**Продуктовые требования**

**“GeekTest – закрытая социальная сеть для тестировщиков”**

**\*запрашивать доступ к файлу не надо\***

Информационная система "GeekTest – закрытая cоциальная сеть для тестировщиков" предназначена для общения начинающих и опытных QA-специалистов.

**1.1. Авторизация**

Страница авторизации доступная по следующему адресу: <https://test-stand.gb.ru/login> .

**Пользовательский сценарий**

**Пользователь:** вводит логин и пароль

**Система:** проверяет пару логин и пароль (креды) с БД. Если креды есть, то авторизует пользователя и переводит на страницу Блога

**ФЛК полей**

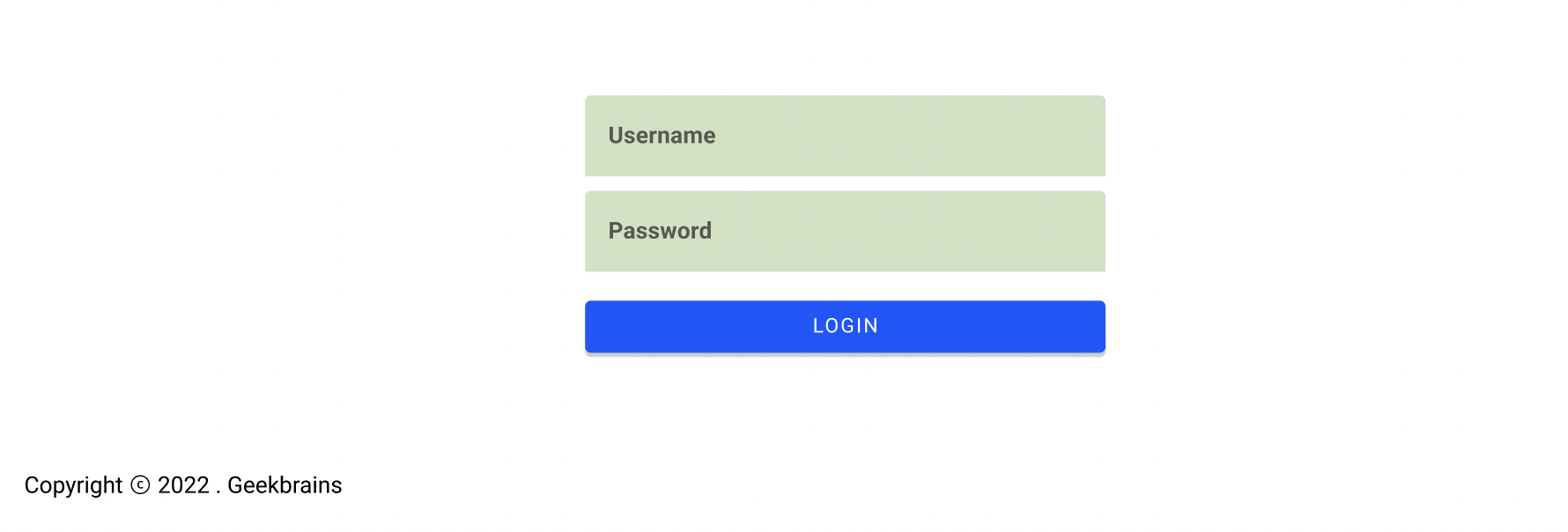
| **Требование** | **Поля** | **Текст ошибки** |
| --- | --- | --- |
| Только латинские символов и цифр, без спецсимволов | Только логин | Неправильный логин. Может состоять только из латинских букв и цифр, без спецсимволов |
| Не менее 3 и не более 20 символов в одном поле | Только логин | Неправильный логин. Может быть не менее 3 и не более 20 символов |
| Поле не может быть пустым | Логин и пароль | Поле не может быть пустым |

Проверка ФЛК срабатывает только после нажатия на кнопку Login.

В случае нарушения требований ФЛК запрос на бэкенд не отправляется, выводится текст ошибки под полем (см. дизайн)

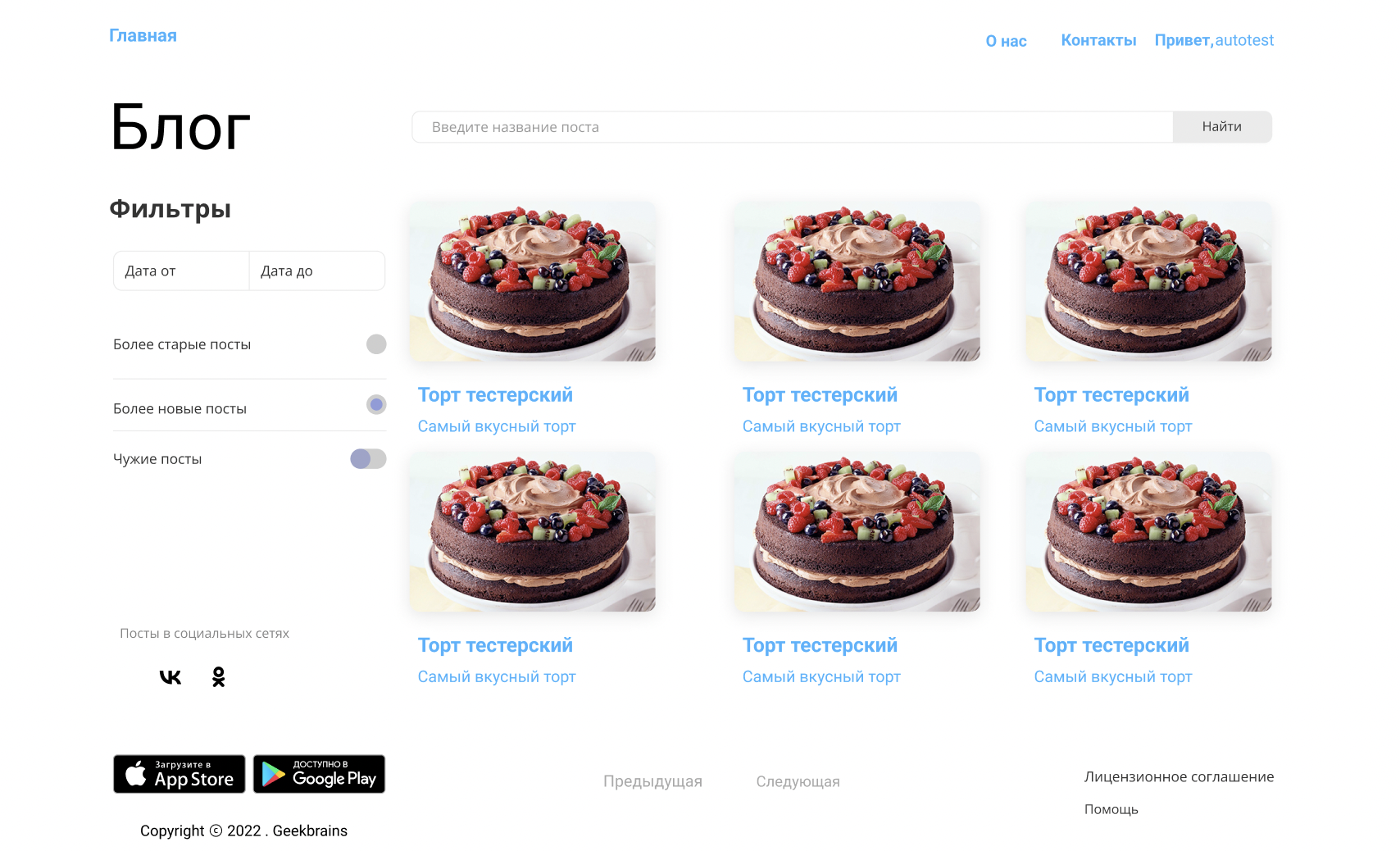
**Ошибки при взаимодействии с бэкендом:**

| **Код ошибки** | **Текст ошибки** |
| --- | --- |
| 404 | Проверьте соединение с интернетом и попробуйте еще раз |
| 401 | Проверьте логин и пароль |
| другие | Ошибка. Попробуйте еще раз позже |



**1.2. Лента своих постов**

Домашняя страница социальной сети. Отображает список превью своих постов и позволяет открыть каждый из них.



**URL:** https://test-stand.gb.ru/

**Дизайн:** [Miro](https://miro.com/app/board/uXjVPXrEyJU=/?share_link_id=733261292455), [Figma](https://www.figma.com/file/XMboQDGNkOBzbRSx0mrwQ8/test-stand.gb.ru-(Copy)?type=design&node-id=0%3A1&mode=design&t=cWYz4MUgRJSrTgfh-1)

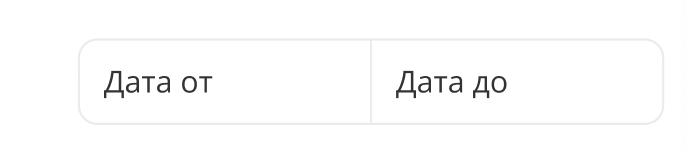
**Пользовательский сценарий**

**Пользователь:** переходит по ссылке или нажимает на кнопку Главная в хедере

**Система:** открывает экран личного блога

**Пользователь:** может просматривать свои посты и переходить к ним, перемещаться по к следующей и прошлой странице. Также пользователь не может добавить свой пост в избранное.

Также пользователь имеет возможность искать посты по датам, воспользовавшись фильтром “Дата от” и “Дата до”.



**Требования, условия и ограничения**

| **Требование** | **Отработка ошибки** |
| --- | --- |
| Превью постов содержит картинку, тайтл и дескрипшн | При отсутствии элемента, он просто не отображается. При отсутствии картинки – заглушка |
| Картинки обрезаются до соотношения сторон 2:3, независимо от исходного формата. При необходимости расширения сторон фон сзади “дозаливается” | - |
| На одной странице отображается 10 постов. Сортировка по дефолту: от новых к старым | - |
| По клику на иконку можно изменить сортировку: от старых к новым | - |

**Техническая реализация**

GET [https://test-stand.gb.ru/api/posts](https://test-stand.gb.ru/api/posts?sort=createdAt&order=ASC)

где в качестве query параметров передаются:

| **параметр** | **принимаемые значения** | **описание** |
| --- | --- | --- |
| **sort** | createdAt | сортировка по дате публикации |
| **order** | [ASC | DESC] | сортировка, где параметр принимает значения ASC | DESC для сортировки в порядке возрастания или убывания. ASC используется по умолчанию |
| **page** | любые значение от 0 до бесконечности | запрос страницы с постами |

Если пользователь запрашивает несуществующую страницу, результатом ответа будет пустой массив data.

| {  "data": [],  "meta": {  "prevPage": "{page\_number}",  "nextPage": "{page\_number}",  "count": {count\_posts}  }  } |
| --- |

где:

**data** – параметр, значение которого является массив с объектами, в которых находится информация по каждому из посту,

**meta** – параметр, значение которого является объект, содержащий информацию про количество страниц и количество публикаций,

**prevPage** – параметр, значение которого является целое число {page\_number} – обозначает предыдущую страницу - тип String,

**nextPage** – параметр, значение которого является целое число {page\_number} – обозначает следующую страницу (если такой страницы нет, то будет null - тип String) - тип String,

**count** – параметр, значение которого является целое число {count\_posts} – указывает на количество постов, тип Integer.

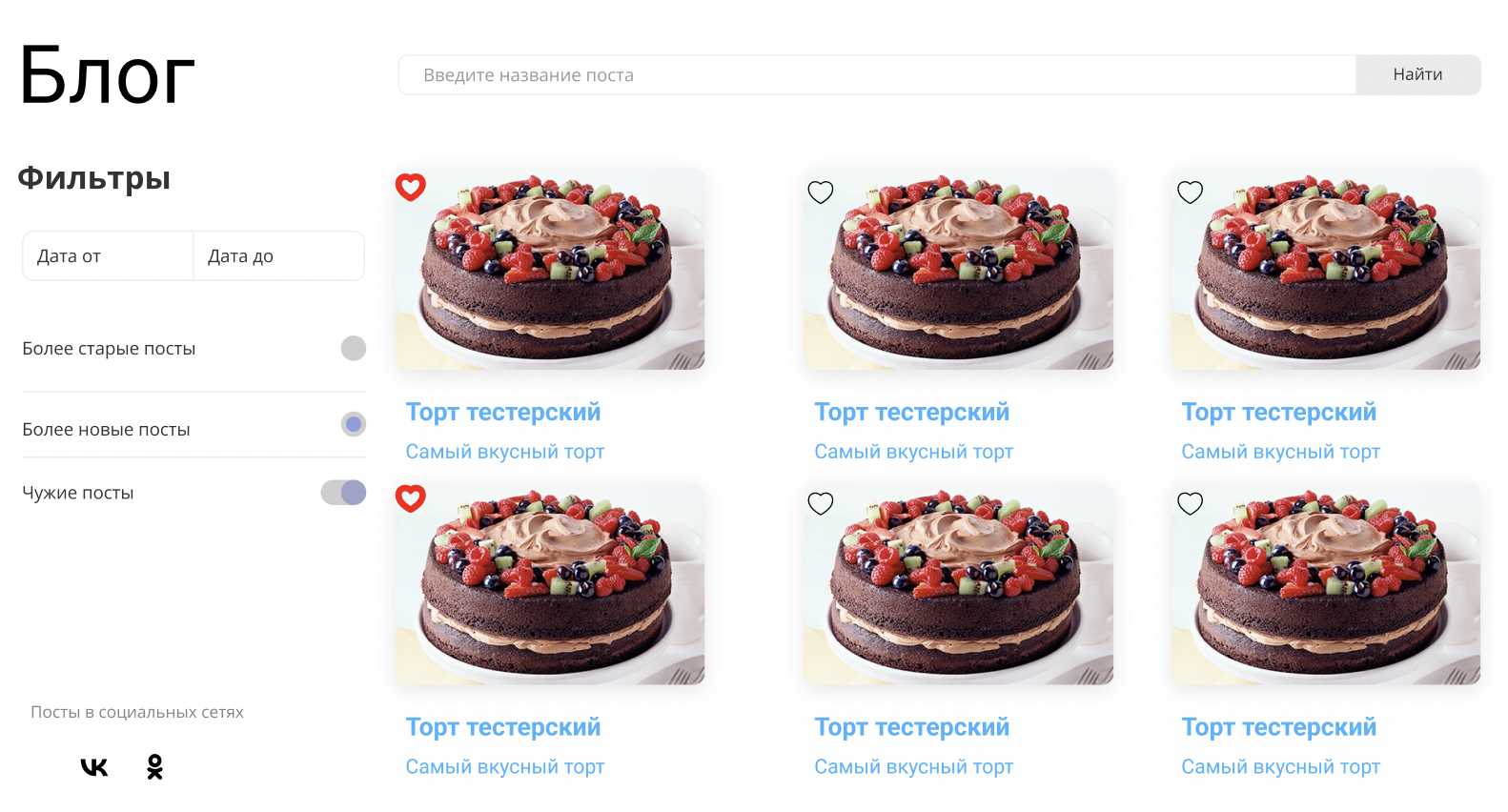
Если пользователь запрашивает страницу на которой есть посты, то результатом ответа будет массив data с объектом или объектами, содержащими информацию про публикации пользователя.

Пример запроса и ответа в Postman (как реализовано сейчас):

|  |
| --- |

**1.3. Лента чужих постов**

Содержит список всех постов, опубликованных в социальной сети, кроме постов автора.



**URL:** https://test-stand.gb.ru/

**Дизайн:** [Miro](https://miro.com/app/board/uXjVPXrEyJU=/?share_link_id=733261292455), [Figma](https://www.figma.com/file/XMboQDGNkOBzbRSx0mrwQ8/test-stand.gb.ru-(Copy)?type=design&node-id=0%3A1&mode=design&t=cWYz4MUgRJSrTgfh-1)

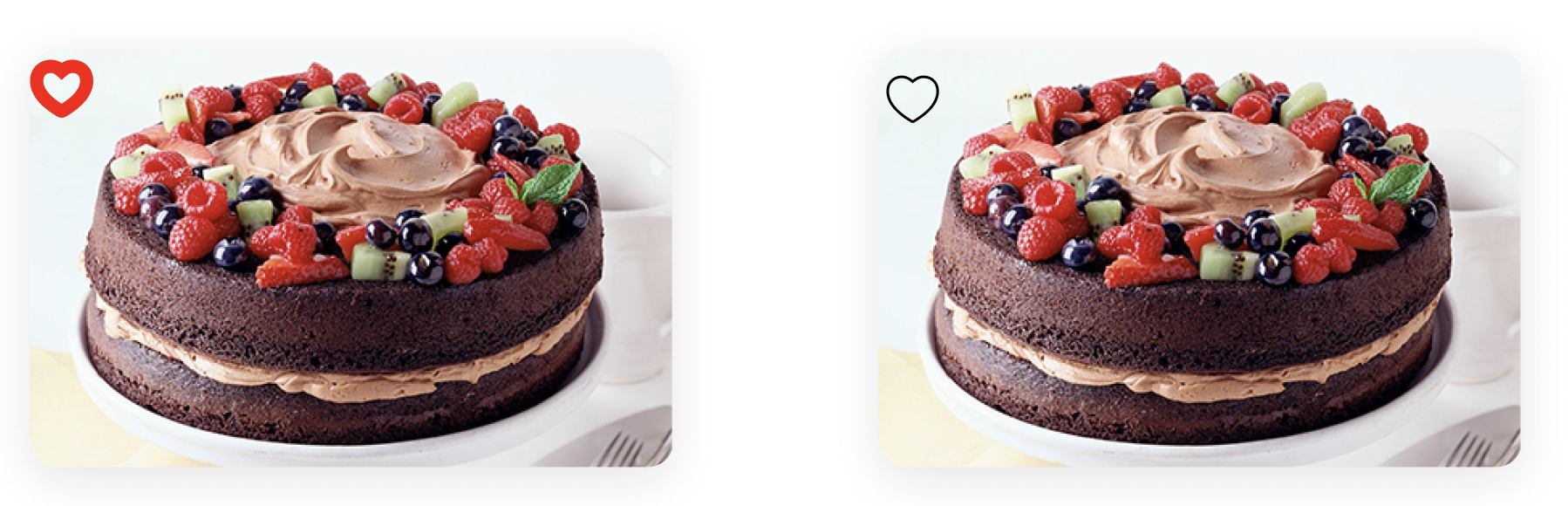
**Точка входа:** тумблер Not my Posts

**Пользовательский сценарий**

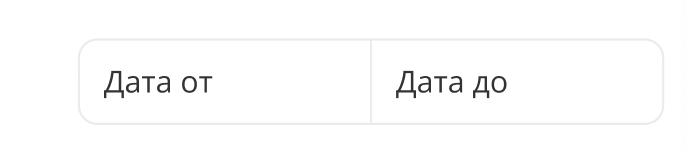
**Пользователь:** переключаеттумблер Not my Posts

**Система:** открывает ленту постов от других пользователей в хронологическом порядке от новых к старым

**Пользователь:** может просматривать чужие посты и переходить к ним, перемещаться к следующей и прошлой странице. Также пользователь может добавить пост в избранное. Если пользователь переключит тумблер “Чужие посты”, то возможность добавления в избранное пропадет. При добавлении в избранное, сердечко закрашивается красным.



Также пользователь имеет возможность искать посты по датам, воспользовавшись фильтром “Дата от” и “Дата до”.



**Требования, условия и ограничения**

| **Требование** | **Отработка ошибки** |
| --- | --- |
| Посты отсортированы в хронологическом порядке от новых к старым | - |
| Превью поста содержит: изображение, заголовок и дескрипшн | - |
| На одной странице максимум 4 объявлений | - |
| Перемещение по странице происходит по кнопкам Previous Page, Next Page. Если следующей, прошлой страницы нет, кнопка дизейблится | - |
| Просмотр постов доступен только авторизованным пользователям | API возвращает ошибку авторизации; фронт редиректит на страницу авторизации |

**Техническая реализация**

GET [https://test-stand.gb.ru/api/posts](https://test-stand.gb.ru/api/posts?owner=notMe&sort=createdAt&order=ASC&page=9)

где в качестве query параметров передаются:

| **параметр** | **принимаемые значения** | **описание** |
| --- | --- | --- |
| **owner** | notMe | параметр, отвечающий за запрос чужих постов пользователей |
| **sort** | createdAt | сортировка по дате публикации |
| **order** | [ASC | DESC | ALL] | сортировка, где параметр принимает значения ASC | DESC | ALL для сортировки в порядке возрастания или убывания, а также вывод всех постов в рандомном порядке. ASC используется по умолчанию. ALL выводит все посты. |
| **page** | любое значение от 0 до бесконечности | запрос страницы с постами |

Если пользователь запрашивает несуществующую страницу, результатом ответа будет пустой массив data.

| {  "data": [],  "meta": {  "prevPage": "{page\_number}",  "nextPage": "{page\_number}",  "count": {count\_posts}  }  } |
| --- |

где:

**data** – параметр, значение которого является массив с объектами, в которых находится информация по каждому из посту,

**meta** – параметр, значение которого является объект, содержащий информацию про количество страниц и количество публикаций,

**prevPage** – параметр, значение которого является целое число {page\_number} – обозначает предыдущую страницу - тип String,

**nextPage** – параметр, значение которого является целое число {page\_number} – обозначает следующую страницу (если такой страницы нет, то будет null) - тип String,

**count** – параметр, значение которого является целое число {count\_posts} – указывает на количество постов.

Если пользователь запрашивает страницу на которой есть посты, то результатом ответа будет массив data с объектом или объектами, содержащими информацию про публикации пользователя, например:

| {  "id": "3",  "title": "пельмешки",  "description": "dfghjkljhvcxvbnm,nbcghkjlkljhgvc",  "content": "dfghjkljhvcxvbnm,nbcghkjlkljhgvc",  "authorId": 21,  "mainImage": {  "id": 8,  "cdnUrl": "http://test-stand.gb.ru/files/public/image/c33c760ff7b4b046b820af8bec744c07.png"  },  "updatedAt": "2022-09-17T09:16:35+00:00",  "createdAt": "2022-09-17T09:16:35+00:00",  "labels": [],  "delayPublishTo": **null**,  "draft": **false**  } |
| --- |

Пример запроса, как это реализовано сейчас:

|  |
| --- |