

Django forms in the age of HTMx:

The single field form

Hanne Moa
DjangoCon EU 2026

Who am I?

- Hanne Moa. Some websites consider my name to be too short
- aka. hm, hmpf, harrumpf
- Developer @ Sikt (Norway's NREN, formerly known as Uninett)
- Open source all the way and All! the! Things! since the nineties
- hanne.moa@sikt.no, working hours, central European time
- No *professional* presence in social media, by choice

Overview

- Horrors of the past
- Introduction to single field forms
- Demo
- The magic
- Bonus x 2, if time permits (more magic!)
- Links galore
- That mandated slide with all the logos

Form

[Form , Form , Form]

Form

[

Form

,

Form

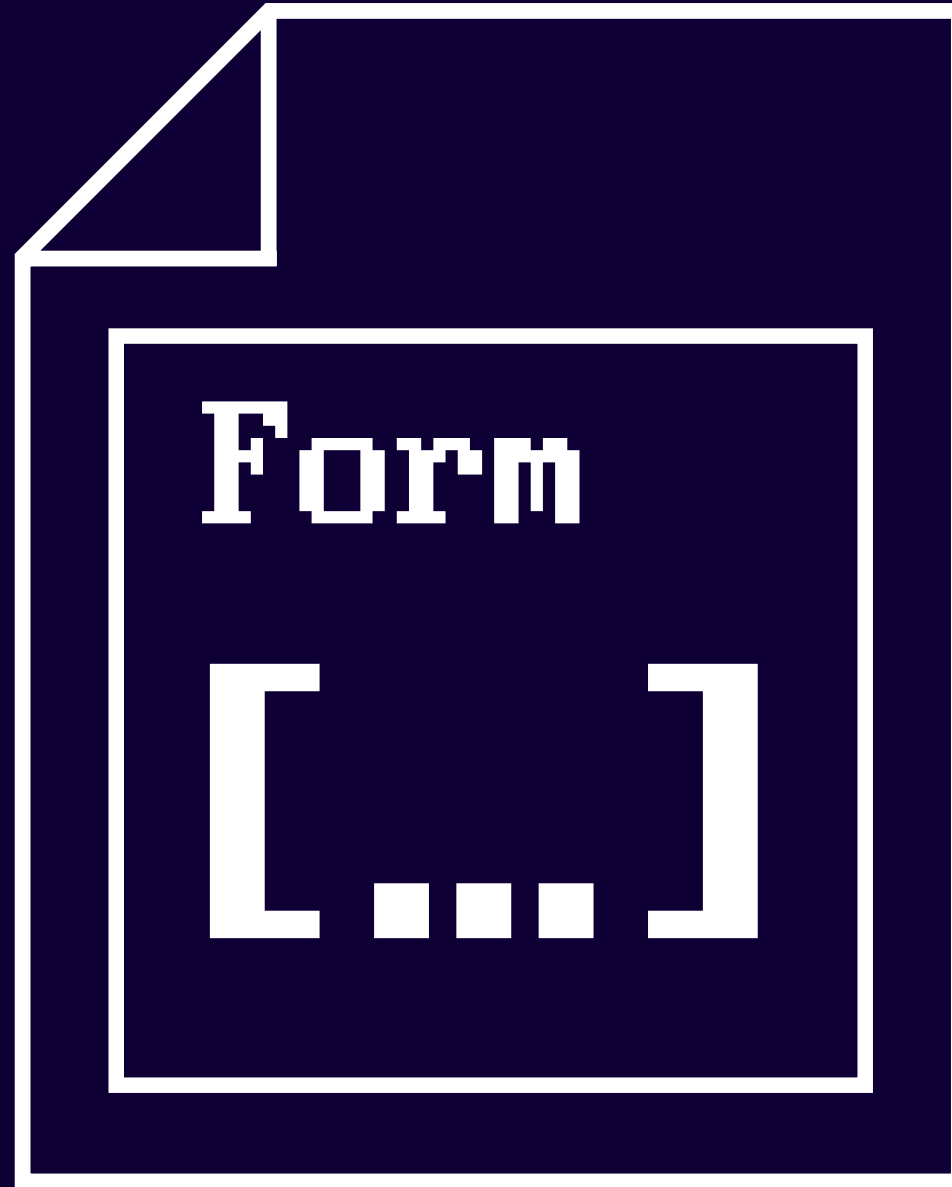
,

Form

]

Form

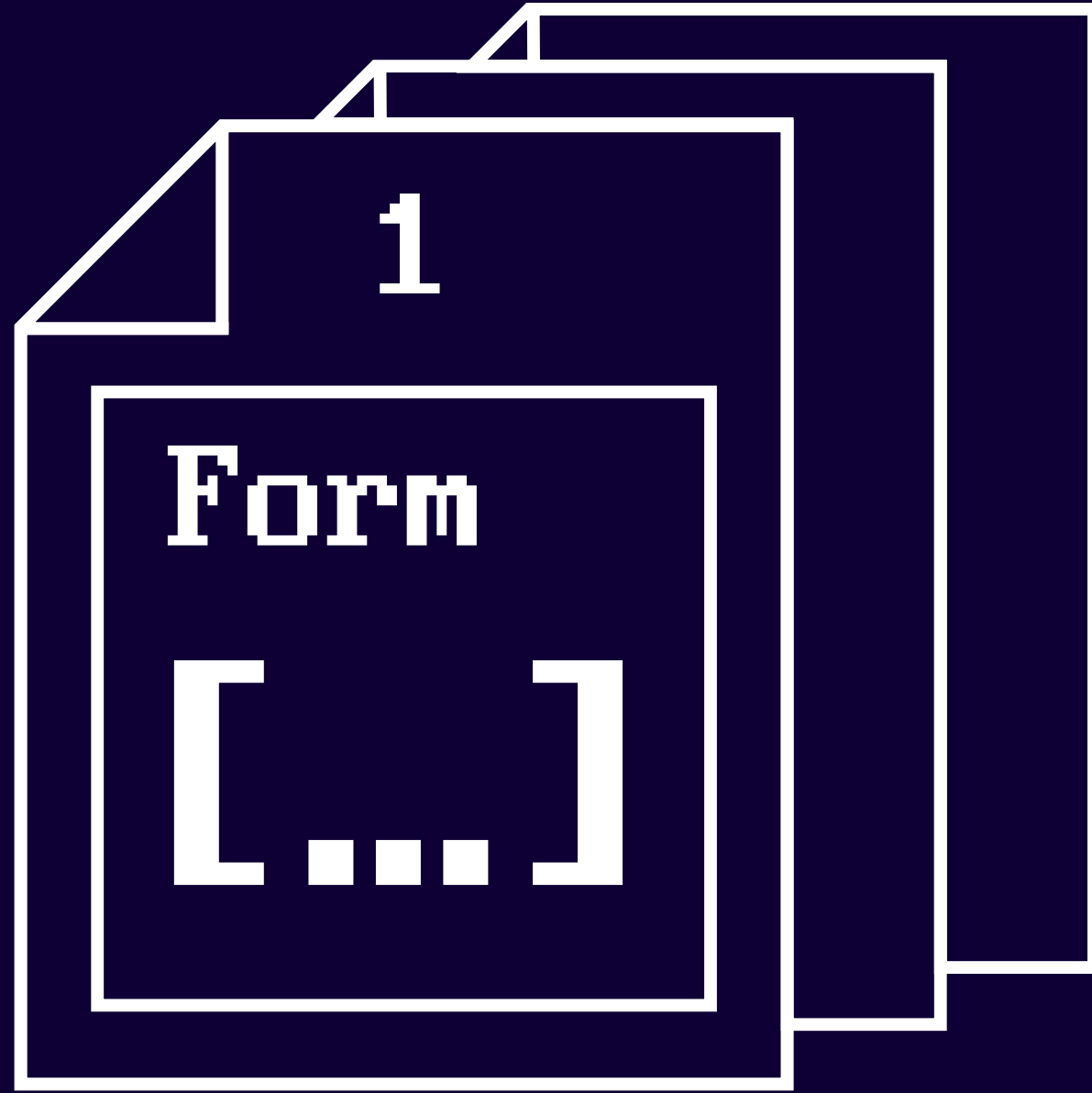
[...]

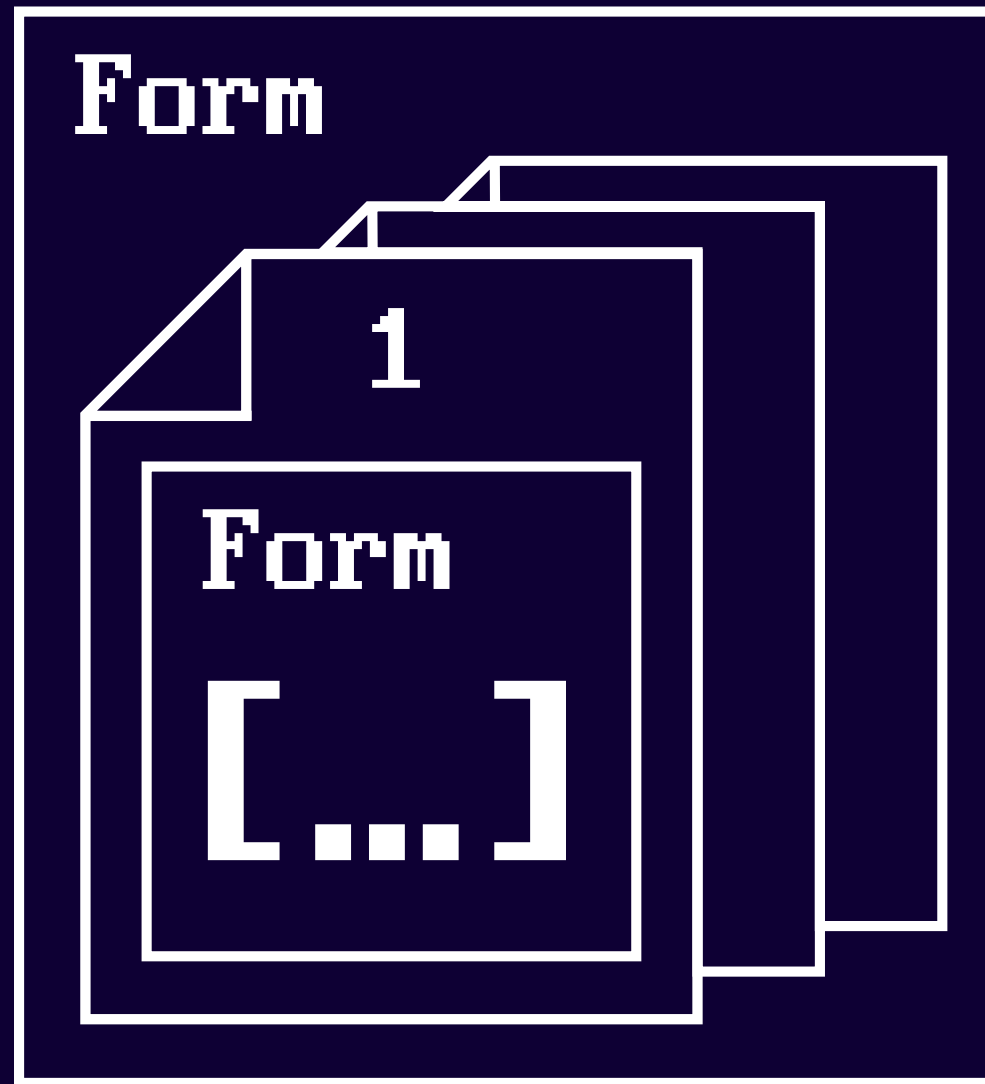


1

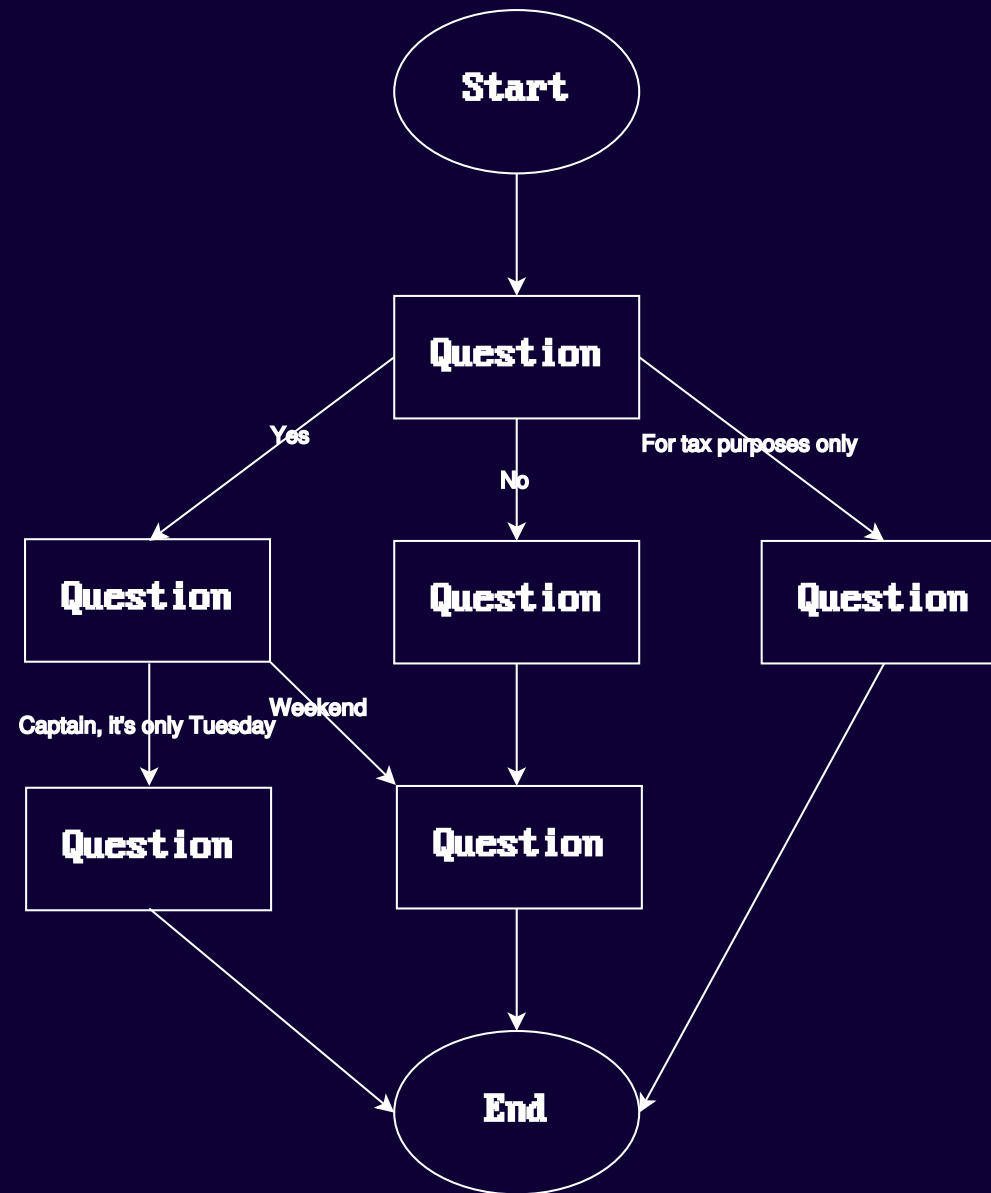
Form

[...]









The single field form: a pattern

Example model

```
class Book(models.Model):  
    title = models.CharField(max_length=255)  
    author = models.CharField(max_length=255)  
    misc = models.JSONField(default=dict, blank=True)
```

Forms and views

```
class SingleFieldFormMixin:
    fieldname: str

    def get_field(self):
        return self[self.fieldname]

class BookTitleForm(SingleFieldFormMixin, forms.ModelForm):
    fieldname = "title"

    class Meta:
        model = Book
        fields = ['title']

def edit_view(request):
    """
    if request.POST:
        forms = get_forms(data=request.POST, obj=object)
    """

def get_field_template_view(request, fieldname):
    """
    forms = get_forms(obj=object)
    form = forms[fieldname]
    context = super().get_context_data(fieldname=fieldname, form=form, **kwargs)
    return render(request, '_form_for_field.html', context=context)
```

Templates

```
<!-- part of detail template -->
<h2>
  {{ object.title }}
  <button hx-get="{% url 'get_field' pk=object.pk fieldname='title' %}"
    hx-target="closest h2">Edit</button>
</h2>

<!-- _form_for_field.html -->
<form class="singlefield"
      method="post"
      action="{% url 'edit' pk=object.pk fieldname=fieldname %}">
  {{ form.get_field }}
  <input type="submit" value="Save">
</form>
```

DEMO

If you want to follow along:

singlefieldform @ Github: <https://github.com/Uninett/singlefieldform>

Argus Demo @ Sikt: <http://argus-demo.uninett.no/>

The magic

```
def get_forms(data=None, obj=None):
    data = {} if data is None else data
    forms = {}
    for Form in SingleFieldFormMixin.__subclasses__():
        kwargs = {}
        kwargs['instance'] = obj
        if Form.fieldname in data:
            kwargs['data'] = data
        form = Form(**kwargs)
        forms[form.fieldname] = form
    return forms
```

Bonus #1: more context in widget templates

```
class HxWidgetMixin:
    def __init__(self, hx: Optional[dict] = None, **kwargs):
        super().__init__(**kwargs)
        self.hx = {} if hx is None else hx

    def __deepcopy__(self, memo):
        obj = super().__deepcopy__(memo)
        obj.hx = self.hx.copy()
        memo[id(self)] = obj
        return obj

    def get_context(self, name, value, attrs):
        context = super().get_context(name, value, attrs)
        context["widget"]["hx"] = self.extra
        return context
```

Bonus #2: better pluginable fields

```
class BetterPluginableSingleFormField(SingleFormFieldMixin):
    registry: dict = dict()

    @classmethod
    def register(cls):
        cls.registry[cls.fieldname] = cls

    @classmethod
    def unregister(cls):
        cls.registry.pop(cls.fieldname, None)

    def __init_subclass__(cls, **kwargs):
        cls.register()
```

Argus

Homepage @ Géant	https://wiki.geant.org/spaces/NETDEV/pages/417792154/Argus
Docs	https://argus-server.readthedocs.io
PyPI	argus-server
Repo, issues, discussions	https://github.com/Uninett/Argus



The singlefieldform demo

Repo <https://github.com/Uninett/singlefieldform>

This presentation

Slides	Single file: HTML + CSS (mostly flexbox), <u>vim classic</u> (thanks, Drew Devault)
Illustrations	Inline SVG (via inkscape), PNG as data-uri

The End

Funded/backed by



Hanne Moa – hanne.moa@sikt.no

