

Full-stack Django Rest Framework

Israel da Silva Teixeira

israel.teixeira@metr.systems

israelst@gmail.com

[@israelst](#)

Thiago Garcia da Silva

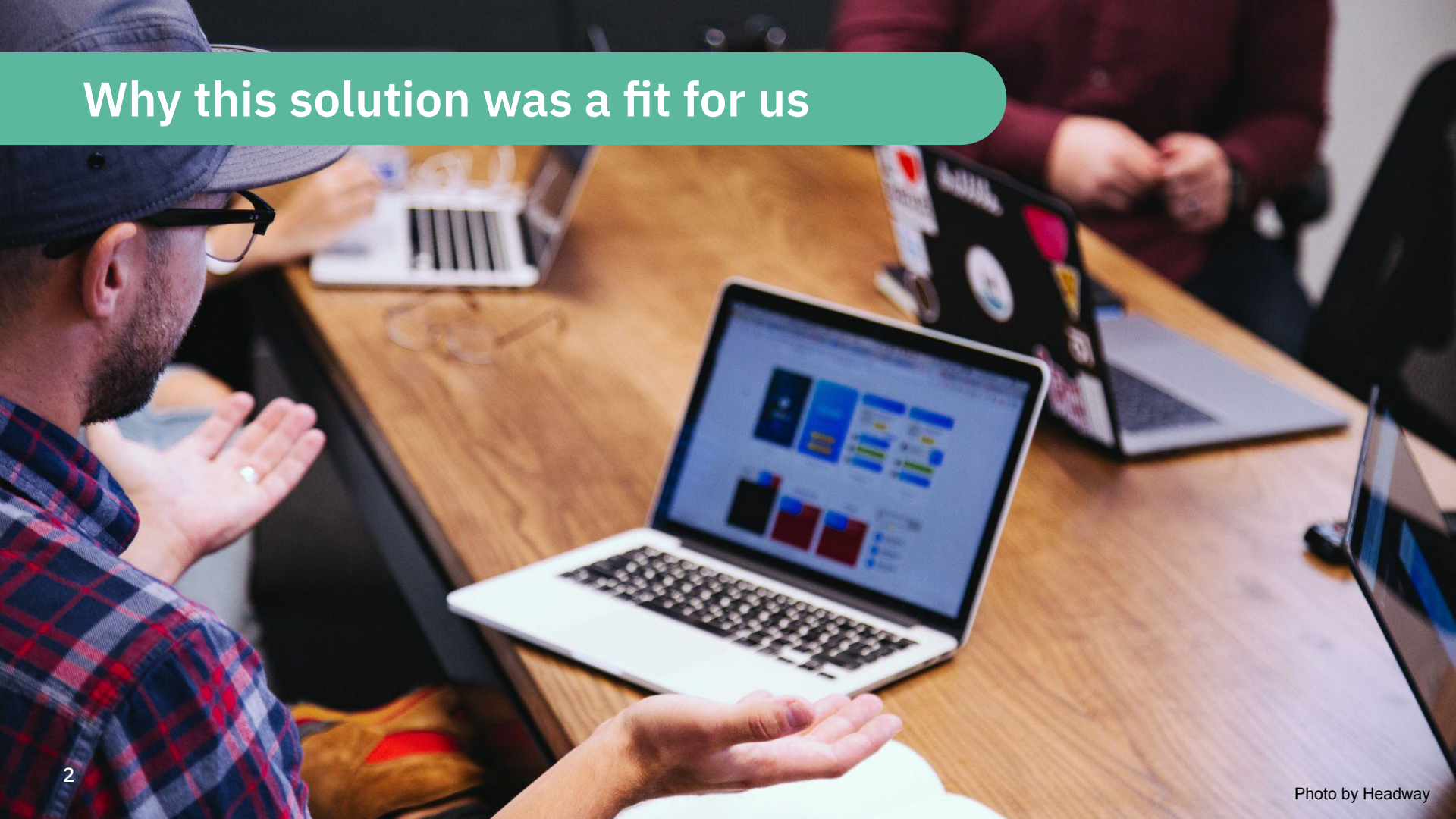
thiago.garciadasilva@metr.systems

hello@thiagogds.com

[@thiagogds](#)



Why this solution was a fit for us



The progressive enhancement way

- Start with an API
- Render Django Templates
- Add new renderers
- Enhance HTML with Javascript

Start with the API

```
class MyListView(generics.ListAPIView):  
    serializer_class = MySerializer
```

The base for the view is the same, we found easier to start with a `GenericAPIView` that can respond a JSON before doing the HTML Templates. Our intention is to be data first.

Render Django Templates

What is my context?

```
<span>{{ created_at |date:"D d M Y" }}</span>
```

Don't lose Python convenience

```
class DummySerializer:  
    def __init__(self, data, *args, **kwargs):  
        self.data = data
```

Rendering forms

templates/rest_framework/vertical/input.html

```
<div class="form-item {% if field.errors %}has-error{% endif %}">
  {% if field.label %}
    <label {% if style.hide_label %}class="sr-only"{% endif %}>
      {{ field.label }}</label>
    {% endif %}

    <input name="{{ field.name }}" {% if style.input_type != "file" %}class=
"form-control"{% endif %} type="{{ style.input_type }}" {% if style.placeholder
%}placeholder="{{ style.placeholder }}"{% endif %} {% if field.value is not
None %}value="{{ field.value }}"{% endif %} {% if style.autofocus and
style.input_type != "hidden" %}autofocus{% endif %}>

    {% if field.errors %}
      <ul class="errorlist">
        {% for error in field.errors %}
          <li>{{ error }}</li>
        {% endfor %}
      </ul>
    {% endif %}

    {% if field.help_text %}
      <span class="help-block">{{ field.help_text|safe }}</span>
    {% endif %}
</div>
```


Tricky parts ahead

Pagination

```
class StandardResultsSetPagination(pagination.LimitOffsetPagination):
    default_limit = 10
    max_limit = 100
    template = "device_manager/pagination.html"

    def get_html_context(self):
        context = {
            "count": self.count,
            "offset": self.offset,
            "limit": self.limit,
            "upper_offset": min(self.limit + self.offset, self.count),
            "previous_offset": max(self.offset - self.limit, 0),
            "last_offset": max(self.count - self.limit, 0),
            "previous_url": self.get_previous_link(),
            "next_url": self.get_next_link(),
        }
        return context
```

Pagination

```
def get_paginated_response(self, data):
    paginated_response = super().get_paginated_response(data)

    if self.is_html_format:
        paginated_response.data["paginator"] = self.paginator

    return paginated_response
```

Login required

```
def our_exception_handler(exc, context):
    request = context["request"]
    if request.accepted_media_type == "text/html":
        if isinstance(exc, exceptions.NotAuthenticated):
            next_url = request.get_full_path()
            return redirect_to_login(next=next_url)

    return exception_handler(exc, context)
```

Renderers

How they work

<https://www.django-rest-framework.org/api-guide/renderers/#third-party-packages>

```
from django.utils.encoding import smart_text
from rest_framework import renderers

class PlainTextRenderer(renderers.BaseRenderer):
    media_type = 'text/plain'
    format = 'txt'

    def render(self, data, media_type=None, renderer_context=None):
        return smart_text(data, encoding=self.charset)
```

How to change the default behaviour

```
def get_serializer_class(self):  
    if self.is_html_format:  
        return DummySerializer  
    elif self.is_xlsx_format:  
        return XLSXSerializer  
    else:  
        return super().get_serializer_class()
```

How to change the default behaviour

```
@property
def is_xlsx_format(self):
    accepted_renderer = self.request.accepted_renderer
    return type(accepted_renderer) == XLSXRenderer
```


How to change the default behaviour

```
def paginate_queryset(self, queryset):  
    if self.is_xlsx_format:  
        self.pagination_class = None  
    return super().paginate_queryset(queryset)
```

Enhance HTML with Javascript

localhost:8000/device-manager/?limit=10&offset=10

Network Console Inspector

Filter URLs | Disable Cache | No Th...

All HTML CSS JS XHR Fonts Images Media WS Other

St...	M...	Domain	File	I...	T...	T...	Siz
200	GET	localhost:8000	/device-manager/?limi...	Brov h...	3...		31.3

http://localhost:8000/device-manag

Waiting for localhost...

1 request | 31.32 KB / 32.11 KB transferred | Finish: 1.97 s | DOMContentLoaded

What the code looks like

```
<form action="{% url 'device_manager:device_error_update' row.id %}" method="post" class="ajax">
  {% csrf_token %}
  <input class="text" type="hidden" name="{{ name }}" value="{% if value %}false{% else %}true{% endif %}" >
  <input class="checkbox-submit" type="submit" value="" />
</form>
```

What the code looks like

```
function triggeringFormData(ev) {
  const form = ev.target
  const formData = new FormData(form)
  function requestLoad() {
    if (this.status == 200) {
      const response = JSON.parse(this.response)
      form.acknowledged.value = !response.acknowledged
      form.classList.toggle('checked', !response.acknowledged)
    }
  }
  const request = new XMLHttpRequest()
  request.addEventListener('load', requestLoad)
  request.open('POST', form.action + '?format=json', true)
  request.send(formData)
  ev.preventDefault()
}
```

What the code looks like

```
def update(self, request, *args, **kwargs):
    response = super().update(request, *args, **kwargs)
    if self.is_html_format:
        next_url = request.META.get("HTTP_REFERER", "device_manager:home")
        return redirect(next_url)
    else:
        return response
```



Conclusion

Find a solution that fits you.

Change just where and what you need.

Progressive enhancement.

Questions

Israel da Silva Teixeira

israel.teixeira@metr.systems

[@israelst](#)

Thiago Garcia da Silva

thiago.garciadasilva@metr.systems

[@thiagogds](#)

