

Тестовое задание на позицию Vue разработчик (Уровень Middle)

Рекомендуемое время выполнения: 3-4 часа

Общие требования

На фреймворке VueJS без использования любых дополнительных **client-side библиотек**, кроме указанных реализовать приложение, выложить в приватный репозиторий и открыть доступ для аккаунта matveyson@mail.ru ([GitHub.com](https://github.com)) или mrmotvik@gmail.com ([GitLab.com](https://gitlab.com)). Обязательно укажите, сколько времени потребовалось!

Перечень разрешенных библиотек

- VueJS@2 / VueJS@3
- Vue Router
- Vuex (Реализация state-management без vuex на js/ts + composition api будет плюсом)

Использование любых плагинов для библиотек **не разрешено!**

Использование стороннего js/ts-кода, в том числе любых других импортируемых скриптов/библиотек/фреймворков/утилит для **не разрешено!**

CSS библиотеки **не разрешаются!**

SVG и картинки **можно** грузить с CDN или включить напрямую в проект

Описание приложения

Приложение предназначается для работы в стиле CRUD над двумя сущностями: пользователями и группами. Приложение содержит две страницы (Пользователи и Группы) на каждой из которой есть таблица с соответствующими сущностями. В качестве api используется REST-API сервер, следует использовать [json-server](#). Код для запуска сервера и генерации базы приведен в конце.

- Должен быть реализован и использован компонент таблицы (описано далее);
- Должен быть state-management
- Должна быть реализована валидация
- Должно быть четкое разграничение ролей между компонентами
- Компоненты должны быть спроектированы с целью максимальной переиспользуемости
- Все действия с данными в приложении должны использовать персистентными (т.е. использовать api - получать, создавать, изменять и удалять соответствующие записи на backend)
- Получение списка с api должно использовать пагинацию по 15 записей

Описание crud таблицы

Компонент должен отображать таблицу. Сверху компонента должна быть кнопка добавления новой записи. В конце каждой строки должны быть кнопки для изменения и удаления. При нажатии кнопки добавления или изменения должна отображаться форма (во всплывающем окне) для создания /

редактирования сущности (в случае редактирования - с заполненными полями).

- Форма должна производить валидацию.
- Таблица должна содержать пагинацию.
- Таблица должна содержать возможность сортировки (по всем полям, кроме идентификатора и email).
- Дата и время должно отображаться в формате 2021-01-01 13:00

Описание сущностей

Между пользователями и группами связь Many-To-One - у группы могут быть пользователи (а может и не быть), но пользователь принадлежит одной и только одной группе;

Пользователь

Сущность "Пользователь" должна хранить следующие поля:

1. Идентификатор;
2. Имя;
3. Email;
4. Баланс;
5. Идентификатор группы;
6. Дата и время создания (также можно редактировать).

Валидация согласно смыслу полей и комментариям. Все поля обязательные.

Группа

Сущность "Группа" должна хранить следующие поля:

- Название группы;
- Комментарий;
- Дата и время создания (также можно редактировать).

Валидация согласно смыслу полей и комментариям.

JsonServer

Запуск с помощью json-server db.js

Содержимое db.js:

```
1 const names = ["Oleg", "Alex", "Dan", "Robert", "Artem", "Rick"];
2 const domains = ["mail.org", "mail.com", "mail.ru", "mail.net", "mail.uk"];
3 module.exports = () => ({
4   groups: Array.from({ length: 10 }, (v, k) => ({
5     id: k + 1,
6     name: `Group ${k + 1}`,
7     comment: "Lovely group",
8     createdAt: new Date(2021, 0, k % 28, k % 12),
9   })),
10  users: Array.from({ length: 100 }, (v, k) => ({
11    id: k + 1,
12    name: names[k % names.length],
13    email: `user${k + 1}@${domains[k % domains.length]}`,
14    balance: Math.ceil(Math.random() * 1e4),
15    groupId: k % 9 || 1,
```

```
16     createdAt: new Date(2021, 0, k % 28, k % 4),
17   })),
18 });
```

Желаем удачи!

Тестовое задание на позицию Vue разработчик (Уровень Senior)

Рекомендуемое время выполнения: 4 увлекательных часа

Тестовое задание на уровень Senior расширяет тестовое задание на уровень Middle, указанное выше.

Дополнительные требования:

- Можно использовать только Vue (Нельзя использовать Vuex и Vue Router);
- Для хранилища и работы с api требуется использовать service worker;
- Пагинация в таблице должна осуществляться с помощью Infinite Loader (следующая страница подгружается автоматически при появлении последней строки на экране, строки добавляются в конец, необходимо отображать спиннер при загрузке данных);
- У таблицы должен быть переключатель отображения в режим отображения сводной таблицы (pivot table). Сводная таблица должна быть активна только на таблице пользователей.
Поле в ячейке, агрегационная функция (минимальное, максимальное, среднее арифметическое, среднее геометрическое), а также поля заголовков строк и столбцов должны конфигурироваться.
Допустимые поля для заголовков строк и колонок: Имя, Группа, Дата создания(только дата, без времени), Домен Email;
- Данные для pivot table должны также обрабатываться в service worker;
- Будет оцениваться скорость работы (performance)

Будет плюсом

- Возможность указания вложенных полей (те несколько рядов заголовков колонок, с указанием colspan для верхних рядов, аналогично для заголовков строк)
- Возможность работы в условиях переменной работоспособности доступа к api
- Наличие каких-либо тестов разной специфики

Желаем удачи!