

[🏠](#) / [home](#) / [monitoring](#) / [manual3](#)

Тестовые задания (frontend)

Для поиска кандидатов на стек: Vue.js 3, TypeScript, Echarts, Maptalks/Leaflet

Тестовое задание № 1

Реализовать взаимодействие **Списка** и **Карты**

Стек

1. TypeScript
2. VueJS 3+
3. Leaflet/Maptalks/Openlayers — на выбор

Базовые компоненты

1. Список точек;
2. Карта;

Функционал

Для успешной реализации тестового задания, необходимо проявить ваши способности владения функциями, типизацией, чистоту и организацию кода, **DRY**

Список

1. Сгенерировать **10 000** точек;
2. При прокрутке Списка, не должно быть зависаний - реализовать **virtual/infinite scroll**;
3. Поиск по точкам в Списке;
4. Одиночный и множественный выбор точек по клику на checkbox (подсветка фиолетовая);
5. Подсветку активной точки (current).

По-умолчанию должна подсвечиваться первая точка в Списке.

Активная точка может быть только одной и назначается по клику на строку (row) Списка.

Карта

1. По умолчанию должна быть показана первая точка из Списка + tooltip, содержащий сведения об этой точке.
2. Должна отображать все выбранные точки из Списка, а при снятии выбора - их скрывать.
3. При клике на точку, расположенную на карте, в Списке она должна подсветиться и прокрутиться на ее позицию (эта точка должна оказаться по центру Списка).

Условия

При решении тестового задания, обязательным является:

1. Взаимодействие с хранилищем;
2. Типизация в функциях, переменных и т.д.;
3. Реализация моделей (списка, карты);
4. Уверенная демонстрация работы с компонентами

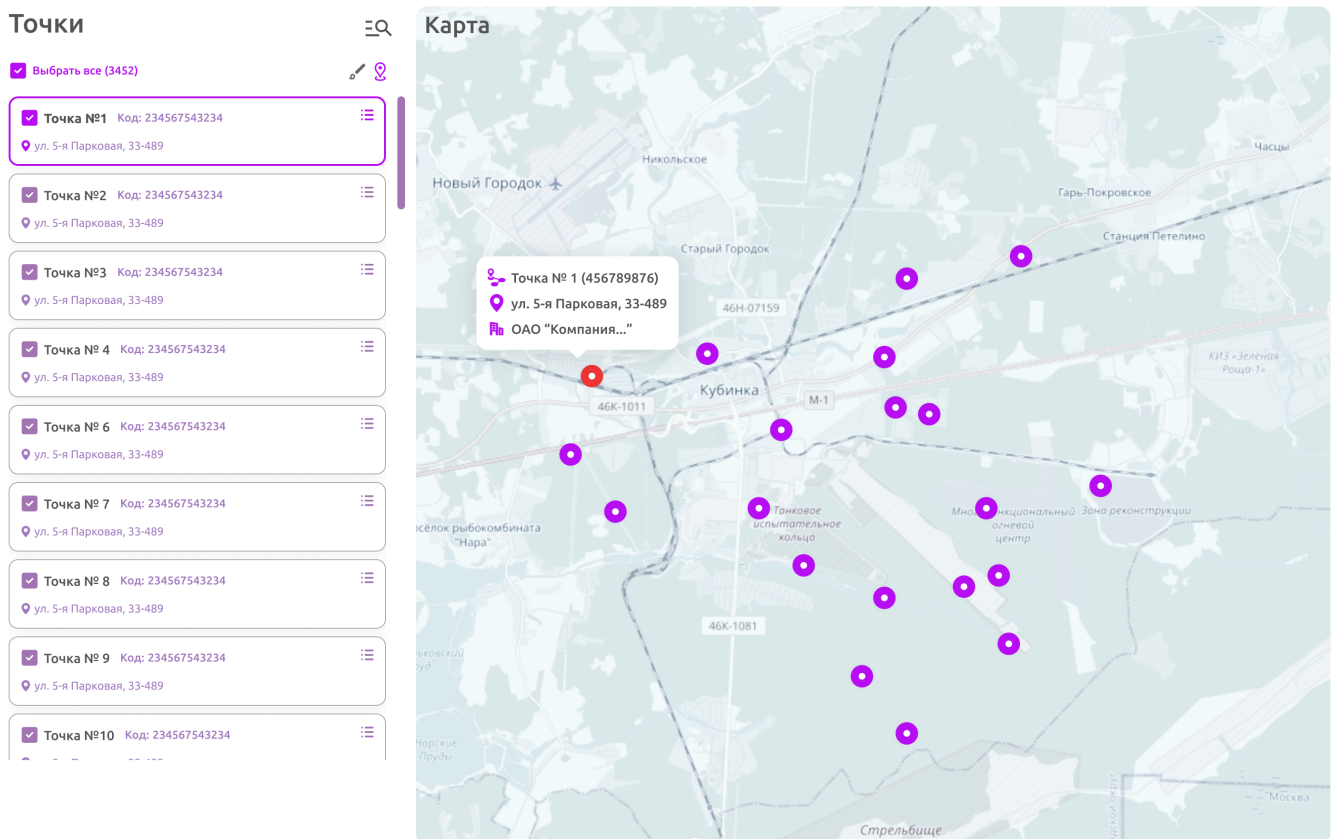


При проверке задания, в первую очередь будет рассмотрена уверенная работа с функционалом. Не стоит делать упор на стили

Рекомендуемая последовательность действий

1. Описываем модель точек в файле (.ts), желательно через class, не interface. Создаем необходимые методы для работы со Списком и Картой, например, `getCurrentPoint()`, `findPoint()`
2. Создаём хранилище для точек. В нем создаём массив из 10 тыс точек через функцию-генератор, не вручную!
3. Выводим точки в компоненте Списка и связываем их с картой.
Для списка можно взять компонент `q-virtual-scroll` (если на Quasar создан проект) - не будет зависеть список на большом количестве или обычный `q-list`.
4. Для карты можно использовать `maptalks`, для `leaflet` документация в РФ не доступна или любую либу на ваше усмотрение.

Внешнее представление



Тестовое задание № 2

Реализовать взаимодействие Списка и Формы

Стек

1. TypeScript
2. VueJS 3+

Базовые компоненты

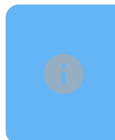
1. Список точек;
2. Форма создания/редактирования;

Функционал

Для успешной реализации тестового задания, необходимо проявить ваши способности владения функциями, типизацией, чистоту и организацию кода, **DRY**

Список

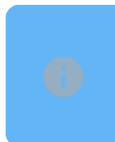
1. Сгенерировать **10 000** точек;
2. При прокрутке Списка, не должно быть зависаний - реализовать **virtual/infinite scroll**;
3. Поиск по точкам в Списке;
4. Одиночный и множественный выбор точек по клику на checkbox (подсветка фиолетовая);
5. Подсветку активной точки (**current**).
6. Сделать кнопку для создания точки в Форме.
7. Созданная точка должна появиться в Списке, с прокруткой на ее позицию. А при редактировании - обновиться данные в Списке (синхронизировать).



По-умолчанию должна подсвечиваться первая точка в Списке.
Активная точка может быть только одной и назначается по клику на строку (**row**) Списка.

Форма

1. При выборе **current** точки в Списке, должна выводиться в полях формы для редактирования.
2. По клику на Сохранить, данные по этой точке должны обновиться в Списке.



Для точки необходимо создать **class**, описать основные поля. Создать базовые методы для редактирования и создания точки и применить их в компонентах.

Условия

При решении тестового задания, обязательным является:

1. Взаимодействие с хранилищем;
2. Типизация в функциях, переменных и т.д.;
3. Реализация моделей (списка, формы);
4. Уверенная демонстрация работы с компонентами



При проверке задания, в первую очередь будет рассмотрена уверенная работа с функционалом. Не стоит делать упор на стили

Рекомендуемая последовательность действий

- 1. Описываем модель точек в файле (.ts), желательно через class, не interface. Создаем необходимые методы для работы со Списком и Формой, например, getCurrentPoint(), findPoint())
- 2. Создаём хранилище для точек. В нем создаём массив из 10 тыс точек через функцию-генератор, не вручную!
- 3. Выводим точки в компоненте Списка и связываем их с формой.
Для списка можно взять компонент q-virtual-scroll (если на Quasar создан проект) - не будет зависать список на большом количестве или обычный q-list.

Внешнее представление

Точки

Выбрать все (3452)

Точка №1

Код: 234567543234

ул. 5-я Парковая, 33-489

Точка №2

Код: 234567543234

ул. 5-я Парковая, 33-489

Точка №3

Код: 234567543234

ул. 5-я Парковая, 33-489

Точка № 4

Код: 234567543234

ул. 5-я Парковая, 33-489

Точка № 6

Код: 234567543234

ул. 5-я Парковая, 33-489

Точка № 7

Код: 234567543234

ул. 5-я Парковая, 33-489

Точка № 8

Код: 234567543234

ул. 5-я Парковая, 33-489

Точка № 9

Код: 234567543234

ул. 5-я Парковая, 33-489

Точка №10

Код: 234567543234

ул. 5-я Парковая, 33-489

Редактирование точки

Основное

Название

Точка № 1

Код точки

12345678765432345

Широта

23,324342

Долгота

47,987789

Радиус

342

Время

Время прибытия с

06:00

Время прибытия по

12:00

Время загрузки

10:40

Время прибытия (факт)

06:00

Время убытия (факт)

12:00

Сохранить

Отмена