

Leaflet on steroids with Django

Valentino Gagliardi

Pycon IT 2022 - June 5, 2022





Valentino Gagliardi

Frontend developer

- JavaScript developer with a passion for tech writing
- Django lover
- I strongly believe in inclusion

20tab.com



20tab

Agenda

1 | **Challenges in rendering large amounts of GeoJSON features**

2 | **Backend optimizations**

3 | **Frontend optimizations**

The problem

The problem

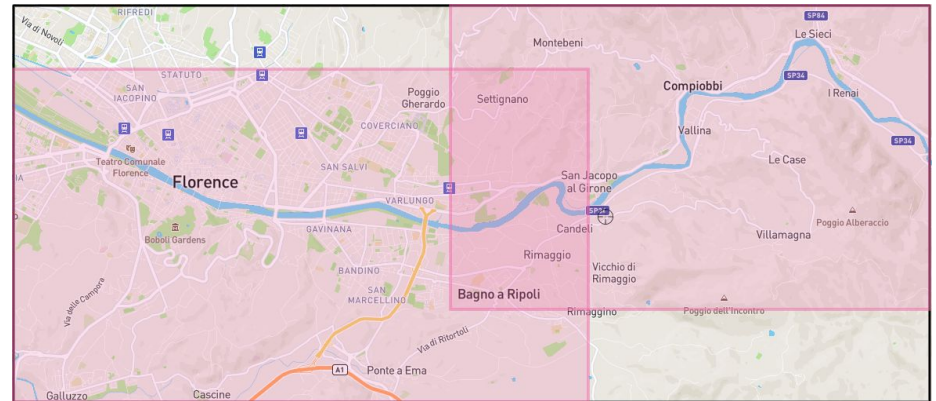
Challenges in rendering large amounts of spatial data.

A painfully slow map

Bounding box filter wasn't enough

Clustering wasn't an option

SW



NE

The solution

Backend optimizations

Backend optimizations

1 | Streaming response

2 | In geometry filter

Streaming response

It's streaming all the way down ...

```
class JSONLinesStreamingResponse(StreamingHttpResponse):  
  
    ...  
  
    def get_chunk_list(self, queryset, serializer_class, chunk_size):  
        qs = queryset.iterator(chunk_size=chunk_size)  
        while True:  
            items = list(islice(qs, chunk_size))  
            if not items:  
                break  
            yield "{}\n".format(  
                "\n".join(json.dumps(serializer_class(x).data) for x in items)  
            )
```

In geometry filter

When bounding box filter is not enough.

```
class InGeometryFilter(BaseFilterBackend):
    geometry_param = "in_geometry"
    ...

    def filter_queryset(self, request, queryset, view):
        ...
        geometry = self.get_geometry(request)
        if not geometry:
            return queryset
        return queryset.filter(**{f"{filter_field}__intersects": geometry})
```

Frontend optimizations

Frontend optimizations

1 | Streaming

2 | Diffing (with in geometry filter)

3 | Lazy loaded layers

Streaming

Harnessing the power of asynchronous iteration. (ES 2018)

```
const streamAsyncIterable = async function* (
  stream: ReadableStream<Uint8Array>
) {
  const reader = stream.getReader()

  try {
    while (true) {
      const { value, done } = await reader.read()
      if (done) return
      yield value
    }
  } finally {
    reader.releaseLock()
  }
}
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/function*

Streaming

Loading data into Leaflet chunk by chunk.

```
const addDataFromStream = async (body, parentLayer) => {
  const decoder = new TextDecoder()

  for await (const chunk of streamAsyncIterable(body)) {
    try {
      const features = decoder.decode(chunk)
      // Parse chunks, then add to layer
      parentLayer.addData({
        type: 'FeatureCollection',
        features
      })
    } catch (err) {
      console.error(err)
    }
  }
}
```

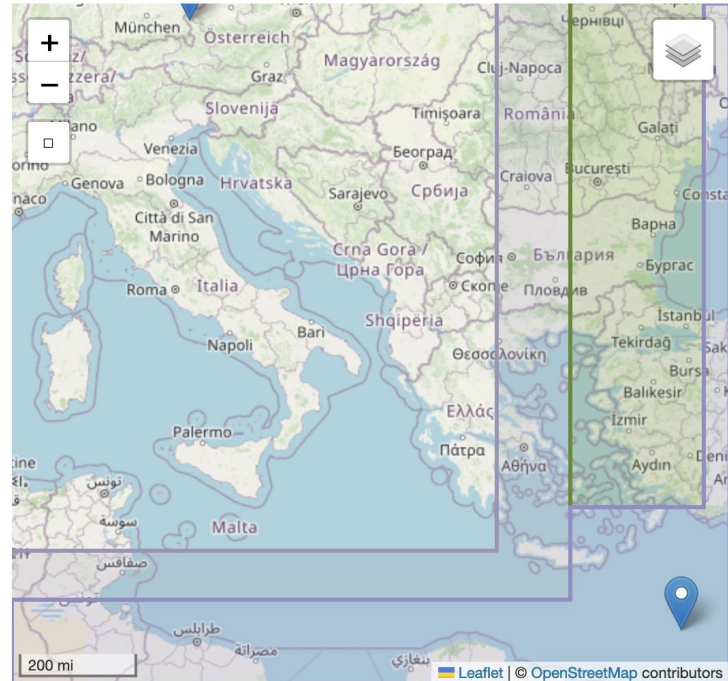
<https://jakearchibald.com/2017/async-iterators-and-generators/>



Diffing and lazy loaded layers

When the bounding box filter is not enough.

**A demo is better
than a thousand
words!**



Thank you



valentino@20tab.com



[valentinogagliardi](https://github.com/valentinogagliardi)



[valentinogagliardi](https://www.linkedin.com/in/valentinogagliardi)



[gagliardi_vale](https://twitter.com/gagliardi_vale)

20tab.com

