10. Аналогичную ситуацию имеем и для получения чужих постов.

Возможно, стоит пересмотреть документацию, т.к. на мой взгляд более подходящим типом данных для полей prevPage и nextPage является не string, а number. Однако на данный момент это отличие заведено как баг.

Небольшой комментарий к постман-коллекции. Одним из кейсов является проверка сортировок постов по дате в поле “createdAt” - от самого первого до самого последнего поста: для случая своих постов и чужих постов. Учитывая, что сейчас механизм получения всех постов через query param order=ALL не работает, были составлены тесты в коллекции со следующими алгоритмами.

* Для постов пользователя - просто последовательно получаем все страницы с постами и сохраняем все посты в список, а затем проверяем его сортировку по полю createdAt.
* Для чужих постов - учитывая их огромное количество, выбираются рандомно 50 страниц из общего количества и запрашиваются в порядке возрастания; все респонсы этих страниц сохраняются в список. Далее из списка респонсов отбираются в отдельный список посты и уже этот список проверяется на определенную сортировку. Количество страниц 50 (50 запросов) выбрано как оптимальное на мой взгляд, учитывая возможные ограничения на сервере по количеству запросов в единицу времени для клиента.

Тестирование API ленты своих постов:

*passed - 13*

*failed - 6*

*skipped - 0*

*blocked - 0*

Тестирование API ленты чужих постов:

*passed - 15*

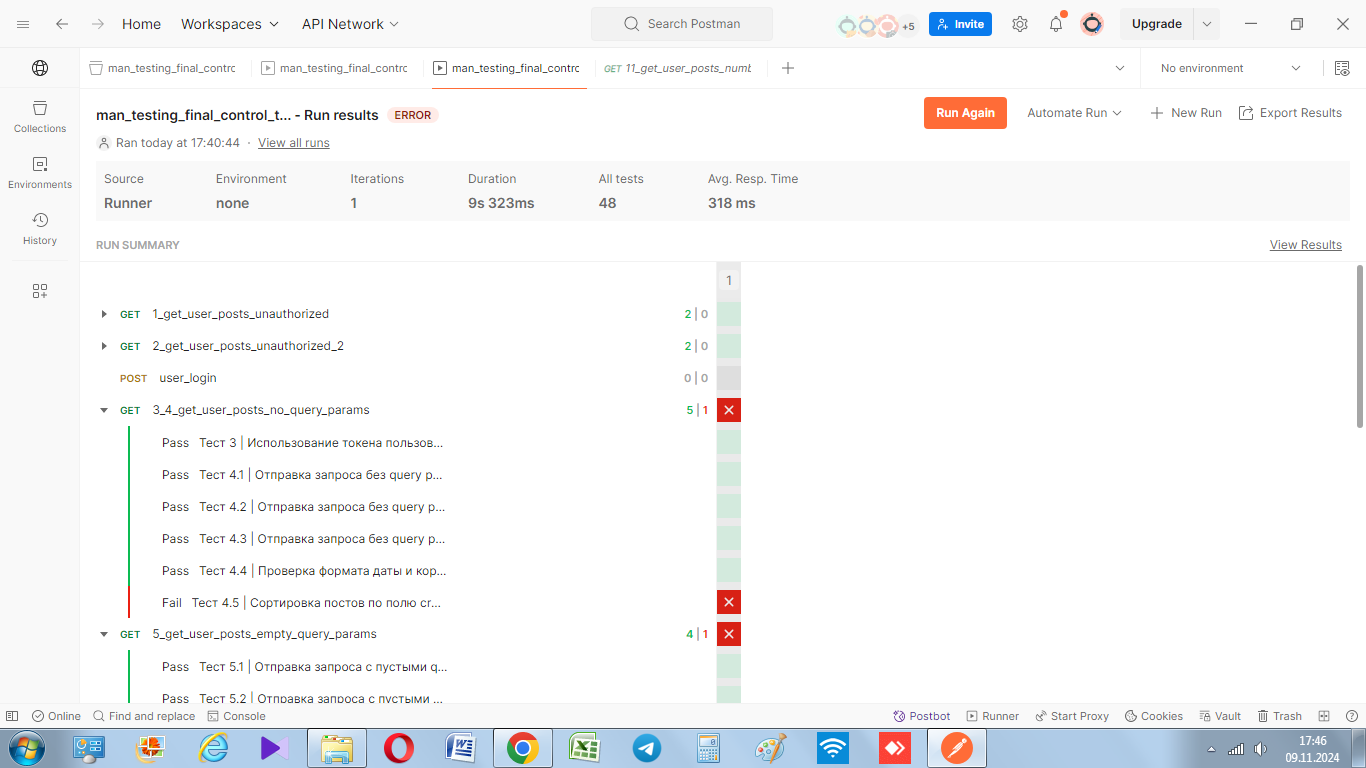
*failed - 4*

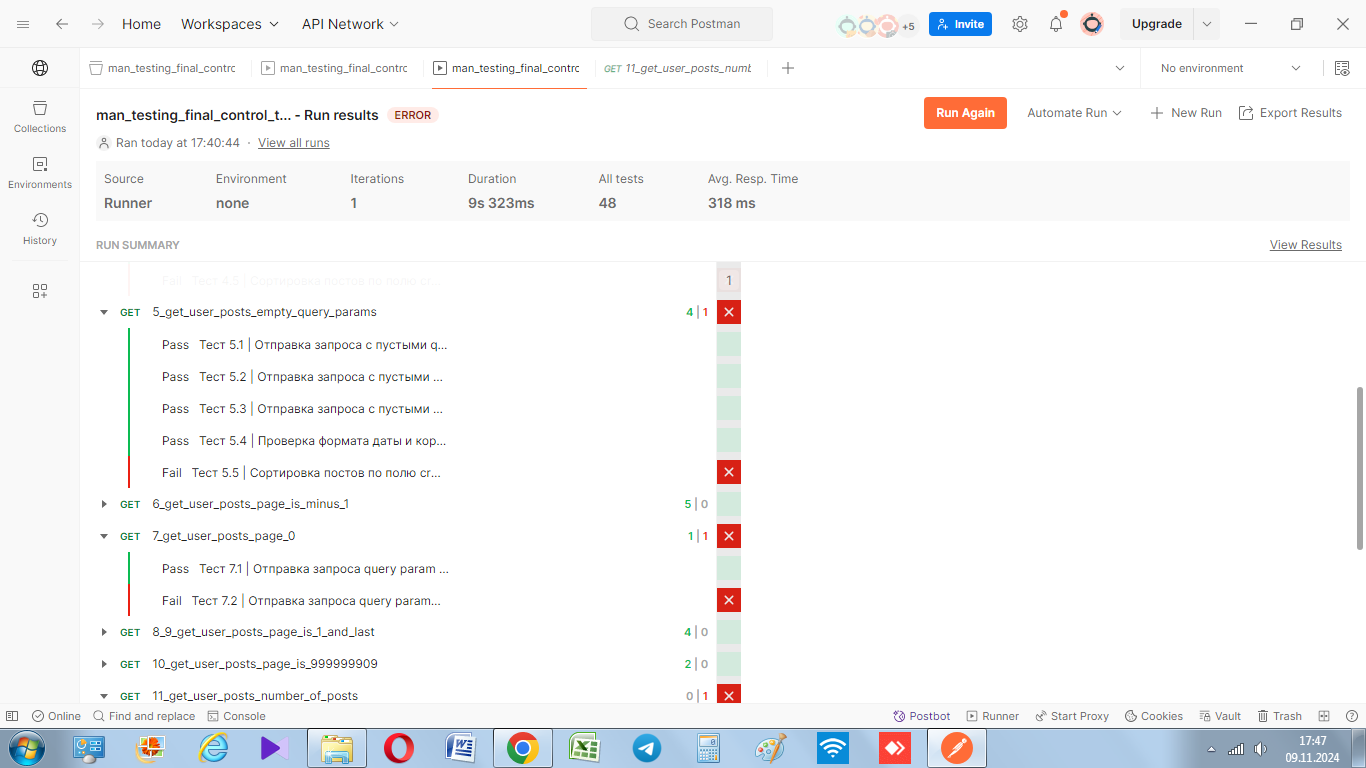
*skipped - 0*

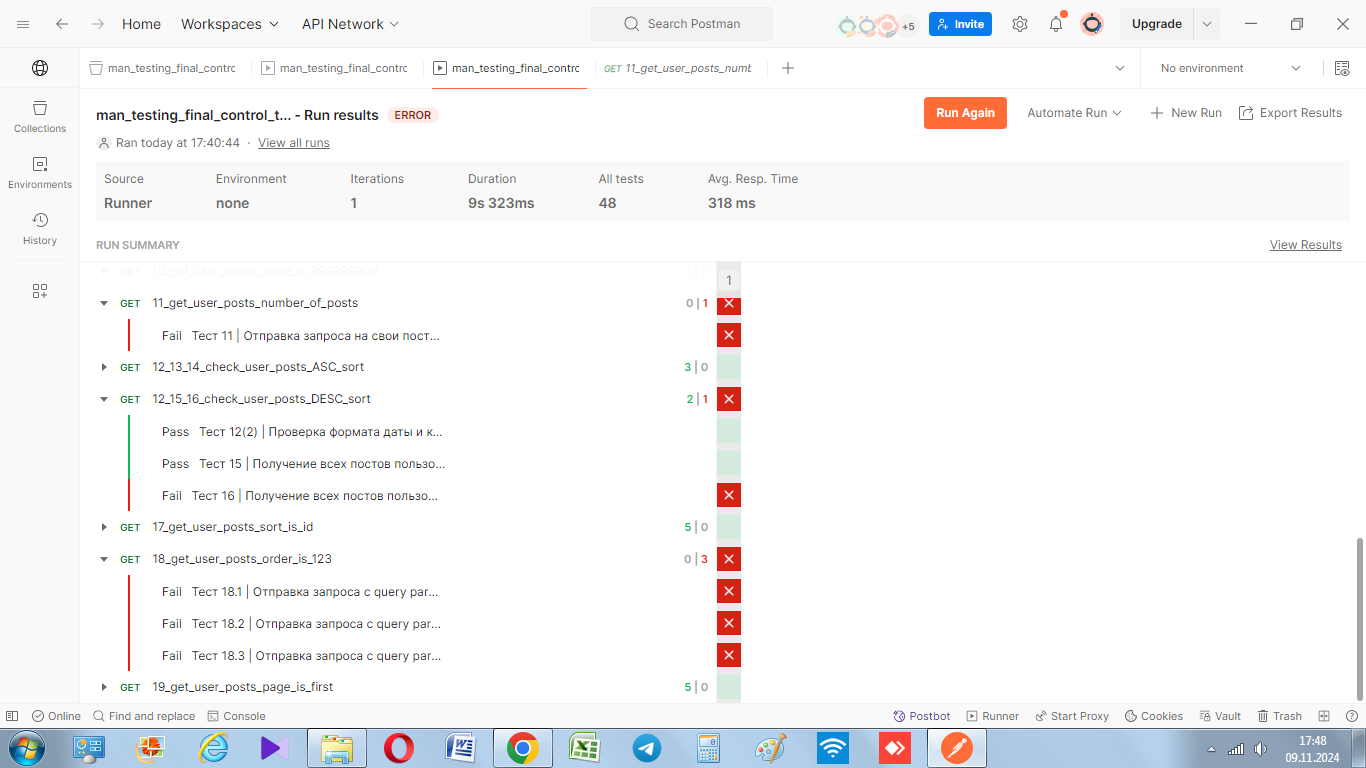
*blocked - 2*

Каждый запрос был покрыт тестами, результат прогона всей коллекции можно видеть на скриншотах ниже:

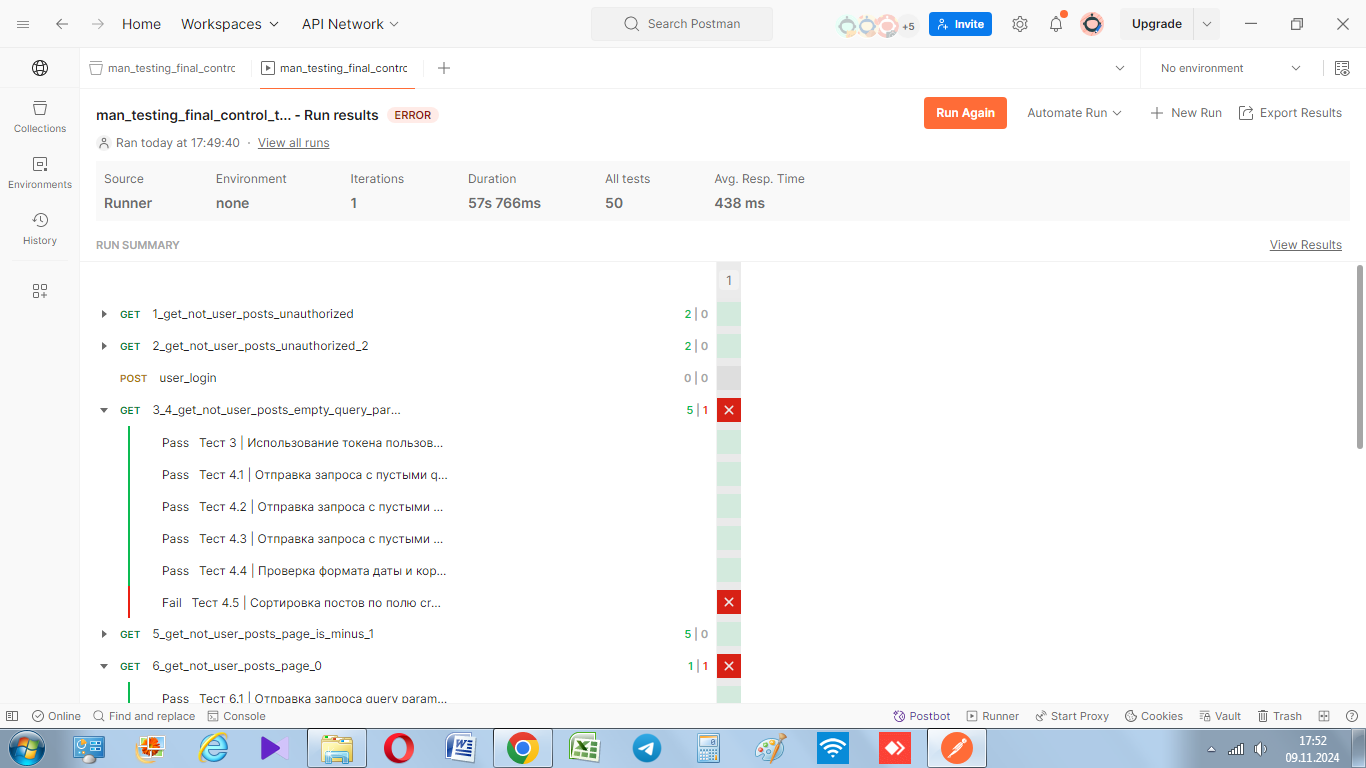
Прогон тестов API ленты своих постов:

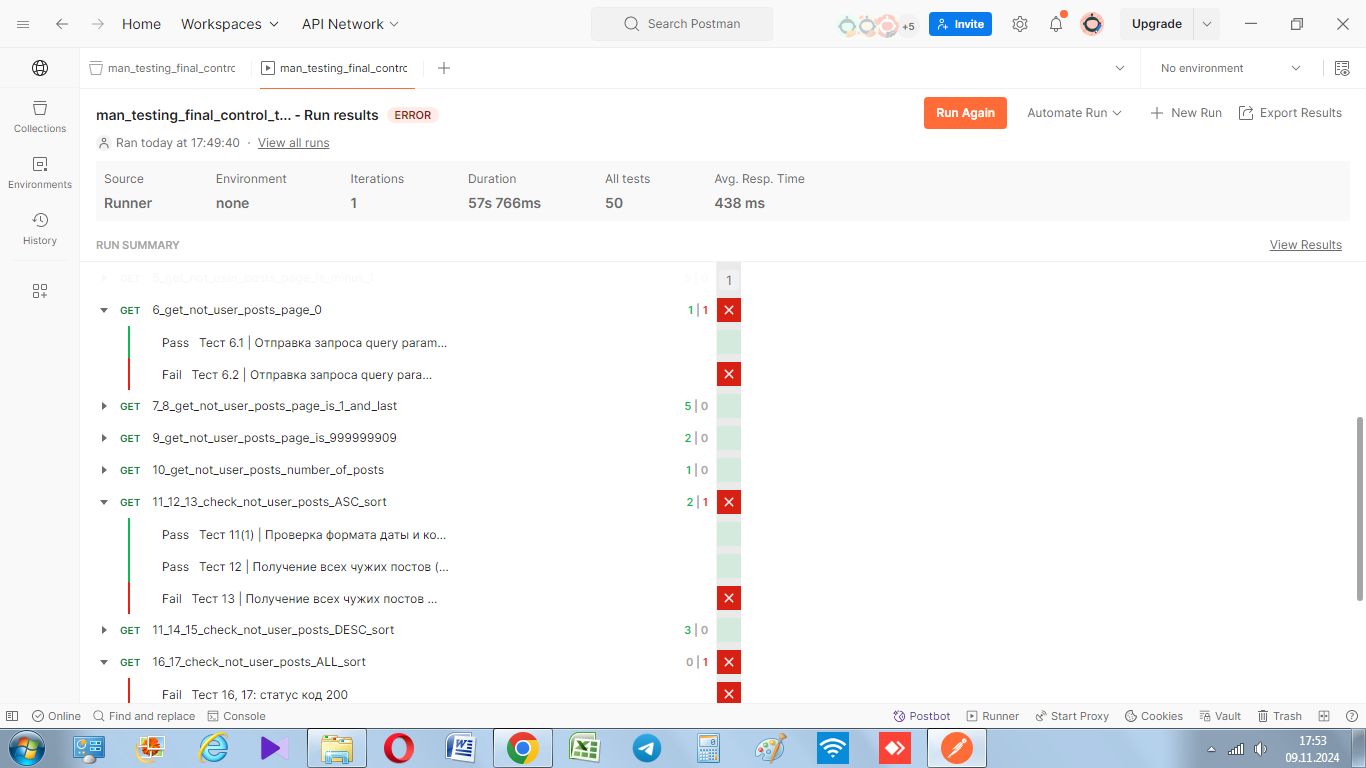


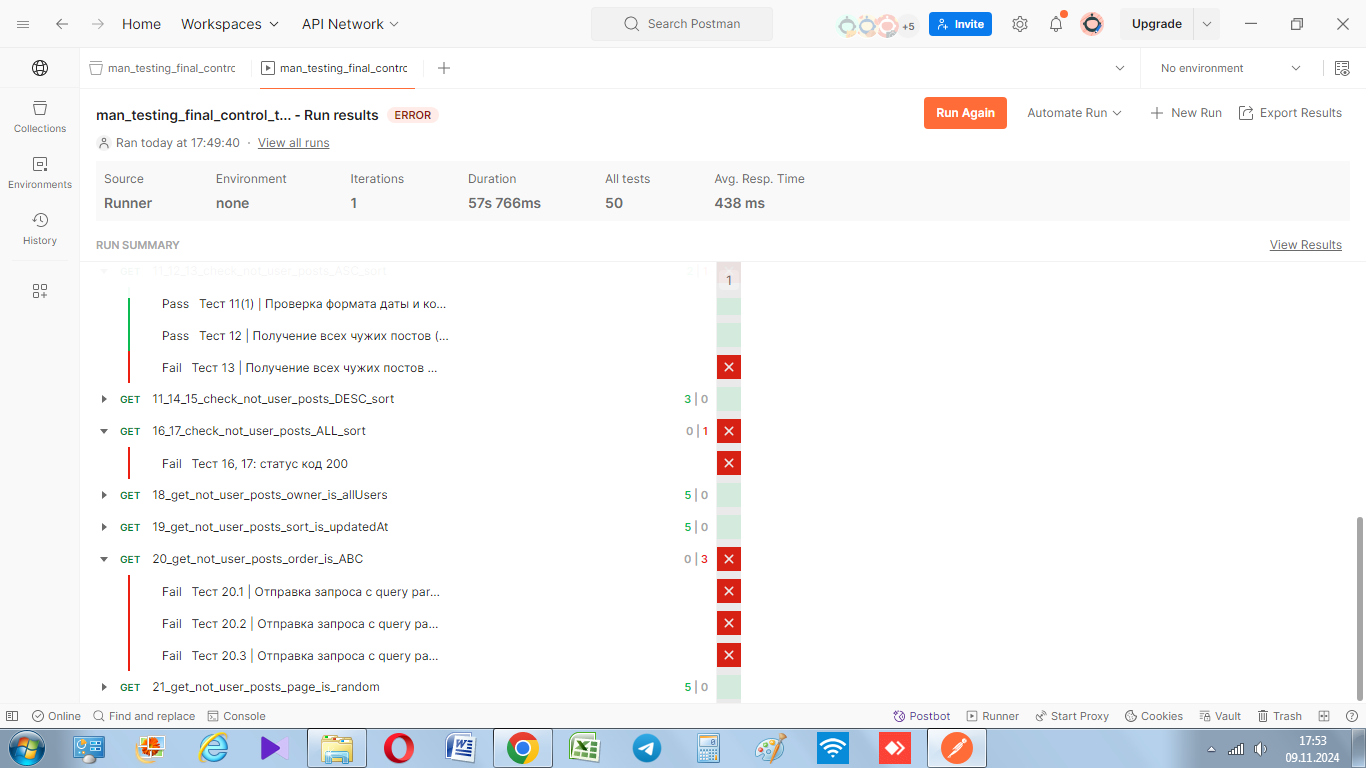




Прогон тестов API ленты чужих постов:







Задание 3. Статистика по дефектам.

Заведенные баг-репорты можно найти по следующему адресу:

[Тестирование: итоговая аттестация - API (Тимофеев Д. А.)](https://docs.google.com/spreadsheets/d/1G3WBtq4C_Lly2M8rlLHTL20WCZxpJBV5S4uRj42ruNA/edit?gid=1643944151#gid=1643944151)

Сводка по багам API ленты своих постов::

* 4 - major
* 2 - critical

Сводка по багам API ленты постов::

* 3 - major
* 2 - critical