

Анализ больших географически распределенных данных через визуализацию на карте



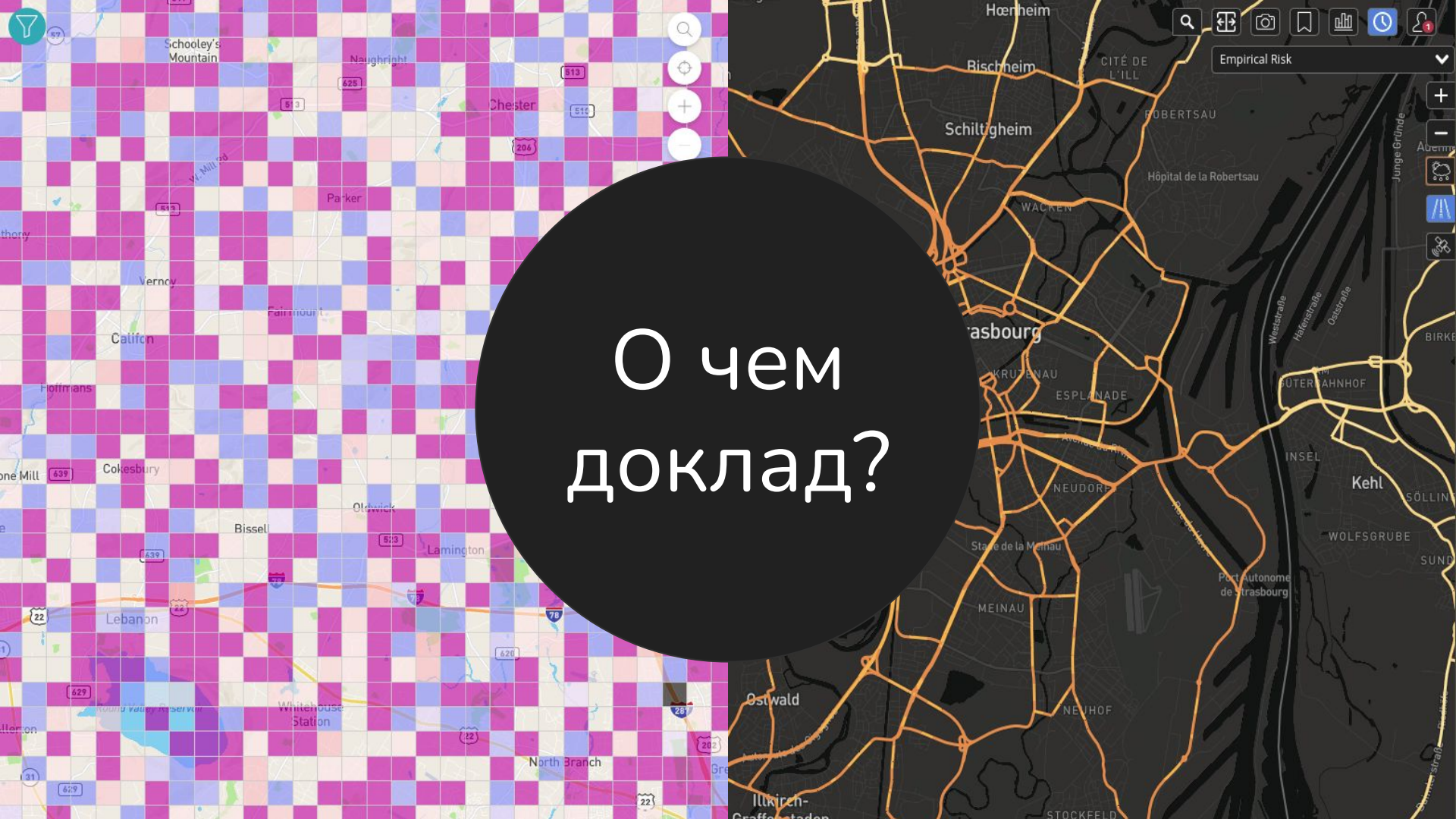
Давайте знакомиться!

Меня зовут **Никита Швыряев**,
я – ведущий разработчик компании **Lineate**.
Наша компания занимается аутсорсингом.

В IT с 2017 года.

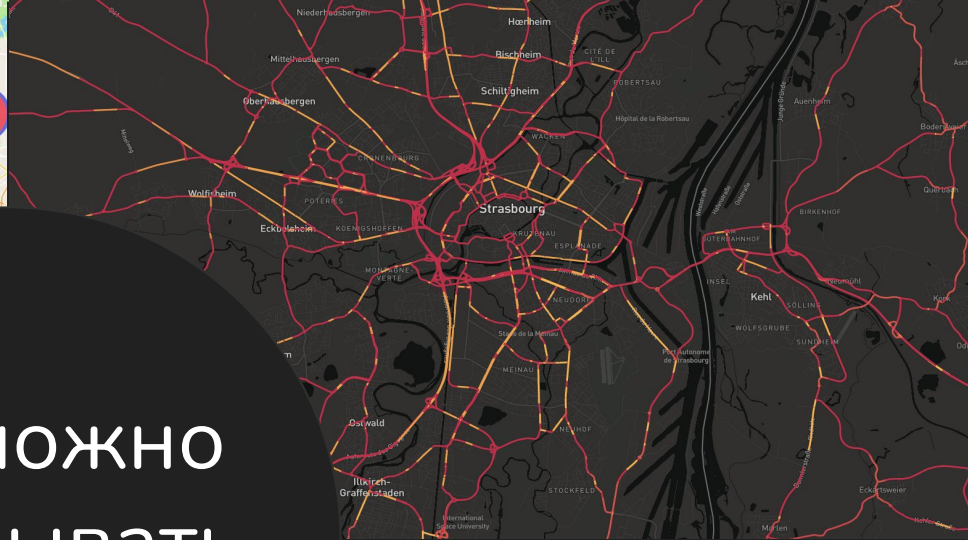
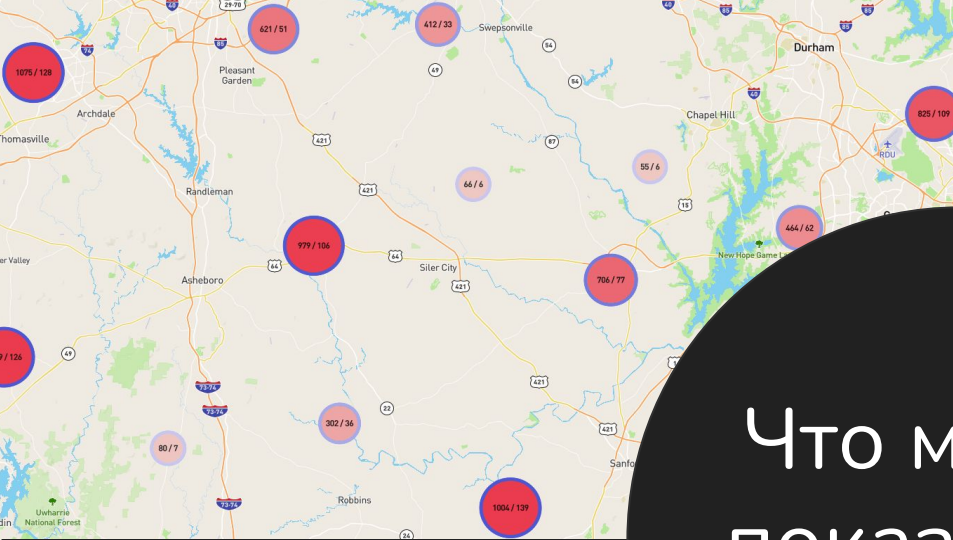
Вел разные проекты, от 1 до 13 человек в команде.

Имею опыт в JS, PHP, развертывании, решении проблем с производительностью приложений.

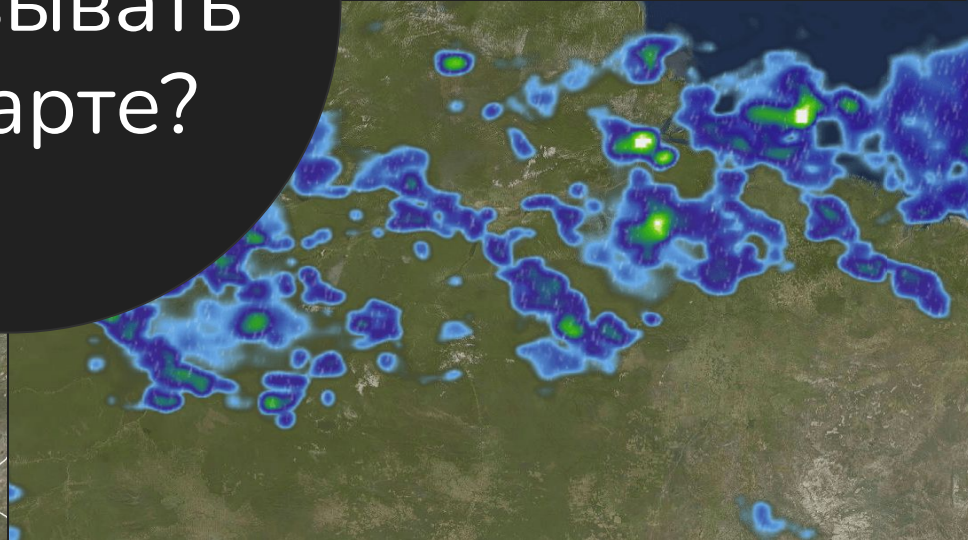
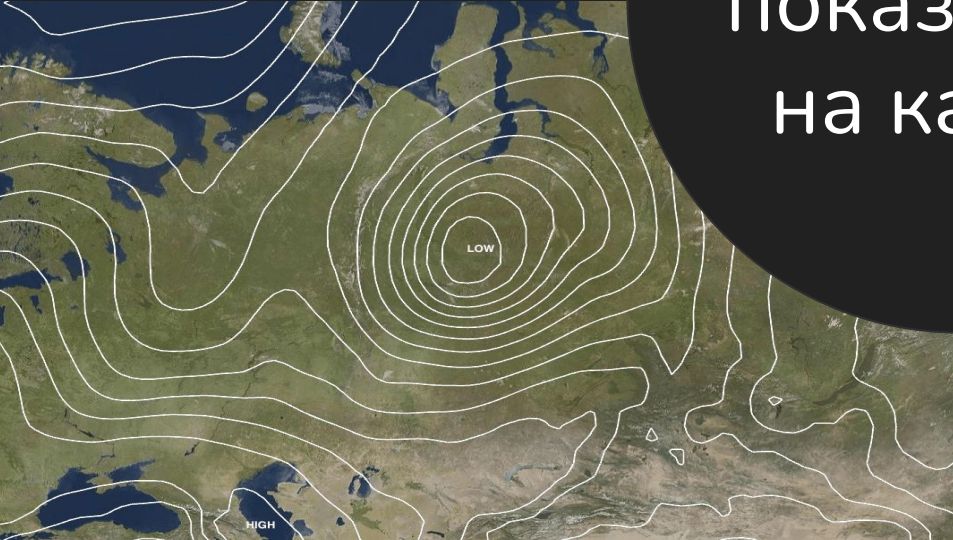


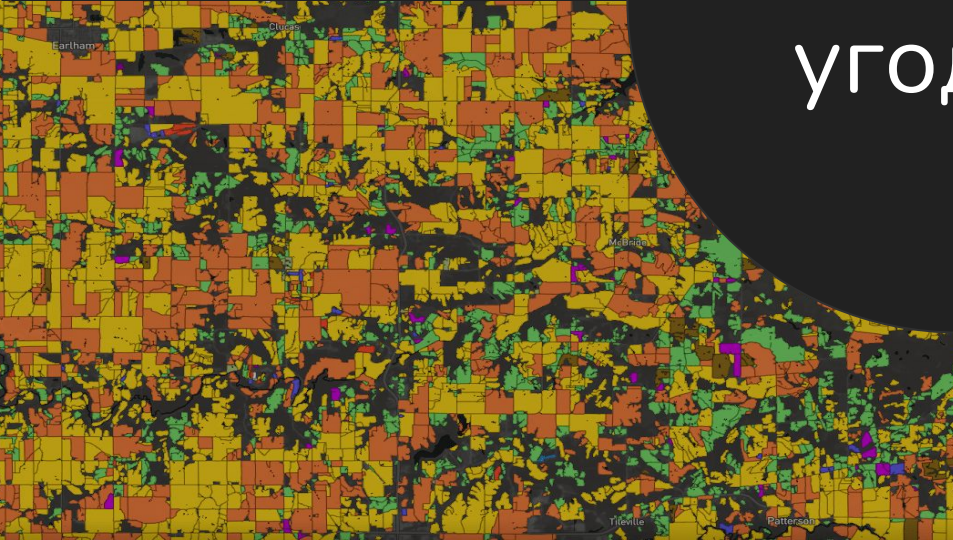
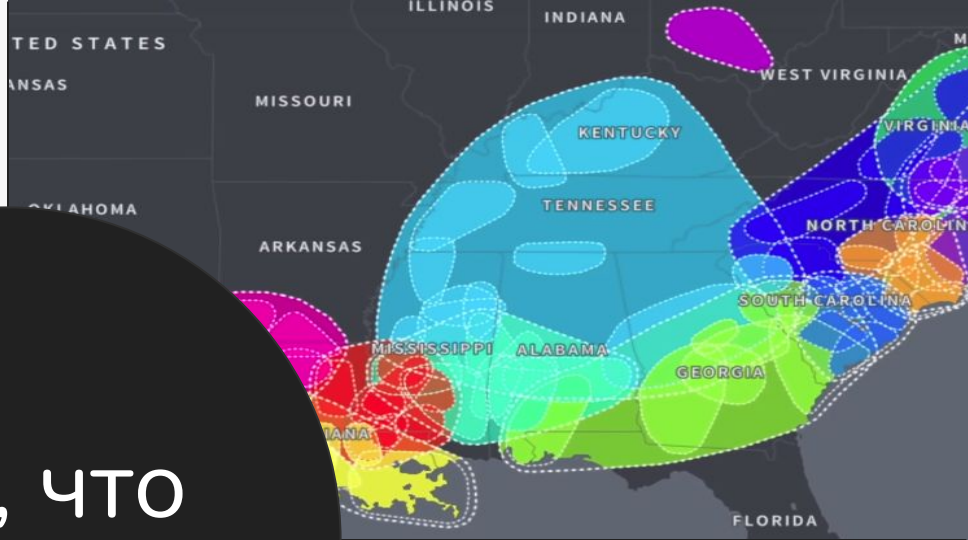
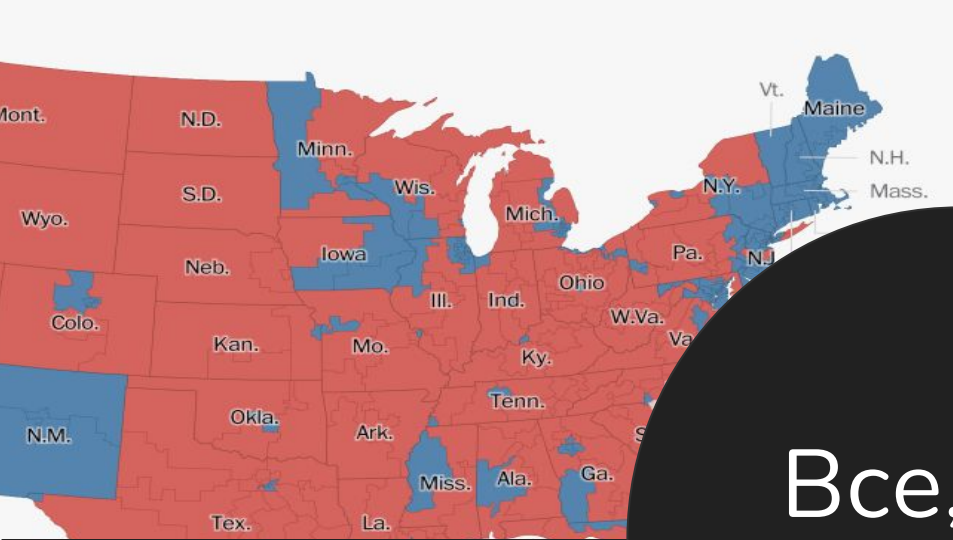
О чем доклад?

Теоретическая часть



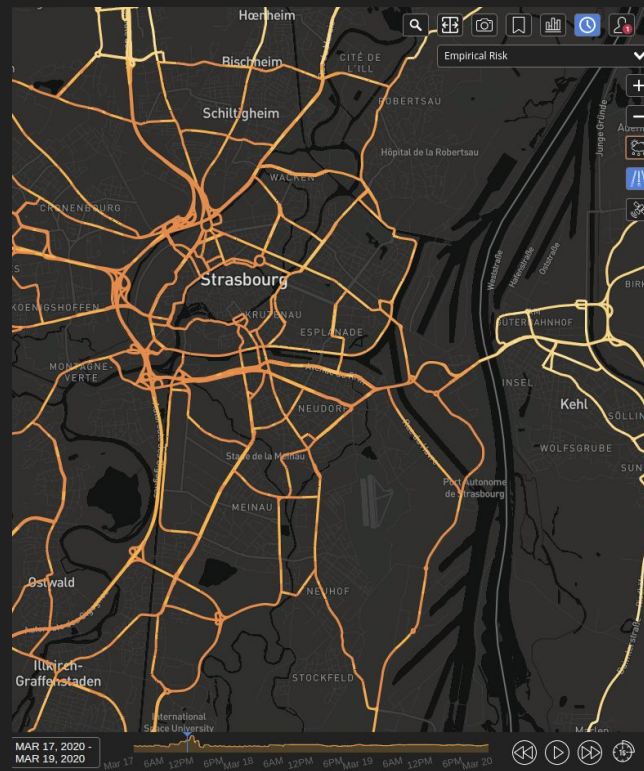
Что можно
показывать
на карте?





Зачем визуализировать данные?

ID	Score	Geometry
cm:segment:144859563	0.394567951	<pre>{"coordinates": [[7.78209,48.56731],[7.7 8242,48.56692],[7.7824 8,48.56686],[7.78257,48 .56676],[7.78272,48.566 61]], "type": "LineString"}</pre>
cm:segment:144859564	0.398361563	
cm:segment:144859565	0.375316516	
cm:segment:144859566	0.426448462	
cm:segment:144859567	0.389515648	
cm:segment:144859568	0.376795156	
cm:segment:144859569	0.394546955	
.....		



Что ждут от визуализации на карте?



Наглядность



Скорость



Аналитика



Сравнение



Сохранение



Проигрывание



Об инструментах

Популярные провайдеры карт



OpenStreetMap
+ *Leaflet* 



Google Maps

Основные возможности карт



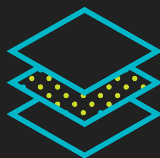
Данные



Стили



Фичи



Слои



События



Навигация

Слои и данные

Векторный



- Vector tiles
- GeoJSON

Растровый



- Image tiles
- Image overlay
- Video

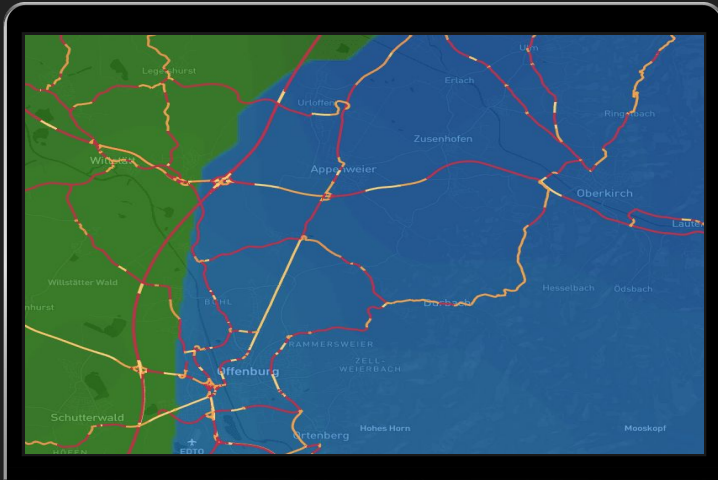
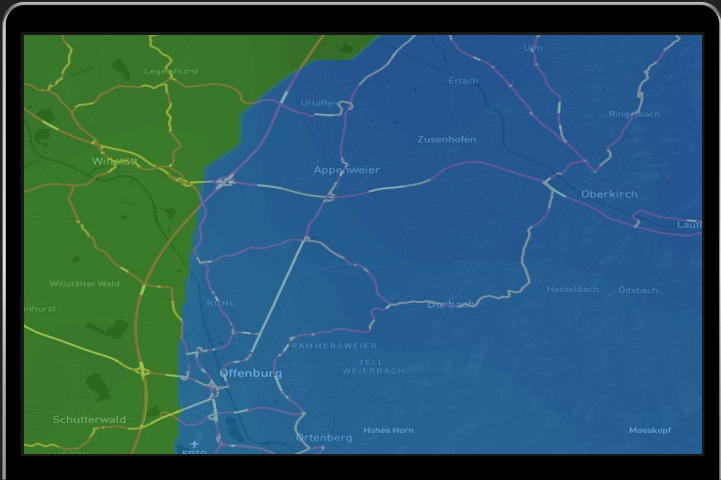


Практическая часть



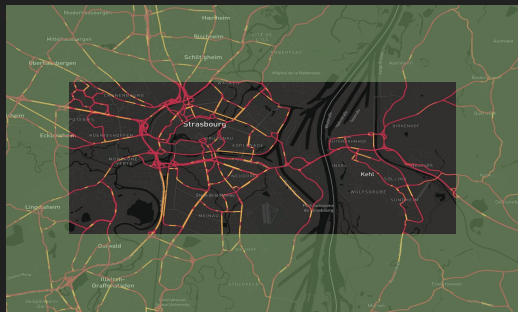
Отображение данных

Помните про порядок слоев...

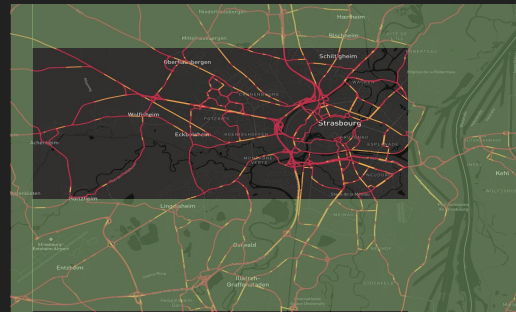


Как показать данные. v1: GeoJSON

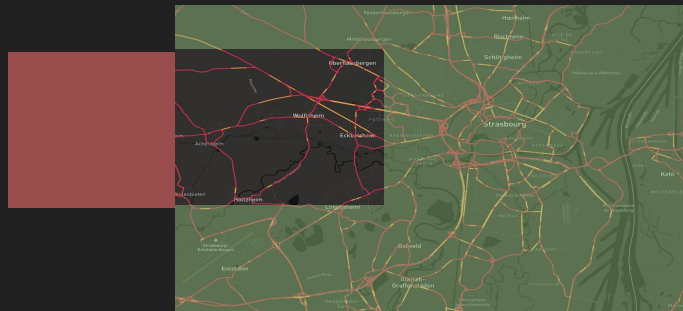
Первая загрузка



Просмотр в зоне без загрузки



Выход из зоны,
перезагрузка



Начиналось все просто...

Frontend:

// bbox - прямоугольник на карте

```
const data = await axios.get('/api/data', { bbox });  
map.addSource('my-geojson-source', {  
  type: 'vector',  
  data,  
});
```

Backend:

// база умеет выполнять поиск по координатам

```
data = db.query('SELECT * FROM data WHERE  
ST_CONTAINS(ST_GEOGFROMTEXT(@bbox), data.centroid)')  
response = as_geojson(data)
```

Но система усложнялась. Frontend тоже



Логика

Обработка
данных

Определение
границ

Отслеживание
загрузки

Кэширование



Проблемы

Перезагрузка части
ранее загруженных
данных

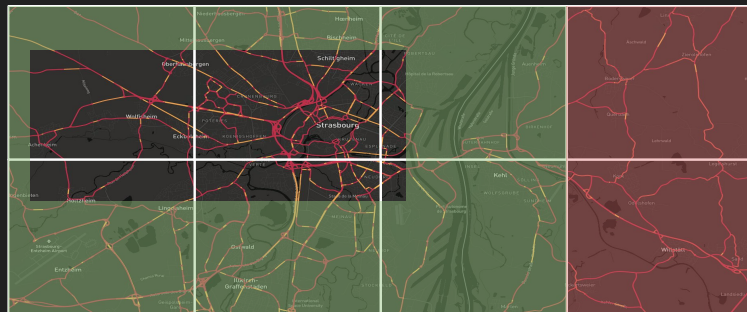
Ожидание одного
большого запроса

Как показать данные. v2: Tiles

Первая загрузка



Загрузка дополнительных сегментов



Перемещение без загрузки



Сегменты
временно
закэшированы

v2. Запрашиваем тайлы. Frontend

```
map.addSource('my-tile-source', {  
  type: 'vector',  
  tiles: [ `/api/data/tile/{z}/{x}/{y}?&{filter}` ],  
});
```

Собственно, все... Всю работу делает библиотека карты.

Позже ещё пришлось добавить отслеживание загрузки, но это потом.

v2. Рисуем векторные тайлы. Backend

```
bbox = to_bounding_box(z, x, y)
```

```
// запрос в базу остался прежний
```

```
data = db.query('SELECT * FROM data WHERE  
ST_CONTAINS(ST_GEOGFROMTEXT(@bbox), data.centroid)')
```

```
response = as_vector_tile(data)
```

```
def as_vector_tile(data) -> list:
```

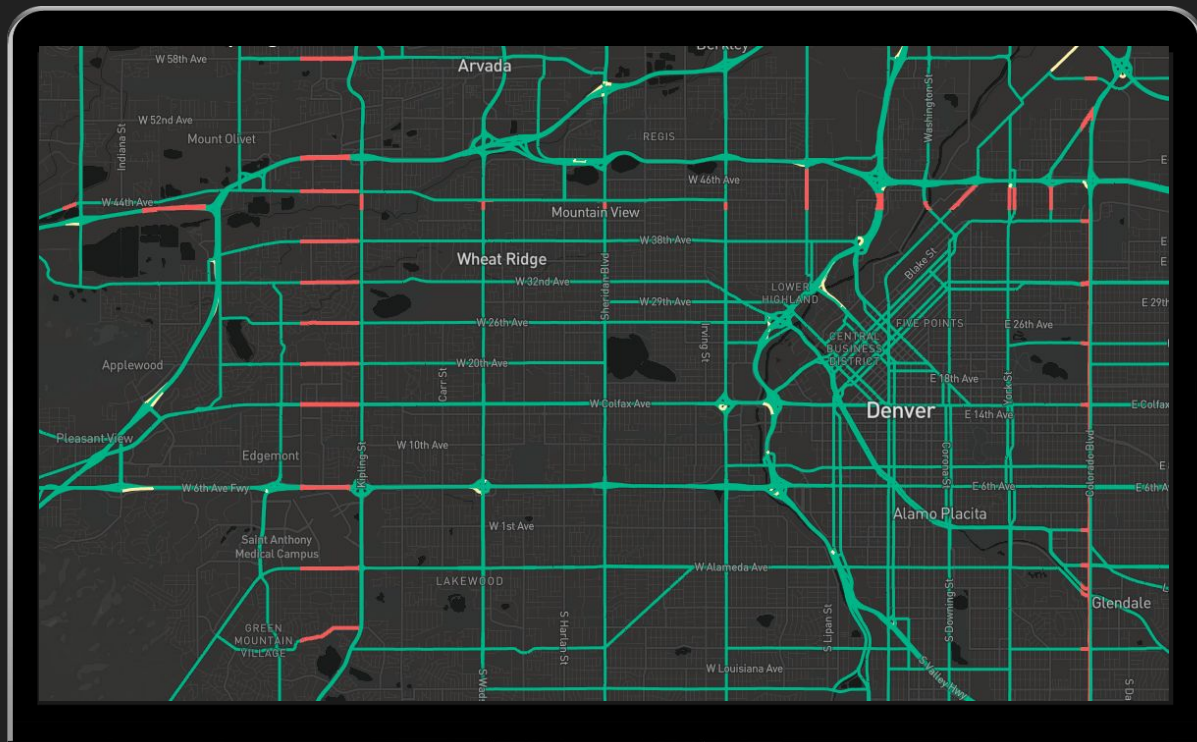
```
    return mapbox_vector_tile.encode([ {
```

```
        'name': 'road_segments',
```

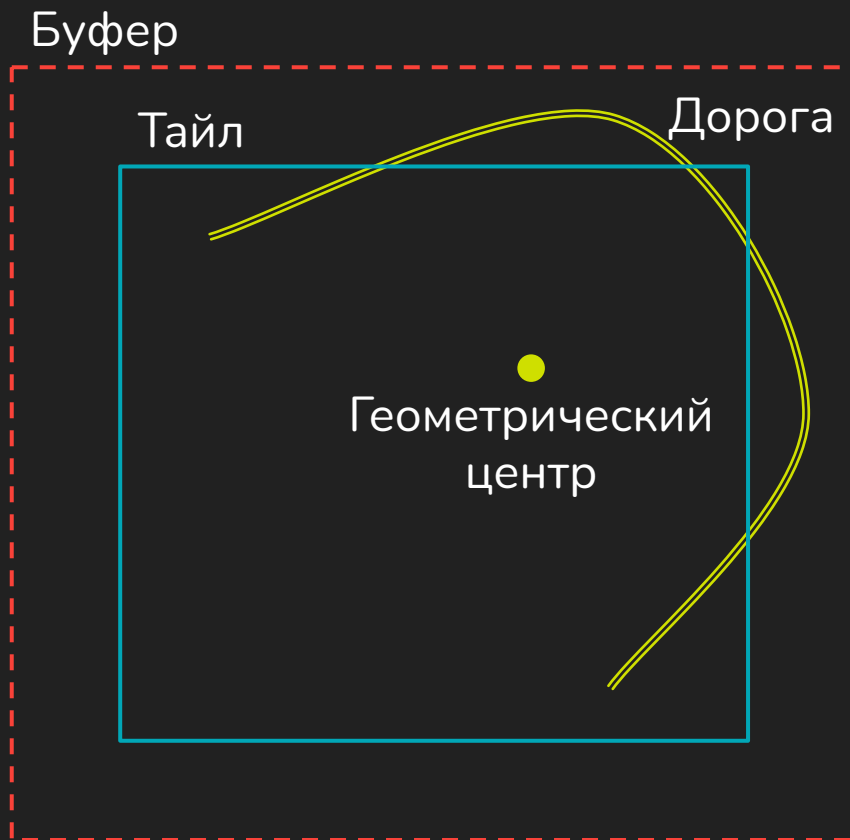
```
        'features': [as_geojson(row) for row in data]
```

```
    ]])
```

Осторожно! У тайлов острые края



Поиск данных в базе по позиции



Когда загрузка налажена, пора фильтровать

Критерии



Время



Структура



Значение

Порядок действий



Обновляем фильтр



Теряем старые данные



Загружаем новые данные



Показываем новые данные

Объем данных и производительность

10 тыс. точек



100 MB

100 тыс. точек



450 MB

Ограничения

10 000	100 MB	Мгновенно
--------	--------	-----------

30 000	250 MB	Заметно
--------	--------	---------

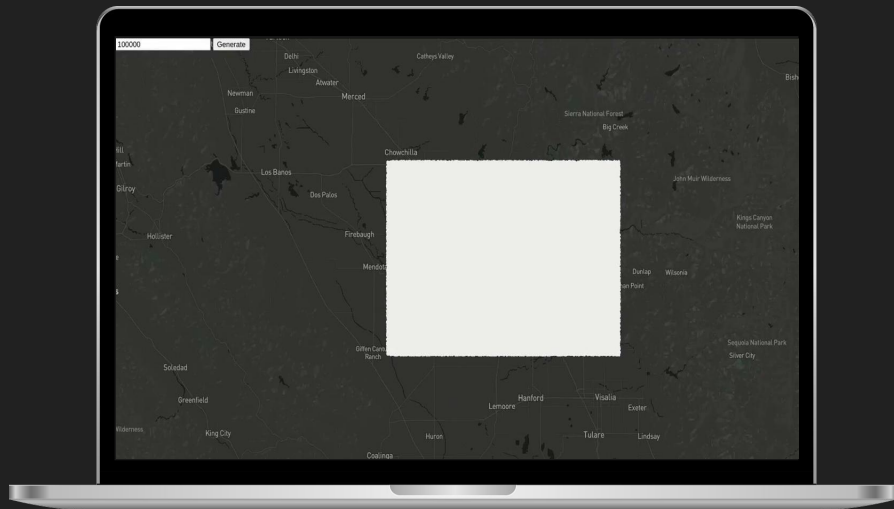
100 000	450 MB	Медленно
---------	--------	----------

200 000	800 MB	Не стоит...
---------	--------	-------------

Агрегация данных

Кластеризация на UI

100 тыс. точек



450 MB

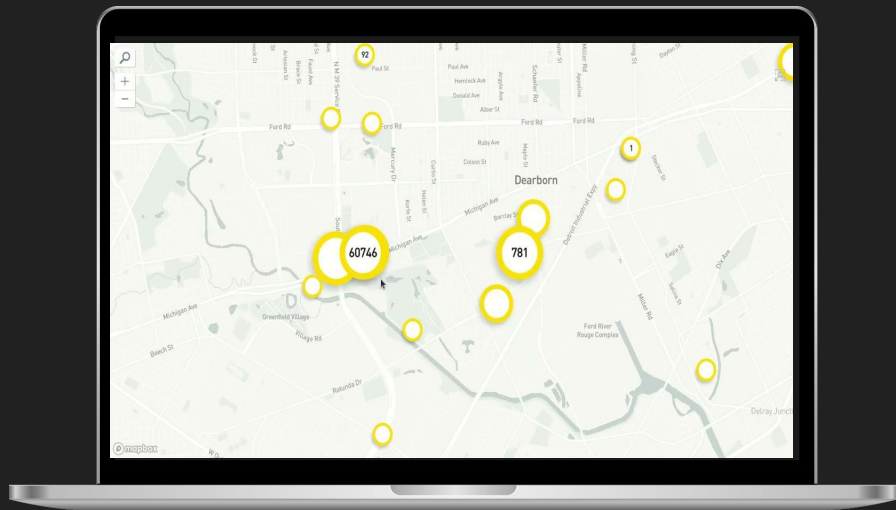
100 тыс. точек



100 MB

Агрегация на бэкенде

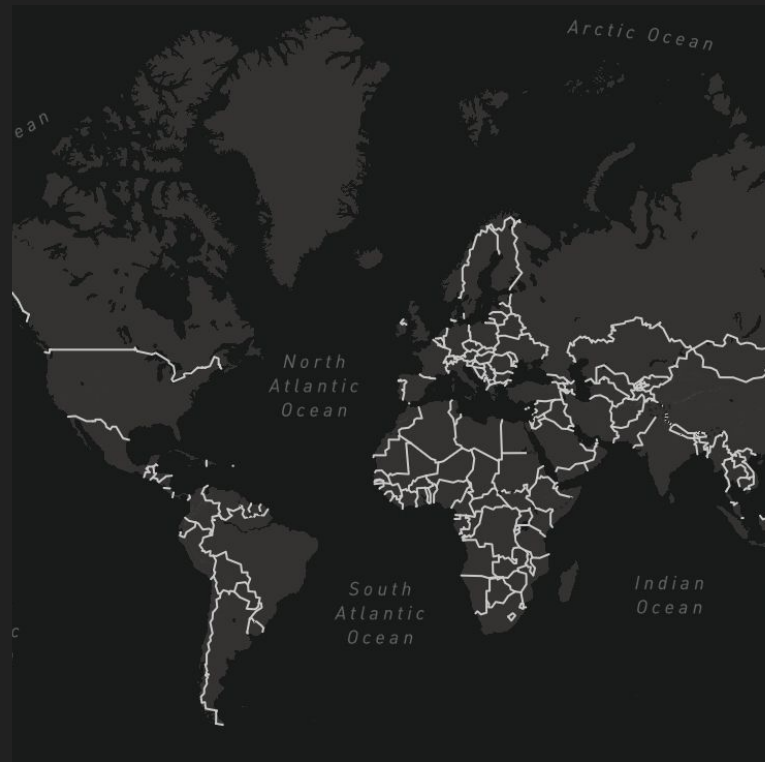
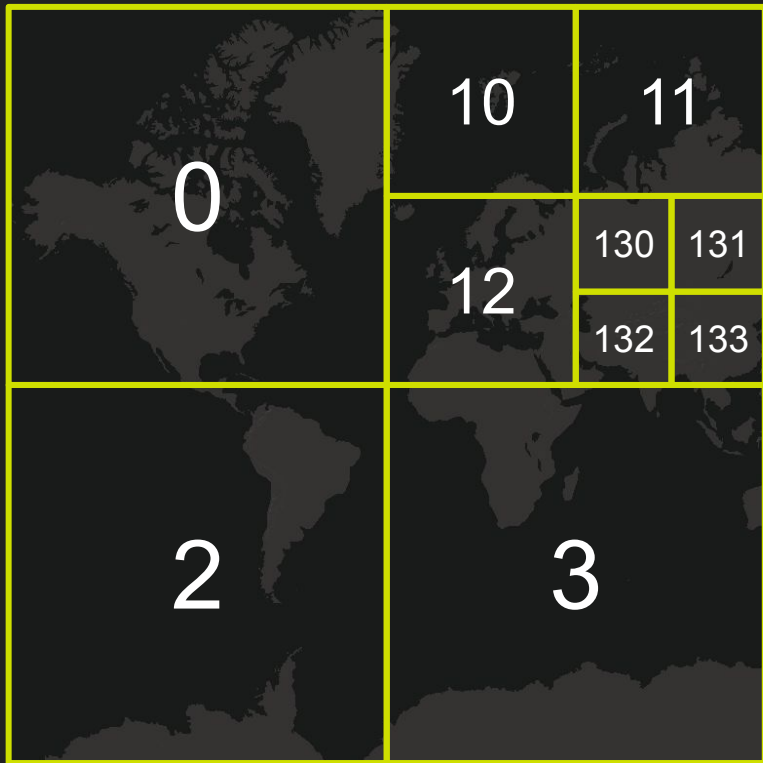
∞ точек



- Нет нагрузки на трафик
- Нет нагрузки на рендеринг
- Есть необходимость настройки хранения и получения данных

100 MB

Агрегация по позиции: GeoHash и Admin Area



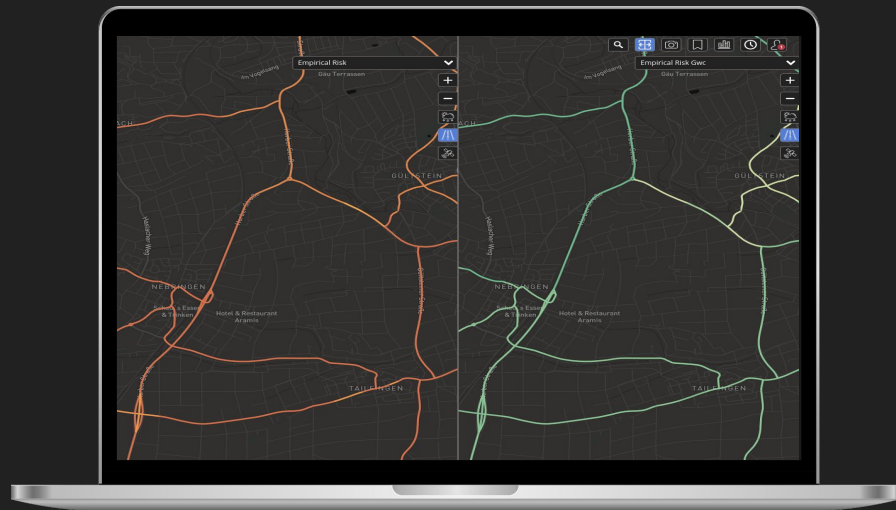
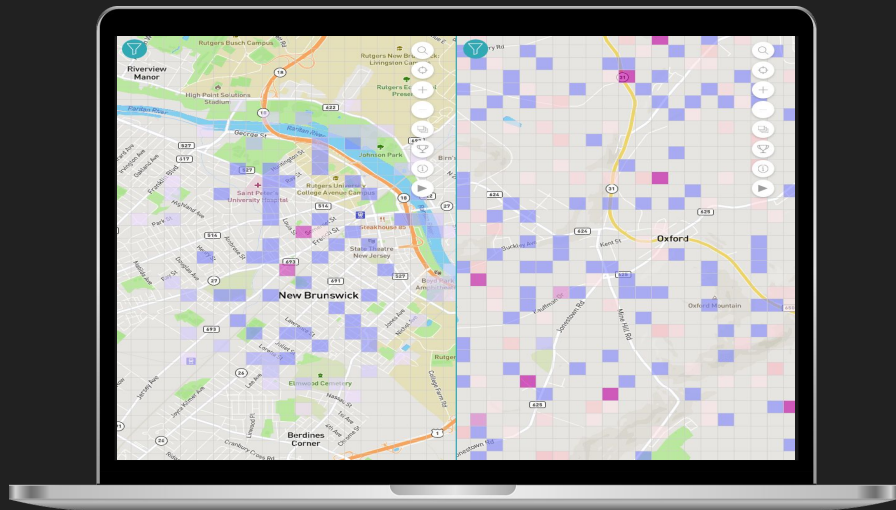


Интеграция карты с приложением

У вас есть карта? Дайте две!

2 недели разработки + регрессия

Добавили одну кнопку



Архитектура для множества карт на примере Vue

| -- modules

Разделяем модули глобального Store на общие и специфичные для каждой карты.

| -- report

| -- map

Специфичные модули динамически добавляем для каждой карты.

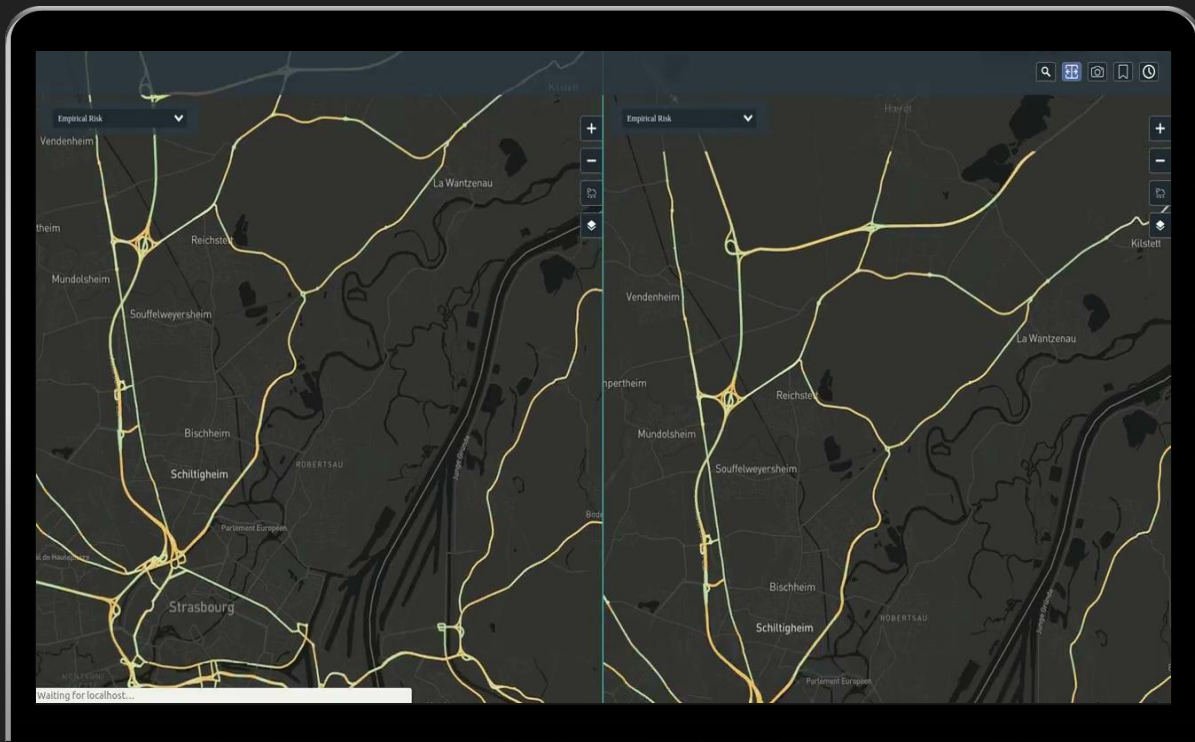
MapWrapper.vue

```
created() {  
  this.stateName = `${MAP_PREFIX}${this.id}`;  
  this.$store.registerModule(this.stateName, getState(this.id));  
},  
destroyed() {  
  this.$store.unregisterModule(this.stateName);  
},
```

Map.vue

```
<map-weather-layer  
  ...  
  :state-name="stateName"  
></map-weather-layer>
```

Тут мы немного расслабились...



“Мы хотим делиться прекрасными видами”

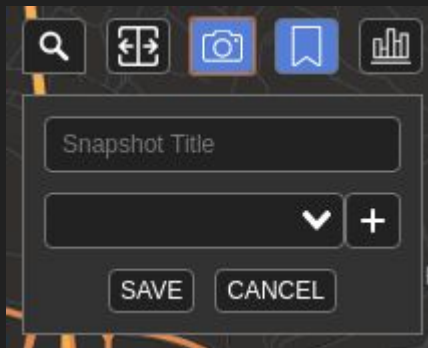
Через URL

?lng=7.84165
&lat=48.50844
&time=1583820000000
&zoom=11.05
&player=true
&time_interval_start=1583690400000
&time_interval_end=1584122399999
&map_left_weather=true
&map_left_segmentsTile=true
&...

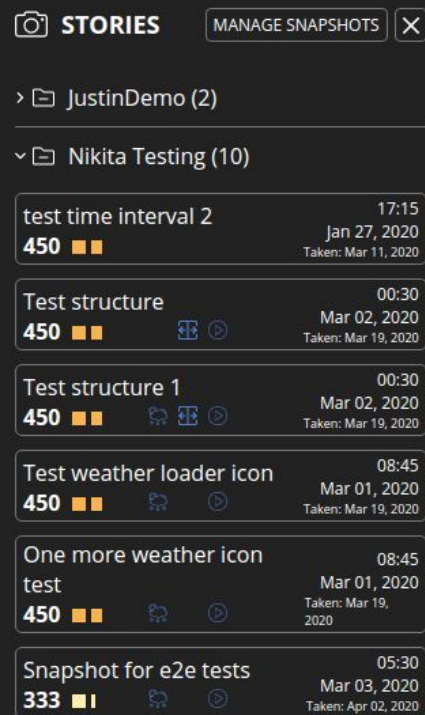
Будьте спокойны,
пока URL меньше

2 KB

Сохранять в базу



Брать из базы



Плохие и хорошие идеи

Раскидать код
обновления URL
по компонентам

Сделать абстрактный
компонент, который
будет следить за всем
сразу

Восстанавливать
состояние из URL
в компонентах

Восстанавливать
состояние полностью
при запуске
приложения

Сделать в базе
данных таблицу со
строгой структурой
для сохранения
снимков состояния

Сделать в базе
свободное JSON поле

Вишенка на торте - проигрыватель



Но не все сразу... Особенно загрузка данных

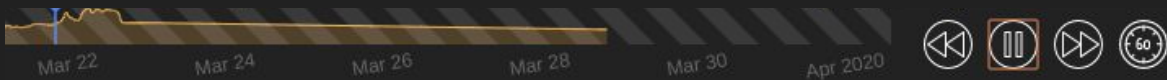
Это пользователь видит



А вот это пока не стоит...



Поэтому ждем



Логика проигрывания проста

```
this.timer = setInterval(  
    () => {  
        if (nextMoment) {  
  
            if (this.dataLoaded && shownFor > FRAME_DELAY) {  
                setTimeInUse(nextMoment);  
                setWaitingData(false); //скрываем полосу загрузки  
  
            } else if (shownFor > FRAME_DELAY) {  
                setWaitingData(true); //показываем полосу загрузки  
            }  
  
            } else {  
                resetTimer();  
                setPlaying(false);  
            }  
        },  
        300,  
    );
```

Добавляем новый слой, показываем только один

```
prepareLayers(type, preloadFrames) {  
  const layers = preloadFrames.map((time, index, frames) => {  
    const before = frames[index + 1] ?? currentLayer.key; // В начале позднее.  
    return {  
      key: `${type}-${time.getTime()}`,  
      before,  
      isNext: before === currentLayer.key,  
    };  
  });  
  
  ...  
  layers.push(currentLayer);  
  return layers;  
},  
-----  
map.setPaintProperty(  
  layerId,  
  'line-opacity-transition',  
  visible ? { duration: 0, delay: 0 } : { duration: 300, delay: 50 },  
);  
map.setPaintProperty(layerId, 'line-opacity', visible ? 1 : 0);
```

Доверяй, но проверяй

isSourceLoaded(id)

Returns a Boolean indicating whether the source is loaded.

... or hasn't been requested yet.

Следим за загрузкой данных для слоя

```
if (this.isVisible || this.isNext) { // Запись в состояние приложения
  setVisibleTilesForLayer({ id: layerId, visibleTiles });
}
```

```
map.on('sourcedata', updateLoadingStatus);
map.on('error', onError);
```

```
updateLoadingStatus(event) {
  if (!this.toBeRemoved && event.sourceId === layerId) {
    const { z, x, y } = event.coord.canonical;
    setTileStatusForLayer({ // Запись в состояние приложения
      tile: this.formatTileId(z, x, y),
      loaded: true,
      id: layerId,
    });
  }
}
```

```
onError(event) {
  ... // Все как при успешной загрузке
  setTileStatusForLayer({ error: true, ... });
},
```

А что если загрузить заранее?

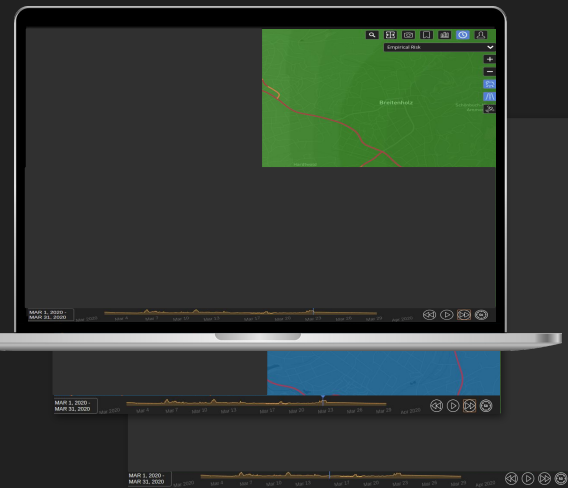
1 показываем



1 ждем



N ГОТОВИМ



Теперь проигрыватель должен работать плавно...

Где мои кадры, Хром? Ограничения HTTP1/1

Нужно кадров
для плавности

3

Нужно запросов
на 1 кадр

8..14

Итого запросов
на 3 кадра

..42

MAX запросов
на один домен

6

Запросов
в очереди

..36

Как включить HTTP2? Довольно просто.

На клиенте не нужно ничего. Браузер сам определит протокол.

На сервере (Nginx) нужно настроить SSL (обязательно для версии 2).

```
server {  
    listen 80; // перехватить http запрос  
    server_name my-site.com;  
    return 301 https://$host$request_uri; // перенаправить на https  
}  
server {  
    listen 443 ssl http2; // включить HTTP2  
    server_name my-site.com;  
    ...  
}
```

Что ждут от визуализации на карте?



Наглядность



Скорость



Аналитика



Сравнение



Сохранение



Проигрывание

Полезные ссылки

- [Mapbox GL JS](#)
- [Список библиотек для работы с Mapbox Vector Tiles](#)
- [Leaflet](#)
- [Leaflet.VectorGrid - плагин для рисования Vector Tiles](#)
- [Geohash - определение](#)

Вопросы?

Контакты:

nikitagsh@gmail.com

vk.com/shvyryaevng

Никита Швыряев