

Задача

1. Используя схему API с плейграунда <http://152.228.215.94:82/graphql-playground>, сгенерировать TypeScript типы и даталоадеры с помощью www.graphql-code-generator.com. Предпочтение отдается реализациям не использующим Create React App.
2. Создать переиспользуемый компонент CustomSelect на основе @mui Autocomplete. (с кнопкой добавления нового элемента расположенного внутри списка; фильтрация списка сохраняется даже при добавлении нового элемента; должна быть возможность отменить добавление элемента в список; после добавления нового элемента он становится выбранным)
3. Создать 2 контролла использующих CustomSelect, options должны быть загружены через сгенерированные даталоадеры от API используя следующие запросы:

- запрос Relations

```
{
  applicantIndividualCompanyRelations {
    data {
      id
      name
    }
  }
}
```

- запрос Positions

```
{
  applicantIndividualCompanyPositions {
    data {
      id
      name
    }
  }
}
```

Создать строго типизированную форму произвольного дизайна (можете показать свой скилл владения @mui или css) используя форму из библиотеки react-hook-form. Форма должна состоять из полей: Два контролла из п.3, TextInput и TextArea

При ошибках валидации поле должно становиться красным, текст ошибки внутри поля. Один из контроллов п.3 должен иметь prop: multiple=true. Submit формы должен выводить ее содержимое в консоль. Не допускать any types, и использовать linter typescript

Вопросы

Отвечать в произвольной форме в 3-4 предложениях.

1. В чем преимущества graphql перед rest/rpc для клиентских приложений?
2. Чем хороша локализация бизнес логики и отображения внутри компонента? Как достичь этого для удаленных данных?