Тестовое задание для React.js разработчика

# Задача

Разработать максимально удобный для переиспользования, расширения и поддержки компонент состоящий из вариативно используемых функций.

## Список функций

* Фильтрация
  + Первичная (инпуты сразу на странице)
  + Вторичная (инпуты скрываются под кнопкой “Другие фильтры” и находятся в модальном окне)
* Кнопки действий  
  Результат нажатия - модальные окна с дальнейшими действиями.
  + Создать
  + Просмотреть
  + Редактировать
  + Удалить
* Табличный вид отображения данных
* Карточный вид отображения данных
* Пагинация

# Дополнительные требования

* Время выполнения задания не ограничено.
* Каждая из функций может быть не обязательной. Это должно быть настраиваемо программистом.
* Используй [Grommet](https://v2.grommet.io) UI библиотеку.
* Интегрироваться обязательно с любым из RESTful открытых API.
* Результат задания должен быть выложен в отдельный открытый репозиторий на Github.
* В репозитории в корне директории должен быть файл README.md со следующим содержанием:
  + ФИО.
  + Дата и время:
    - начала работы над тестовым.
    - конца работы над тестовым.
  + Сумма в часах затраченная на выполнение задания.
* Запускаться результат выполнения задания должен одной командой - “npm install && npm run start”.
* У разработчика, который будет пользоваться твоим компонентом должна оставаться возможность повлиять на содержание компонента, например, добавь кнопку, форму, модалку и т.д. без переписывания твоего компонента и копипасты исходников твоего компонента.

# Подсказки

* Наличие некоторых функций может влиять на поведение других функций. Каких? Сам(а) подумай.
* Авторизацию/Регистрацию не делай, так как нет требований.
* Реализуй множественные действия.
* Для пользователя при использовании твоего компонента не должно быть лишних кнопок, селекторов и т.д. которые просто “технически” ты не смог(ла) убрать.
* Самое главное показать подход к разработке и соблюдать описанные требования. Оверхед приветствуется, но необязателен.
* Можешь оставлять в этом файле комментарии - мы ответим на любые вопросы.
* Конфигурировать все возможности как пропсы одного компонента - плохой паттерн. Так как слишком много возможностей есть у компонента и пропсы будут выглядеть как огромная портянка.