DynamicForms Documentation

Release 0.9 dev

Velis Ltd

Jan 12, 2022

Contents:

CHAPTER 1

What is DynamicForms?

DynamicForms performs all the visualisation & data entry of your DRF Serializers & ViewSets and adds some candy of its own: It is a django library that gives you the power of dynamically-shown form fields, auto-filled default values, dynamic record loading and similar candy with little effort. To put it differently: once defined, a particular ViewSet / Serializer can be rendered in multiple ways allowing you to perform viewing and authoring operations on the data in question.

It is based on django-rest-framework

1.1 Why DynamicForms

- Turn your rest-framework ViewSets into HTML forms
- Powerful HTML based CRUD
  - Support for fetching "new" records, both in JSON or in HTML
  - Render to HTML, dialog html or from your own template
  - Render form (embedded or dialog) or table, depending on situation
  - Easily add actions and place the buttons to execute them anywhere you like
- Clear separation of list & dialog templates
- Dynamic loading of additional records for table views
- Easy implementation of simple filtering
- Action items, declared globally, placed where you need them
- Custom templates whenever & wherever you want them
- Render to full html or work with dialogs within same page or both at the same time
- Each form and field have a unique HTML id for easy finding & manipulation
- Bootstrap 3 & 4 templates, jQuery UI coming soon, easy to make your own or enhance existing

- Support for form validation, will show errors even if they are not tied to a field
- Convenient JS functions for easier action scripting
- Progress dialog for long lasting ajax operations

CHAPTER 2

Using DynamicForms

2.1 Quick start guide

```
pip install dynamicforms
```

Then you need to add following setting to your project's settings.py .. code-block:: python

caption settings.py

name settings.py

REST_FRAMEWORK = {

'DEFAULT_RENDERER_CLASSES': ('rest_framework.renderers.JSONRenderer', 'rest_framework.renderers.BrowsableAPIRenderer', 'dynamicforms.renderers.TemplateHTMLRenderer',

)

}

DynamicForms has been designed to cause minimal disruption to your existing code patterns.

So instead of DRF ModelViewSet just use DynamicForms ModelViewSet, instead of ModelSerializer - DynamicForms ModelSerializer.

Currently only the *ModelViewSet* is supported for ViewSets. We have others planned, but not implemented yet.

```
:caption: examples/rest/page_load.py
:name: examples/rest/page_load.py

from dynamicforms import serializers, viewsets
from ..models import PageLoad


class PageLoadSerializer(serializers.ModelSerializer):
    form_titles = {
```

(continues on next page)

(continued from previous page)

```
        'table': 'Dynamic page loader list',
        'new': 'New object',
        'edit': 'Editing object',
    }

    class Meta:
        model = PageLoad
        exclude = ()


class PageLoadViewSet(viewsets.ModelViewSet):
    template_context = dict(url_reverse='page-load')
    pagination_class = viewsets.ModelViewSet.generate_paged_loader(30)  # enables␣
↪pagination

    queryset = PageLoad.objects.all()
    serializer_class = PageLoadSerializer
```

Listing 1: examples/models.py (excerpt)

```
from django.db import models

class PageLoad(models.Model):
    """
    Shows how DynamicForms handles dynamic loading of many records in ViewSet result
    """
    description = models.CharField(max_length=20, help_text='Item description')
```

If you want filter in list view just set serializers property show_filter value to True. Filter will be applied if user press enter in filter field. If you want to have filter button in list header, call Actions with add_default_filter = True.

Listing 2: examples/filter.py

```
from dynamicforms import serializers, viewsets
from dynamicforms.action import Actions
from ..models import Filter


class FilterSerializer(serializers.ModelSerializer):
    form_titles = {
        'table': 'Dynamic filter list',
        'new': 'New object',
        'edit': 'Editing object',
    }
    actions = Actions(add_default_crud=True, add_default_filter=True)
    show_filter = True

    class Meta:
        model = Filter
        exclude = ()


class FilterViewSet(viewsets.ModelViewSet):
    template_context = dict(url_reverse='filter')
    pagination_class = viewsets.ModelViewSet.generate_paged_loader(30)  # enables␣
↪pagination
```

(continues on next page)

(continued from previous page)

```
    queryset = Filter.objects.all()
    serializer_class = FilterSerializer
```

2.1.1 Custom page template

Following is an example page template to render straight router URLs. Customise this to match your site's look & feel. The emphasized lines show the lines that obtain and render the actual data, be it table or form. See *DYNAMICFORMS_PAGE_TEMPLATE*.

Listing 3: examples/templates/examples/page.html

```
{% extends 'examples/base.html' %}
{% load dynamicforms %}
{% block title %}
  {{ serializer.page_title }}
{% endblock %}
{% block body %}
  {% get_data_template as data_template %}

<div class="{{ DYNAMICFORMS.bs_card_class }}">
  <div class="{{ DYNAMICFORMS.bs_card_header }}">
    {{ serializer.page_title }}
    {% if serializer.render_type == 'table' %}{% render_table_commands serializer
↪'header' %}{% endif %}
  </div>
  <div class="{{ DYNAMICFORMS.bs_card_body }}">
    {% include data_template with serializer=serializer data=data %}
  </div>
</div>
{% endblock %}
```

Done. Point your DRF router to the ViewSet you just created and your browser to its URL - make sure you add ".html" to the URL to specify the renderer. If you forget that, you will get DRF's API renderer.

2.2 Templates

2.2.1 Design

Templates are organised in template packs for different UI libraries. DynamicForms provides template packs for bootstrap v3 / v4 with jQuery UI templates pending.

```
DYNAMICFORMS_TEMPLATE = 'dynamicforms/bootstrap'
```

Main template is base.html. HTML and its head tag are defined here. There are three blocks, that can be used in deriving templates:

- title: For defining the title of HTML.
- head_extra: to add additional definitions or includes in head tag.
- body: to insert the body of HTML.

Head tag includes base_includes.html (for bootstrap we have base_includes_v3.html and base_includes_v4.html). Here all the libraries that are needed for dynamic forms to work are included.

Base_list.html can be used for rendering Viewset in list mode. It shows all records with values or »No data« label if there is no data. When user clicks on record (only if default CRUD functionlity is enabled), this record is shown in form (modal dialog or separate page) and can be edited there.

Base_form.html can be used for rendering ViewSet in form mode. It shows one record, and if crud is enabled in Viewset, it can also be edited.

Form can be shown as modal dialog. For that template which is defined in settings.py - modal_dialog_rest_template is used. When using bootstrap v4 default template is modal_dialog_v4.html.

Template for dialog should have first div with »dynamicforms-dialog« class. JS searches for that to see if the response from server was a dialog or other error message.

For showing fields base template the one that is defined in settings.py -field_base_template. For bootstrap v4 default template is field/base_field_v4.html. That template makes sure that the label, input, errors and help text is correctly shown. This template is extracted by templates that are used for rendering individual field types (e.g.: checkbox.html, input.html, radio.html, etc.)

Some additional functionalities are not supported with jQuery templates (e.g. copy to clipboard on Select 2 fields).

Todo: explain how to change template to another one

Todo: Custom page template

Todo: Custom dialog classes for dialog formatting

Todo: SingleRecordViewSet

Todo: Custom refresh types

Todo: RenderMixin - parameters and what they do

CHAPTER 3

DynamicForms Configuration

Like much of django-based libraries, DynamicForms is also configured from settings.py.

3.1 Activate DynamicForms in DRF

In order to activate DynamicFroms, you need to add its renderers to DRF configuration, like so:

```
REST_FRAMEWORK = {
    'DEFAULT_RENDERER_CLASSES': (
        'rest_framework.renderers.JSONRenderer',
        'rest_framework.renderers.BrowsableAPIRenderer',
        'dynamicforms.renderers.TemplateHTMLRenderer',
    )
}
```

DRF will remain in control of JSON & Browseable API renderers while we activate DynamicForms for *.html* renders.

Note: The DRF renderers are taken from default DRF configuration. If it should change, feel free to change the setting as well.

3.2 List of settings

DYNAMICFORMS_TEMPLATE
: Specifies the template pack that dynamicforms will use for rendering HTML forms, e.g. 'bootstrap', 'jQuery UI', etc.

DYNAMICFORMS_PAGE_TEMPLATE
: Specifies the main page template to be used for plain rendering when you navigate to a ViewSet router URL.

Defaults to DYNAMICFORMS_TEMPLATE + 'page.html' (but currently, there's nothing there - see dynamicforms examples on how to specify base page template)

DYNAMICFORMS_TEMPLATE_OPTIONS

Offers a chance to do some things in the template pack differently. It can be used for anything from choosing version of the underlying framework (bootstrap 3 vs 4) or rendering various subsections differently (e.g. horizontally aligned form labels vs vertically aligned ones or editing record in modal dialog vs editing in new page).

Supported bootstrap versions are v3 and v4.

DYNAMICFORMS_MODAL_DIALOG

Name of template for modal dialog. It will be appended any version modifiers, i.e. bootstrap version postfix if bootstrap template pack.

CHAPTER 4

DynamicForms API reference

4.1 Action controls

4.1.1 Class reference

class FormButtonAction(*btn_type: dynamicforms.action.FormButtonTypes, label: str = None, btn_classes: str = None, action_js: str = None, button_is_primary: bool = None, positions: List[str] = None, name: Optional[str] = None, serializer: rest_framework.serializers.Serializer = None, icon: Optional[str] = None, action=None*)

class FormButtonTypes
An enumeration.

class RenderableActionMixin(*label: str, title: str, icon: str = None, btn_classes: Union[str, dict, None] = None*)
Action that is rendered on screen

class TableAction(*position: dynamicforms.action.TablePosition, label: str, action_js: str, title: Optional[str] = None, icon: Optional[str] = None, field_name: Optional[str] = None, name: Optional[str] = None, serializer: rest_framework.serializers.Serializer = None, btn_classes: Union[str, dict, None] = None, action=None*)

to_component_params(*row_data, serializer*)
generates a dict with parameters for component that is going to represent this action. none means don't render / activate this action on this row

Parameters

- **row_data** –
- **serializer** –

Returns

class TablePosition
An enumeration.

4.2 Context processors

add_dynamicforms_settings(*request: django.http.request.HttpRequest*)
context processor that adds DynamicForms configuration variables to template context

Example for supporting different versions of bootstrap:

```
{% if DYNAMICFORMS.bootstrap_version == 'v3' %}
  {% set_var card_class='panel panel-default' card_header='panel-heading' card_
↪body='panel_body' %}
{% else %}
  {% set_var card_class='card' card_header='card-header' card_body='card-body' %}
{% endif %}
```

Note: Using DynamicForms renderers automatically adds this variable into the context as it is required by template packs.

Parameters **request** – see django documentation

Returns dict with *DYNAMICFORMS* context variable set

4.3 Dialogs

Dialogs offer more flexibility by allowing you to place them on screen when needed instead of having them pre-rendered in the page. Their contents are always fresh and theirHTML adapts to the task at hand. For example, when user makes a mistake entering data, the returned dialog will already contain all the warnings in the HTML.

To use, either add *?df_render_type=dialog* to URL or add a *X_DF_RENDER_TYPE=dialog* HTTP header to request.

The default dialog templates allow for some customisation of dialog rendered.

4.3.1 Dialog classes

```
class MyViewSet(viewsets.ModelViewSet):
    template_context = dict(url_reverse='my-item', dialog_classes='dialog-lg')
```

Good for specifying additional dialog classes, like how large the dialog should be.

4.3.2 Dialog header classes

```
class MyViewSet(viewsets.ModelViewSet):
    template_context = dict(url_reverse='my-item', dialog_header_classes='bg-info')
```

Good for specifying additional dialog header classes, like typeof dialog (warning, info, primary, etc)

4.4 Fields

Re-declares all DRF fields such that they also inherit DynamicForms' mixins.

To use, import like so:

```
from dynamicforms.fields import {your desired field classes}
```

Make sure you don't import DRF's field classes over these.

4.4.1 Class reference

class AutoGeneratedField

class BooleanField(*read_only=False, write_only=False, required=None, default=<class 'rest_framework.fields.empty'>, initial=<class 'rest_framework.fields.empty'>, source=None, label=None, help_text=None, style=None, error_messages=None, validators=None, actions: dynamicforms.action.Actions = None, uuid: uuid.UUID = None, display: dynamicforms.mixins.field_render.DisplayMode = None, display_table: dynamicforms.mixins.field_render.DisplayMode = None, display_form: dynamicforms.mixins.field_render.DisplayMode = None, table_classes: str = '', alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>, render_params: Optional[Dict] = None, **kw*)

class CharField(*allow_blank: bool = False, trim_whitespace: bool = True, min_length: Optional[int] = None, max_length: Optional[int] = None, read_only=False, write_only=False, required=None, default=<class 'rest_framework.fields.empty'>, initial=<class 'rest_framework.fields.empty'>, source=None, label=None, help_text=None, style=None, error_messages=None, validators=None, allow_null=False, password_field=False, actions: dynamicforms.action.Actions = None, uuid: uuid.UUID = None, display: dynamicforms.mixins.field_render.DisplayMode = None, display_table: dynamicforms.mixins.field_render.DisplayMode = None, display_form: dynamicforms.mixins.field_render.DisplayMode = None, table_classes: str = '', alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>, render_params: Optional[Dict] = None, **kw*)

class ChoiceField(*choices, html_cutoff: int = None, html_cutoff_text: str = 'More than {count} items...', allow_blank: bool = False, read_only=False, write_only=False, required=None, default=<class 'rest_framework.fields.empty'>, initial=<class 'rest_framework.fields.empty'>, source=None, label=None, help_text=None, style=None, error_messages=None, validators=None, allow_null=False, allow_tags=False, actions: dynamicforms.action.Actions = None, uuid: uuid.UUID = None, display: dynamicforms.mixins.field_render.DisplayMode = None, display_table: dynamicforms.mixins.field_render.DisplayMode = None, display_form: dynamicforms.mixins.field_render.DisplayMode = None, table_classes: str = '', alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>, render_params: Optional[Dict] = None, **kw*)

class DateField(*format=<class 'rest_framework.fields.empty'>, input_formats=None, read_only=False, write_only=False, required=None, default=<class 'rest_framework.fields.empty'>, initial=<class 'rest_framework.fields.empty'>, source=None, label=None, help_text=None, style=None, error_messages=None, validators=None, allow_null=False, actions: dynamicforms.action.Actions = None, uuid: uuid.UUID = None, display: dynamicforms.mixins.field_render.DisplayMode = None, display_table: dynamicforms.mixins.field_render.DisplayMode = None, display_form: dynamicforms.mixins.field_render.DisplayMode = None, table_classes: str = '', alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>, render_params: Optional[Dict] = None, **kw*)

class DateTimeField(*format=<class 'rest_framework.fields.empty'>*, *input_formats=None*, *default_timezone=None*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class DecimalField(*max_digits*, *decimal_places*, *coerce_to_string=None*, *max_value=None*, *min_value=None*, *localize=False*, *rounding=None*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.RIGHT: 1>*, *render_params: Optional[Dict] = None*, ***kw*)

class DictField(*child=_UnvalidatedField()*, *allow_empty: bool = True*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class DurationField(*max_value: int = None*, *min_value: int = None*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class EmailField(*allow_blank: bool = False*, *trim_whitespace: bool = True*, *min_length: Optional[int] = None*, *max_length: Optional[int] = None*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *password_field=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class FileField(*max_length: Optional[int] = None*, *allow_empty_file: bool = False*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class FilePathField(*path*, *match=None*, *recursive=False*, *allow_files=True*, *allow_folders=False*, *required=None*, *html_cutoff: int = None*, *html_cutoff_text: str = 'More than {count} items...'*, *allow_blank: bool = False*, *read_only=False*, *write_only=False*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *allow_tags=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class FloatField(*max_value: int = None*, *min_value: int = None*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.DECIMAL: 2>*, *render_params: Optional[Dict] = None*, ***kw*)

class HStoreField(*child=_UnvalidatedField()*, *allow_empty: bool = True*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class HiddenField(*read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class HyperlinkedIdentityField(*view_name=None*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *url_reverse: Optional[str] = None*, *placeholder: Optional[str] = None*, *additional_parameters: Optional[Dict] = None*, *query_field: str = 'query'*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class HyperlinkedRelatedField(*view_name=None*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *url_reverse: Optional[str] = None*, *placeholder: Optional[str] = None*, *additional_parameters: Optional[Dict] = None*, *query_field: str = 'query'*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class IPAddressField(*protocol='both', allow_blank: bool = False, trim_whitespace: bool = True, min_length: Optional[int] = None, max_length: Optional[int] = None, read_only=False, write_only=False, required=None, default=<class 'rest_framework.fields.empty'>, initial=<class 'rest_framework.fields.empty'>, source=None, label=None, help_text=None, style=None, error_messages=None, validators=None, allow_null=False, password_field=False, actions: dynamicforms.action.Actions = None, uuid: uuid.UUID = None, display: dynamicforms.mixins.field_render.DisplayMode = None, display_table: dynamicforms.mixins.field_render.DisplayMode = None, display_form: dynamicforms.mixins.field_render.DisplayMode = None, table_classes: str = '', alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>, render_params: Optional[Dict] = None, **kw*)

class ImageField(*max_length: Optional[int] = None, allow_empty_file: bool = False, read_only=False, write_only=False, required=None, default=<class 'rest_framework.fields.empty'>, initial=<class 'rest_framework.fields.empty'>, source=None, label=None, help_text=None, style=None, error_messages=None, validators=None, allow_null=False, actions: dynamicforms.action.Actions = None, uuid: uuid.UUID = None, display: dynamicforms.mixins.field_render.DisplayMode = None, display_table: dynamicforms.mixins.field_render.DisplayMode = None, display_form: dynamicforms.mixins.field_render.DisplayMode = None, table_classes: str = '', alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>, render_params: Optional[Dict] = None, **kw*)

class IntegerField(*max_value: int = None, min_value: int = None, read_only=False, write_only=False, required=None, default=<class 'rest_framework.fields.empty'>, initial=<class 'rest_framework.fields.empty'>, source=None, label=None, help_text=None, style=None, error_messages=None, validators=None, allow_null=False, actions: dynamicforms.action.Actions = None, uuid: uuid.UUID = None, display: dynamicforms.mixins.field_render.DisplayMode = None, display_table: dynamicforms.mixins.field_render.DisplayMode = None, display_form: dynamicforms.mixins.field_render.DisplayMode = None, table_classes: str = '', alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.RIGHT: 1>, render_params: Optional[Dict] = None, **kw*)

class JSONField(*binary: bool = False, encoder=None, decoder=None, read_only=False, write_only=False, required=None, default=<class 'rest_framework.fields.empty'>, initial=<class 'rest_framework.fields.empty'>, source=None, label=None, help_text=None, style=None, error_messages=None, validators=None, allow_null=False, actions: dynamicforms.action.Actions = None, uuid: uuid.UUID = None, display: dynamicforms.mixins.field_render.DisplayMode = None, display_table: dynamicforms.mixins.field_render.DisplayMode = None, display_form: dynamicforms.mixins.field_render.DisplayMode = None, table_classes: str = '', alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>, render_params: Optional[Dict] = None, **kw*)

class ListField(*child=_UnvalidatedField(), allow_empty: bool = True, max_length: Optional[int] = None, min_length: Optional[int] = None, read_only=False, write_only=False, required=None, default=<class 'rest_framework.fields.empty'>, initial=<class 'rest_framework.fields.empty'>, source=None, label=None, help_text=None, style=None, error_messages=None, validators=None, allow_null=False, actions: dynamicforms.action.Actions = None, uuid: uuid.UUID = None, display: dynamicforms.mixins.field_render.DisplayMode = None, display_table: dynamicforms.mixins.field_render.DisplayMode = None, display_form: dynamicforms.mixins.field_render.DisplayMode = None, table_classes: str = '', alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>, render_params: Optional[Dict] = None, **kw*)

class ManyRelatedField(*child_relation=None, read_only=False, write_only=False, required=None, default=<class 'rest_framework.fields.empty'>, initial=<class 'rest_framework.fields.empty'>, source=None, label=None, help_text=None, style=None, error_messages=None, validators=None, allow_null=True, actions: dynamicforms.action.Actions = None, uuid: uuid.UUID = None, display: dynamicforms.mixins.field_render.DisplayMode = None, display_table: dynamicforms.mixins.field_render.DisplayMode = None, display_form: dynamicforms.mixins.field_render.DisplayMode = None, table_classes: str = '', alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>, render_params: Optional[Dict] = None, **kw*)

class ModelField(*model_field, max_length: Optional[int] = None, read_only=False, write_only=False, required=None, default=<class 'rest_framework.fields.empty'>, initial=<class 'rest_framework.fields.empty'>, source=None, label=None, help_text=None, style=None, error_messages=None, validators=None, allow_null=False, actions: dynamicforms.action.Actions = None, uuid: uuid.UUID = None, display: dynamicforms.mixins.field_render.DisplayMode = None, display_table: dynamicforms.mixins.field_render.DisplayMode = None, display_form: dynamicforms.mixins.field_render.DisplayMode = None, table_classes: str = '', alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>, render_params: Optional[Dict] = None, **kw*)

class MultipleChoiceField(*allow_empty: bool = True, html_cutoff: int = None, html_cutoff_text: str = 'More than {count} items...', allow_blank: bool = False, read_only=False, write_only=False, required=None, default=<class 'rest_framework.fields.empty'>, initial=<class 'rest_framework.fields.empty'>, source=None, label=None, help_text=None, style=None, error_messages=None, validators=None, allow_null=False, allow_tags=False, actions: dynamicforms.action.Actions = None, uuid: uuid.UUID = None, display: dynamicforms.mixins.field_render.DisplayMode = None, display_table: dynamicforms.mixins.field_render.DisplayMode = None, display_form: dynamicforms.mixins.field_render.DisplayMode = None, table_classes: str = '', alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>, render_params: Optional[Dict] = None, **kw*)

class NullBooleanField(*read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class PrimaryKeyRelatedField(*read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *url_reverse: Optional[str] = None*, *placeholder: Optional[str] = None*, *additional_parameters: Optional[Dict] = None*, *query_field: str = 'query'*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class RTFField(*actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class ReadOnlyField(*read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class RegexField(*regex*, *allow_blank: bool = False*, *trim_whitespace: bool = True*, *min_length: Optional[int] = None*, *max_length: Optional[int] = None*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *password_field=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class SerializerMethodField(*method_name=None*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class SlugField(*allow_unicode=False*, *allow_blank: bool = False*, *trim_whitespace: bool = True*, *min_length: Optional[int] = None*, *max_length: Optional[int] = None*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *password_field=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class SlugRelatedField(*slug_field=None*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *url_reverse: Optional[str] = None*, *placeholder: Optional[str] = None*, *additional_parameters: Optional[Dict] = None*, *query_field: str = 'query'*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class StringRelatedField(*read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *url_reverse: Optional[str] = None*, *placeholder: Optional[str] = None*, *additional_parameters: Optional[Dict] = None*, *query_field: str = 'query'*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class TimeField(*format=<class 'rest_framework.fields.empty'>*, *input_formats=None*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class URLField(*allow_blank: bool = False*, *trim_whitespace: bool = True*, *min_length: Optional[int] = None*, *max_length: Optional[int] = None*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *password_field=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

class UUIDField(*format: str = 'hex_verbose'*, *read_only=False*, *write_only=False*, *required=None*, *default=<class 'rest_framework.fields.empty'>*, *initial=<class 'rest_framework.fields.empty'>*, *source=None*, *label=None*, *help_text=None*, *style=None*, *error_messages=None*, *validators=None*, *allow_null=False*, *actions: dynamicforms.action.Actions = None*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kw*)

4.4.2 Field mixins

class DFField(**args*, *uuid: uuid.UUID = None*, *display: dynamicforms.mixins.field_render.DisplayMode = None*, *display_table: dynamicforms.mixins.field_render.DisplayMode = None*, *display_form: dynamicforms.mixins.field_render.DisplayMode = None*, *table_classes: str = ''*, *alignment: dynamicforms.mixins.field_render.FieldAlignment = <FieldAlignment.LEFT: -1>*, *render_params: Optional[Dict] = None*, ***kwargs*)

Class only for type hinting

4.5 Renderers

Contains template renderers.

To use, declare renderers in settings.py, like so:

```
REST_FRAMEWORK = {
    'DEFAULT_RENDERER_CLASSES': (
        'rest_framework.renderers.JSONRenderer',
        'rest_framework.renderers.BrowsableAPIRenderer',
        'dynamicforms.renderers.TemplateHTMLRenderer',
    )
}
```

4.5.1 Class reference

class HTMLFormRenderer

An HTML renderer for use with templates.

The data supplied to the Response object should be a dictionary that will be used as context for the template.

The template name is determined by (in order of preference):

1. An explicit *.template_name* attribute set on the response.
2. An explicit *.template_name* attribute set on this class.
3. The return result of calling *view.get_template_names()*.

For example: data = {'users': User.objects.all()} return Response(data, template_name='users.html')

render(*data*, *accepted_media_type=None*, *renderer_context=None*)

Render serializer data and return an HTML form, as a string.

class TemplateHTMLRenderer

Renderers serializer data into an HTML form.

If the serializer was instantiated without an object then this will return an HTML form not bound to any object, otherwise it will return an HTML form with the appropriate initial data populated from the object.

Note that rendering of field and form errors is not currently supported.

render(*data*, *accepted_media_type=None*, *renderer_context=None*)

Renders data to HTML, using Django's standard template rendering.

The template name is determined by (in order of preference):

1. An explicit .template_name set on the response.

2. An explicit .template_name set on this class.
3. The return result of calling view.get_template_names().

4.6 Serializers

Currently only provides ModelSerializer for model-based data manipulation.

To use, import like so:

```
from dynamicforms.serializers import ModelSerializer
```

Make sure you don't import DRF's ModelSerializer over this one.

4.6.1 Class reference

class ModelSerializer(**args*, *is_filter: bool = False*, ***kwds*)

DynamicForms' ModelSerializer overrides the following behaviour over DRF's implementation:

- Uses own field types for construction
- Adds form UUID (rendered in html too)
- Adds processing for form-wide errors

DRF's docstring copied verbatim:

A *ModelSerializer* is just a regular *Serializer*, except that:

- A set of default fields are automatically populated.
- A set of default validators are automatically populated.
- Default *.create()* and *.update()* implementations are provided.

The process of automatically determining a set of serializer fields based on the model fields is reasonably complex, but you almost certainly don't need to dig into the implementation.

If the *ModelSerializer* class *doesn't* generate the set of fields that you need you should either declare the extra/differing fields explicitly on the serializer class, or simply use a *Serializer* class.

get_extra_kwargs()

Return a dictionary mapping field names to a dictionary of additional keyword arguments.

manage_changed_flds()

When there is a need to only change few parameters of a field put those fields and changed parameters in serializers Meta class in parameter changed_flds.

Example:

```
changed_flds = {
    'id': dict(display=DisplayMode.HIDDEN),
    'comment': dict(label='Comm', help_text='Help text for comment field')
}
```

Returns

serializer_choice_field

alias of *dynamicforms.fields.ChoiceField*

serializer_related_field
alias of *dynamicforms.fields.PrimaryKeyRelatedField*

serializer_related_to_field
alias of *dynamicforms.fields.SlugRelatedField*

serializer_url_field
alias of *dynamicforms.fields.HyperlinkedIdentityField*

4.7 Template tags

4.7.1 Tag reference

get_data_template(*context*, *context_data=None*)
Returns template that should be used for rendering current serializer data

Parameters **context** – template context (automatically provided by django)

Returns template file name

render_field(*field*, *style*)
Renders separate field. Style is a dict, that contains template_pack or form_template and rendered. It is defined when rendering form, so it is best to use that one. To do that dynamicforms should be loaded first.

See example below.

```
{% load dynamicforms %}
{% for field in form %}
  {% if not field.read_only %}
    {% render_field field style=style %}
  {% endif %}
{% endfor %}
```

Parameters

- **field** – Serializer Field instance
- **style** – render parameters

Returns rendered field template

render_form(*serializer*, *form_template=None*)
Renders form from serializer. If form_template is given, then renderer will use that one, otherwise it will use what is defined in self.base_template (e.g.: »form.html«) from template_pack (e.g.: »dynamicforms/bootstrap/field/«)

```
{% set_var template_pack=DYNAMICFORMS.template|add:'field' %}
{% render_form serializer template_pack=template_pack %}
```

Parameters

- **serializer** – Serializer object
- **form_template** – form template file name to use. overrides all other template name declarations

Returns rendered template

set_var(*context*, ***kwds*)
Sets the given variables to provided values. Kind of like the 'with' block, only it isn't a block tag

Parameters

- **context** – template context (automatically provided by django)
- **kwds** – named parameters with their respective values

Returns this tag doesn't render

set_var_conditional(*context*, *condition=None*, *condition_var=None*, *compare=None*, *else_value=None*, ***kwds*)
Sets the given variables to provided values. Kind of like the 'with' block, only it isn't a block tag

Parameters

- **context** – template context (automatically provided by django)
- **kwds** – named parameters with their respective values
- **condition_var** – pair with compare to obtain True or False whether to use original assignment or else_value
- **compare** – pair with condition_var to obtain True or False whether to use original assignment or else_value
- **condition** – alternative to condition_var & compare: original assignment if truthy or else_value if falsy
- **else_value** – value to be assigned to the variable(s) when condition is falsy

Returns this tag doesn't render

render_table_commands(*context*, *serializer*, *position*, *field_name=None*, *button_position=None*)
Renders commands that are defined in serializers controls attribute.

Parameters

- **context** – Context
- **serializer** – Serializer
- **position** – Position of command (See action.py->Action for more details)
- **field_name** – If position is left or right to the field, then this parameter must contain field name
- **button_position** – form button position one of (form, dialog, user-provided)

Returns rendered command buttons. If table_header parameter is given and commands for position are defined, returns only rendered table header

table_columns_count(*serializer*)
Returns number of columns including control columns

Parameters **serializer** – Serializer

Returns Number of all columns

4.7.2 Filter reference

class ExtendTemplateNode(*nodelist*, *template_name*, *kwargs*, *only*, *multiline*)
Node for rendering extended template

render(*context*)
Return the node rendered as a string.

dict_item(*d*, *k*)
Returns the given key from a dictionary.

do_extendtemplate(*parser*, *token*)
Similar to base tag include, but with possibility to use blocks This tag is used as single line (without closing tag) For multiline use extendtemplateblock

Template name can be provided in two ways:

- If template name is static than declare it in first arg
- If template name is in variable than declare variable name in 'template_name_var' kwarg

Keyword parameters will be used as context when rendering template If there is "only" in parameters only keyword parameters will be used as context for rendering. Otherwise keywords will be added to existing context

do_extendtemplateblock(*parser*, *token*)
Same do_extendtemplate, only multiline... with posibillity to use blocks

field_to_serializer_and_data(*context*, *serializer*)
Adjusts context such that the nested serializer field becomes THE serializer

Parameters **serializer** – field to be converted into context

Returns nothing

items(*value*)
Simple filter to return the items of the dict. Useful when the dict may have a key 'items' which is resolved first in Django template dot-notation lookup. See DRF issue #4931 Also see: https://stackoverflow.com/questions/15416662/django-template-loop-over-dictionary-items-with-items-as-key

json(*value*, *field_list: str = None*)
JSON serialises given variable. Use when you want to insert a variable directly into JavaScript

Parameters

- **value** – variable to serialise
- **field_list** – if supplied, the "value" is assumed to be iterable of dict or object. only field_list members will be serialized

Returns JSON serialised string

template_from_string(*template_string*, *using=None*)
Convert a string into a template object, using a given template engine or using the default backends from settings.TEMPLATES if no engine was specified.

4.8 ViewSets

Currently only provides ModelViewSet for model-based data manipulation.

To use, import like so:

```
from dynamicforms.viewsets import ModelViewSet
```

Make sure you don't import DRF's ModelViewSet over this one.

4.8.1 Class reference

class ModelViewSet(**args*, ***kwds*)

In addition to all the functionality, provided by DRF, DynamicForms ViewSet has some extra features:

- Separate templates for rendering list or single record
- You can request a "new" record and even have it pre-populated with values
- To render viewset as API or JSON use the same method as in DRF: To render it in HTML just add ".html" to the URL.
- Standard DRF router URL patterns apply:
  - To render a new record use pk=new.
  - To render an existing record (for editing) use pk={record_id}.

new_object()

Returns a new model instance. If you need it pre-populated with default values, this is the method to override.

Returns model instance

filter_queryset(*queryset*, *query_params=None*)

Applies filters for all fields

Parameters

- **queryset** – Queryset
- **query_params** – Custom query_params if needed

Returns queryset with filters applied

filter_queryset_field(*queryset*, *field*, *value*)

Applies filter to individual field

Parameters

- **queryset** – Queryset
- **field** – Field name
- **value** – Field value

Returns queryset with applied filter for the field

static generate_paged_loader(*page_size: int = 30*, *ordering: Union[str*, *List[str]] = 'id'*)

Generates a Pagination class that will handle dynamic data loading for ViewSets with a lot of data. Use by declaring *pagination_class = ModelViewSet.generate_paged_loader()* in class variables

Parameters

- **page_size** – how many records should be fetched at a time
- **ordering** – This should be a string, or list of strings, indicating the field against which the cursor based pagination will be applied. For example: ordering = 'slug'

Returns a Pagination class

get_queryset()

Returns records from queryset with filters applied

Returns filtered records

CHAPTER 5

Contributor's documentation

5.1 Code style guide

5.1.1 Python

Rules are made to be broken, so. . .

PEP 8 with the following exceptions:

- E5 - line length: use 120 character limit
- E722 - (bare excepts) too broad
- E731 - do not assign a lambda expression, use a def

Our rationale for breaking E722 & E731:

- E722 - sometimes you just need code that doesn't except. Better to do nothing than to break. Depending on importance, such except blocks might have some sort of report added so that the programmer knows something happened.
- E731 - when you have tons of tiny declarations, using a lambda means 1 line of code while using def means 3 lines or more

Note that E722 & E731 may not be broken at all in this project. It's just that our linters have these checks disabled on principle. Apologies. . . but no regrets.

5.1.2 JavaScript

This one is a lot tougher. It is a work in progress, but I hope we (internally) sort it out before you (the would-be contributor) ever see this.

While we would **love** to adopt one of the popular style guides, we so far haven't managed to. No time doing research and finding a style that doesn't break what we feel is important. Suggestions based on the following welcome:

- Pure EcmaScript 5.1

    - No code conversion tools: the code in the repository runs straight out of the box
- Semicolons mandatory
- 2 spaces indentation
- use == instead of ===: type coercion is helpful

The above straight out eliminates:

- standard

These remain in play (we found all 5 on some "most popular linters" list)

- Airbnb
- Google
- Idiomatic
- jQuery

Note: While we're seriously considering EcmaScript 6, so far we haven't really encountered a situation where it would be absolutely required to do something. We try to be as light as possible on JavaScript anyway and while using EcmaScript 5.1 may mean a couple of lines more code, it also keeps away transpilers & missing-feature-completion scripts.

5.1.3 Documentation

- 120 character line length limit

5.2 dynamicforms_dev package

This package contains a helpful code generator so that we don't have to copy-paste an interface change to 35-ish different fields every time we add a cool new feature.

5.2.1 Generate fields

The helpful code generator. :)

This command is used for generating *fields/__init__.py*.

1. First it finds all the field types from *rest_framework/fields.py*
2. Then take care of all the imports that are needed for *fields/__init__.py*.
3. Then it finds all the parameters, that can be used to set up field and includes them in *__init__* functions for individual field. This we have done so that code completion might work better in your IDE. In the end it adds another parameter **kw which makes sure that any new parameters added in newer versions of DRF don't break the functionality.

```
python manage.py generate_fields
```

5.3 Requirements for running tests

```
pip install selenium
```

5.3.1 Gecko driver

Geckodriver is needed for the functional tests. It is available from https://github.com/mozilla/geckodriver/releases. You need to download and extract it and put it somewhere on your system path.

For macOS or Linux, one convenient place to put it is ~/.local/bin

For Windows, put it in your Python Scripts folder

To test that you've got this working, open up a Bash console and you should be able to run:

```
geckodriver --version
geckodriver 0.17.0
```

The source code of this program is available at https://github.com/mozilla/geckodriver.

This program is subject to the terms of the Mozilla Public License 2.0.

You can obtain a copy of the license at https://mozilla.org/MPL/2.0/.

5.4 creating your own template pack

5.4.1 Context variables referring to template packs

These variables will be set in a *dynamicforms.viewsets.ModelViewSet* class declaration.

crud_form: True | False
: If set, template pack is expected to render HTML that will allow user to edit the presented data.

Todo: The above is left just to keep an example. Currently there is nothing in the code that would be used

---

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

d

Index

M

N

P

R

S

T

U