

Building complex GUIs (without tearing your hair out)

Anders Hovmöller - github.com/boxed

TriOptima - github.com/trioptima

The problem with Django forms

```
def django_example1(request):
    class CreateRoomForm(forms.Form):
        name = forms.CharField()
        description = forms.CharField(widget=forms.Textarea)

    if request.method == 'POST':
        form = CreateRoomForm(request.POST)
        if form.is_valid():
            Room.objects.create(
                name=form.cleaned_data['name'],
                description=form.cleaned_data['description']
            )
            return HttpResponseRedirect('/something/')

    else:
        form = CreateRoomForm()

    return render(request, 'forum/room_form.html', {'form': form})
```

```
class DjangoExample1B(CreateView):  
    model = Room  
    fields = ['name', 'description']  
    # implicit template_name = 'forum/room_form.html'
```

```
class DjangoExample2(CreateView):  
    model = Room  
    fields = ['name', 'description', 'auditor_notes']  
  
    def get_form(self, form_class=None):  
        form = super().get_form(form_class=form_class)  
        if not self.request.user.is_staff:  
            del form.fields['auditor_notes']  
        return form
```

```
class DjangoExample3(UpdateView):
    model = Room
    fields = ['name', 'description', 'auditor_notes']

    def get_form(self, form_class=None):
        form = super().get_form(form_class=form_class)
        if not self.request.user.is_staff:
            del form.fields['auditor_notes']
        else:
            form.fields['audit_complete'] = \
                forms.BooleanField(required=False)

    return form
```

```
def form_valid(self, form):
    response = super().form_valid(form)
    if self.request.user.is_staff and \
        form.cleaned_data['audit_complete']:
        self.object.last_audit = datetime.now()
        self.object.auditor = self.request.user
        self.object.save()
    return response
```

```

class DjangoExample4(UpdateView):
    [...snip...]

    def get_form(self, form_class=None):
        form = super().get_form(form_class=form_class)

        form.fields['auditor_notes'].disabled = not self.request.user.contact.is_auditor

        if not self.request.user.contact.is_auditor and not self.request.user.is_staff:
            del form.fields['auditor_notes']
        else:
            if self.request.user.contact.is_auditor:
                form.fields['audit_complete'] = forms.BooleanField(required=False)
            if self.request.user.is_staff:
                form.fields['auditor'] = forms.ModelChoiceField(
                    disabled=True,
                    queryset=User.objects.all(),
                    initial=self.object.auditor,
                )
                form.fields['last_audit'] = forms.DateTimeField(
                    initial=self.object.last_audit,
                    disabled=True,
                )

        return form

    def form_valid(self, form):
        response = super().form_valid(form)
        if self.request.user.contact.is_auditor and \
            form.cleaned_data.get('audit_complete'):
            self.object.last_audit = datetime.now()
            self.object.auditor = self.request.user
            self.object.save()
        return response

```

```

<form method="post">
  {% csrf_token %}
  <table>
    {# I can't loop over the BoundField items :( #}

    {% include "field.html" with field=form.name %}
    {% include "field.html" with field=form.description %}

    {% if user.contact.is_auditor or user.is_staff %}
      <tr>
        <th colspan="2">Audit</th>
      </tr>
    {% endif %}

    {% include "field.html"
      with field=form.last_audit extra_css_class="audit" %}

    {% include "field.html"
      with field=form.auditor extra_css_class="audit" %}

    {% include "field.html"
      with field=form.auditor_notes extra_css_class="audit" %}

    {% include "field.html"
      with field=form.audit_complete extra_css_class="audit" %}
  </table>
  <input type="submit" value="Submit">
</form>

```



```

<tr{% if field.css_classes or extra_css_class %}
class="{{ field.css_classes|join:" " }}"
{{ extra_css_class }}"{% endif %}>
    <th>
<label for="{{ field.id }}">
    {{ field.label }}
</label>
</th>
<td>
    {% if field.errors %}
        <ul class="errorlist">
            {% for error in field.errors %}
                <li class="error">
                    {{ error }}
                </li>
            {% endfor %}
        </ul>
    {% endif %}
    {{ field }}
</td>
</tr>

```

tri.form

github.com/trioptima/tri.form

```
def django_example1(request):
    class CreateRoomForm(forms.Form):
        name = forms.CharField()
        description = forms.CharField(widget=forms.Textarea)

    if request.method == 'POST':
        form = CreateRoomForm(request.POST)
        if form.is_valid():
            Room.objects.create(
                name=form.cleaned_data['name'],
                description=form.cleaned_data['description']
            )
            return HttpResponseRedirect('/something/')

    else:
        form = CreateRoomForm()

    return render(request, 'forum/room_form.html', {'form': form})
```

```
def tri_form_example1(request):
    class CreateRoomForm(Form):
        name = Field()
        description = Field.textarea()

    if request.method == 'POST':
        form = CreateRoomForm(request=request)
        if form.is_valid():
            Room.objects.create(
                name=form.fields_by_name.name.value,
                description=form.fields_by_name.description.value,
            )
            return HttpResponseRedirect('/something/')

    else:
        form = CreateRoomForm()

    return render(request, 'forum/room_form.html', {'form': form})
```

```
class DjangoExample1B(CreateView):  
    model = Room  
    fields = ['name', 'description']
```

```
def tri_form_example1_b(request):  
    return create_object(  
        request=request,  
        model=Room,  
        form__include=['name', 'description'],  
    )
```

```
class DjangoExample2(CreateView):  
    model = Room  
    fields = ['name', 'description', 'auditor_notes']  
  
    def get_form(self, form_class=None):  
        form = super().get_form(form_class=form_class)  
        if not self.request.user.is_staff:  
            del form.fields['auditor_notes']  
        return form
```

```
def tri_form_example2(request):  
    return create_object(  
        request=request,  
        model=Room,  
        form__include=['name', 'description', 'auditor_notes'],  
        form__field__auditor_notes__show=request.user.is_staff,  
    )
```

```
class DjangoExample3(UpdateView):
    model = Room
    fields = ['name', 'description', 'auditor_notes']

    def get_form(self, form_class=None):
        form = super().get_form(form_class=form_class)
        if not self.request.user.is_staff:
            del form.fields['auditor_notes']
        else:
            form.fields['audit_complete'] = \
                forms.BooleanField(required=False)

    return form
```

```
def form_valid(self, form):
    response = super().form_valid(form)
    if self.request.user.is_staff and \
        form.cleaned_data['audit_complete']:
        self.object.last_audit = datetime.now()
        self.object.auditor = self.request.user
        self.object.save()
    return response
```

```

def tri_form_example3(request, pk):
    def on_save(instance, form, **_):
        if request.user.is_staff and \
            form.fields_by_name.audit_complete.value:
            instance.last_audit = datetime.now()
            instance.auditor = request.user
            instance.save()

    return edit_object(
        request=request,
        instance=Room.objects.get(pk=pk),
        on_save=on_save,
        form__exclude=['auditor', 'last_audit'],
        form__extra_fields=[
            Field.boolean(
                name='audit_complete',
                # don't write "audit_complete" to the Room object
                attr=None,
                show=request.user.is_staff,
            ),
        ],
        form__field__auditor_notes__show=request.user.is_staff,
    )

```



```

return edit_object(
    request=request,
    instance=Room.objects.get(pk=pk),
    on_save=on_save,
    form__extra_fields=[
        Field.boolean(
            name='audit_complete',
            attr=None, # don't write "audit_complete" to the Room object
            show=request.user.contact.is_auditor,
        ),
        Field.heading(
            name='audit',
            after='description',
            show=request.user.contact.is_auditor or request.user.is_staff,
        ),
    ],
    form__field=dict(
        auditor_notes__show=request.user.is_staff or request.user.contact.is_auditor,
        auditor_notes__editable=request.user.contact.is_auditor,

        auditor__editable=False,
        auditor__show=request.user.is_staff,

        last_audit__editable=False,
        last_audit__show=request.user.is_staff,

        last_audit__container__attrs__class__audit=True,
        auditor__container__attrs__class__audit=True,
        auditor_notes__container__attrs__class__audit=True,
    ),
)

```

Programmatically
creating forms

```
class CreateRoomForm(forms.Form):  
    name = forms.CharField()  
    description = forms.CharField(widget=forms.Textarea)  
  
form = CreateRoomForm()
```

```
CreateRoomForm = type(
    'CreateRoomForm',
    (forms.Form,),
    dict(
        name=forms.CharField(),
        description = forms.CharField(widget=forms.Textarea),
    )
)

form = CreateRoomForm()
```

```
form = Form(  
    fields=[  
        Field(name='name'),  
        Field.textarea(name='description'),  
    ]  
)
```

```
def create_form(**kwargs):  
    return Form(  
        fields=[  
            Field(name='name'),  
            Field.textarea(name='description'),  
        ],  
        **kwargs  
    )
```

```
form = create_form(  
    field__description__label_container__attrs__class__foo=True  
)
```

tri.query

github.com/trioptima/tri.query

```
class MyQuery(Query):  
    name = Variable(gui__show=True)  
    description = Variable()
```


tri.table

github.com/trioptima/tri.table










Active

Administrator

[Switch to advanced search](#)

	Username	Name	Active	Last login
 Edit	abc	abc test	✓	1y, 1m ago
 Edit	abidi	Abid Ikram	✓	9y, 2m ago
 Edit	adeled	Adele Di Sarno	✓	1mo ago
 Edit	admin34	Admin Test 2		1y, 9m ago
 Edit	adminhiroshi	admin hiroshi		Never
 Edit	ahmati	Zabit Ahmati	✓	5y, 3m ago
 Edit	alexb	Alex Boecking		5y, 2m ago
 Edit	carlottac	Carlotta Carbon	✓	4y, 1m ago
 Edit	case	abc_case	✓	2y, 2m ago

First < **1** 2 3 4 5 6 > Last

```
class MyTable(Table):  
    name = Column()  
    description = Column()  
  
return render_table_to_response(  
    request,  
    table=MyTable(data=Room.objects.all().order_by('name'))  
)
```

```
return render_table_to_response(  
    request,  
    table__model=Room,  
    table__include=['name', 'description'],  
)
```

```
return render_table_to_response(  
    request,  
    table__model=Room,  
    table__column__name__query__show=True,  
    table__column__name__query__gui__show=True,  
    table__column__description__query__show=True,  
)
```

table__column__description__bulk__show=True

Laziness:
later is better than right now

```
class CreateMessage(Form):  
    url = Field.url()  
    text = Field.textfield()
```



```
def create_messages(...):  
    class CreateMessage(Form):  
        if room.has_urls:  
            url = UrlField()  
            text = CharField()  
    ....
```

```
class CreateMessage(Form):  
    url = Field.url(  
        show=lambda form, **_: form.extra.room.has_urls  
    )  
    text = Field.textfield()
```

```
class CreateMessage(Form):  
    url = Field.url(  
        show=lambda form, **_: form.extra.room.has_urls  
    )  
    text = Field.textfield(  
        container__attrs__class__admin= \  
            lambda form, **_: form.extra.room.is_admin_room  
    )
```

```
>>> greeting = evaluate('Hi', name='Anders')  
'Hi'
```

```
>>> evaluate(lambda name, **_: f'hi {name}',  
...          name='Anders', species='Human')  
'Hi Anders'
```

```
MenuItem(title='Manage users', url='/users/'),  
...  
MenuItem(  
    title='Files',  
    url=lambda contact: '/dashboard/files/%s' % '? \'  
        status=Error' if contact.user.is_staff else ''),
```

Escape hatch: extra

```
form = CreateMessage(request, extra__room=room)
```

```
Column(extra__report_name='foo')
```


Everything has a name

```
# models.py
```

```
class IssueType(Model):  
    name = CharField(max_length=255)
```

```
class Issue(Model):  
    issue_type = ForeignKey(IssueType)  
    name = CharField(max_length=255)  
    description = TextField()
```

```
# views.py
```

```
def create_issue(request):  
    return create_object(  
        request,  
        model=Issue,  
    )
```

Time to breathe!










Active

Administrator

[Switch to advanced search](#)

	Username	Name	Active	Last login
 Edit	abc	abc test	✓	1y, 1m ago
 Edit	abidi	Abid Ikram	✓	9y, 2m ago
 Edit	adeled	Adele Di Sarno	✓	1mo ago
 Edit	admin34	Admin Test 2		1y, 9m ago
 Edit	adminhiroshi	admin hiroshi		Never
 Edit	ahmati	Zabit Ahmati	✓	5y, 3m ago
 Edit	alexb	Alex Boecking		5y, 2m ago
 Edit	carlottac	Carlotta Carbon	✓	4y, 1m ago
 Edit	case	abc_case	✓	2y, 2m ago

First



1

2

3

4

5

6



Last

table

label

gui

choices

Administrator

query

Switch to advanced search

	header	Name	Active	Last login
Edit	abc	abc test	✓	1y, 1m ago
Edit	abidi	Abid Ikram	✓	9y, 2m ago
Edit	adeled	Adele Di Sarno	✓	1mo ago
Edit	admin34	Admin Test 2		1y, 9m ago
Edit	column	admin hiroshi		Never
Edit	ahmati	Zabit Ahmati	✓	5y, 3m ago
Edit	alexb	Alex Boecking		5y, 2m ago
Edit	carlottac	Carlotta Carbon	row	4y, 1m ago
Edit	case	abc_case	✓	2y, 2m ago

First

<

1

paginator

6

>

Last

table

label

choices

Administrator

query

Switch to advanced search

	header	Name	Active	Last login
Edit	abc	abc test	✓	1y, 1m ago
Edit	abidi	Abid Ikram	✓	9y, 2m ago
Edit	adeled	Adele Di Sarno	✓	1mo ago
Edit	admin34	Admin Test 2		1y, 9m ago
Edit	column	admin hiroshi		Never
Edit	ahmati	Zabil		
Edit	alexb	Alex Boecking		5y, 2m ago
Edit	carlottac	Carlotta Carbon	row	4y, 1m ago
Edit	case	abc_case	✓	2y, 2m ago

First < 1 paginator 6 > Last

@dispatch and Namespace

```
import feedparser  
import requests
```

```
def get_feed(url):  
    response = requests.get(url=url)  
    return feedparser.parse(response.data)
```



```
import feedparser
import requests
```

```
def get_feed(url, auth=None):
    response = requests.get(
        url=url,
        auth=auth,
    )
    return feedparser.parse(response.data)
```

```
import feedparser
import requests
```

```
def get_feed(url, auth=None, encoding=None):
    response = requests.get(
        url=url,
        auth=auth,
    )
    if encoding is not None:
        response.encoding = encoding
    return feedparser.parse(response.data)
```

```
import feedparser
import requests
```

```
def get_feed(url, auth=None, encoding=None,
             replacement=None):
    response = requests.get(
        url=url,
        auth=auth,
    )
    if encoding is not None:
        response.encoding = encoding
    d = response.data
    if replacement is not None:
        d = d.replace(
            replacement[0],
            replacement[1])
    return feedparser.parse(d)
```

```
import feedparser
import requests
```

```
@dispatch(
    fetch__call_target=requests.request,
    fetch__method='get',
    decode=lambda response: response.data,
)
def get_feed(url, fetch, decode):
    response = fetch(url=url)
    data = decode(response)
    return feedparser.parse(data)
```

```
get_feed('https://foo.com')
```

```
get_feed(  
    'https://private.foo.com',  
    fetch__auth=('username', 'password'))
```

```
get_feed(  
    'https://broken.foo.com',  
    decode=lambda response: \  
        response.data.replace('foo"', 'foo'))
```

Summary

Thanks for listening!

Find our libs at:
github.com/TriOptima

Anders Hovmöller
github.com/boxed