
Django Fiction Outlines API Documentation

Release 0.1.3

Daniel Andrlik

Jun 21, 2018

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 3 |
| 1.1 | Documentation | 3 |
| 1.2 | Code Repo and Issue Tracker | 3 |
| 1.3 | License | 3 |
| 1.4 | Features | 3 |
| 2 | Installation | 5 |
| 3 | Usage | 7 |
| 4 | fiction_outlines_api | 9 |
| 4.1 | fiction_outlines_api package | 9 |
| 5 | Contributing | 29 |
| 5.1 | Types of Contributions | 29 |
| 5.2 | Get Started! | 30 |
| 5.3 | Pull Request Guidelines | 31 |
| 5.4 | Appropriate Conduct | 31 |
| 6 | Contributor Covenant Code of Conduct | 33 |
| 6.1 | Our Pledge | 33 |
| 6.2 | Our Standards | 33 |
| 6.3 | Our Responsibilities | 34 |
| 6.4 | Scope | 34 |
| 6.5 | Enforcement | 34 |
| 6.6 | Attribution | 34 |
| 7 | BSD License | 35 |
| 8 | Credits | 37 |
| 8.1 | Development Lead | 37 |
| 8.2 | Contributors | 37 |
| 9 | History | 39 |
| 9.1 | 0.1.3 (2018-06-21) | 39 |
| 9.2 | 0.1.0 (2018-04-07) | 39 |
| | Python Module Index | 41 |

Contents:

Welcome to Django Fiction Outlines API!

Being a reusable Django app for managing fiction outlines. Part of the broader [maceoutliner](#) project.

1.1 Documentation

The full documentation is at <https://django-fiction-outlines-api.readthedocs.io>.

1.2 Code Repo and Issue Tracker

The code repository and issue list for this project can be found at [Github](#).

1.3 License

BSD License for your convenience.

1.4 Features

- Provides a RESTful API for [django-fiction-outlines](#) suitable for serialization via JSON, XML, or [DRF's](#) browsable API.
- NOTE: Django Fiction Outlines API uses an object permission manager called [django-rules](#). This allows extremely flexible permission schemes without crafting up your database or model logic. By default, *fiction_outlines* will restrict any view or editing to the owner of the object.

CHAPTER 2

Installation

At the command line:

```
$ easy_install django-fiction-outlines-api
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv django-fiction-outlines-api
$ pip install django-fiction-outlines-api
```

Add it and dependencies to your *INSTALLED_APPS*:

```
INSTALLED_APPS = (
    ...
    'rest_framework',
    'taggit',
    'rules.apps.AutodiscoverRulesConfig',
    'rest_framework_rules',
    'fiction_outlines',
    'fiction_outlines_api',
    ...
)
```

If you have not already, add rules to your *AUTHENTICATION_BACKENDS*.

```
AUTHENTICATION_BACKENDS = (
    'rules.permissions.ObjectPermissionBackend',
    'django.contrib.auth.backends.ModelBackend',
)
```

Unless you like to live dangerously, it is **STRONGLY** recommend you configure whichever database you use for outlines to have *ATOMIC_REQUESTS* to *True*.

```
DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.postgresql",
        "NAME": "outlines",
        "ATOMIC_REQUESTS": True,
    }
}
```

Add Django Fiction Outlines API's URL patterns:

```
from fiction_outlines_api import urls as fiction_outlines_api_urls

urlpatterns = [
    ...
    url(r'^$', include(fiction_outlines_api_urls)),
    ...
]
```

If you haven't already installed `fiction_outlines` you should run `python manage.py migrate` now.

CHAPTER 3

Usage

It is expected that this API will be embedded into your project's overall DRF-based API. Simply include this app's URLs as documented in the *Installation* instructions, and follow the configuration documentation for DRF. You can generate full documentation for your resulting API either through DRF's *native generation tool* or you can use a third-party library such as *drf-yasg*.

Warning: The same *known caveats* of *django-fiction-outlines* also apply here.

- Do **NOT** attempt to create a new node using the Django-provided construction method. Use dedicated methods such as `add_root`, `add_child`, and `add_sibling` instead.
- Do **NOT** attempt to directly edit `path`, `step`, `depth`, `num_child`, etc. Use the provided *move* method.
- `MP_Node` uses a lot of raw SQL, so always retrieve the node from the db again after tree manipulation before calling it to do anything else.
- Object permissions come from *django-rules*, and the permission logic lies in the view layer. If you want to introduce your own custom logic, you should subclass the provided views in order to reduce the risk of security breaches.
- For the same reason, if you must define a custom manager, you **NEED** to subclass `treebeard`'s base `MP_Node` manager.

4.1 fiction_outlines_api package

4.1.1 Submodules

4.1.2 fiction_outlines_api.apps module

```
class fiction_outlines_api.apps.FictionOutlinesApiConfig(app_name, app_module)
    Bases: django.apps.config.AppConfig

    name = 'fiction_outlines_api'
    verbose_name = 'Fiction Outlines API'
```

4.1.3 fiction_outlines_api.exceptions module

```
exception fiction_outlines_api.exceptions.ServiceUnavailable(detail=None,
                                                             code=None)
    Bases: rest_framework.exceptions.APIException

    default_code = 'service_unavailable'
    default_detail = 'Service temporarily unavailable, please try again later.'
    status_code = 503

exception fiction_outlines_api.exceptions.TreeUnavailable(detail=None,
                                                            code=None)
    Bases: rest_framework.exceptions.APIException

    default_code = 'tree_unavailable'
    default_detail = 'We are having temporary issues with outlines at this time. Please try again later.'
    status_code = 503
```

4.1.4 fiction_outlines_api.mixins module

class fiction_outlines_api.mixins.**MultiObjectPermissionsMixin**

Bases: object

API Mixin that compares *n* objects and their permissions and returns if both are valid. Objects can be referred to by key afterwards in self.object_dict. permission dict should be:

```
{'obj1': {'obj_class': class_model, 'lookup_url_kwarg', 'object_permission_required': 'your rules perm here'}, ...}
```

Is it ugly, yes. Does it make doing this in view after view repeatable. Yep.

check_object_permissions (*request, perm, obj*)

Verifies the object permissions are valid for the current user.

Parameters

- **request** – The request object. The user is retrieved from here.
- **perm** – The permission to validate against (in **:module:'rules'** format).
- **obj** – The object to evaluate.

Raises **PermissionDenied** – when user lacks the required permission.

get_object_permission_queryset (*obj_class*)

Fetches the queryset for the permission object.

Parameters **obj_class** – Model class of the object to retrieve.

Returns A django queryset for the class.

get_permission_object (*pkval, objectpermission, obj_class*)

Fetches the permission object and checks it against the permission specified.

Parameters

- **pkval** – The pk value to use in the search.
- **objectpermission** – A permission based on the **:module:'rules'** format, e.g. "app_name.permission_name"
- **obj_class** – The class of the object we are searching for.

Returns The found object.

object_class_permission_dict = {}

post (*request, *args, **kwargs*)

class fiction_outlines_api.mixins.**NodeAddMixin**

Bases: object

API Mixin for add_sibling and add_child commands.

Attribute **fields_required_for_add** A tuple of fields that should be required in the submitted serializer in order to create the object.

create (*request, *args, **kwargs*)

Does the final validation of submitted data and if valid proceeds with creation.

Raises **NotImplementedError** – if client attempts to specify many-to-many relationship at creation.

fields_required_for_add = ('description',)

perform_create (*serializer*)

The actual creation part.

Parameters **serializer** – The serializer object representing the object to be created.

Raises **ValidationError** – if the data submitted is invalid or violates the structure of the tree.

Returns A serializer representing the created object.

post (*request, *args, **kwargs*)

Parses kwargs and kicks off evaluation process.

Parameters

- **action** – Either `add_child` or `add_sibling`.
- **position** – If `add_sibling` this is required, and must be a value from `:data: POSITIONS`. If action is `add_child` this is ignored.

Raises **Http404** – if action is not a one of the two permitted options.

class `fiction_outlines_api.mixins.NodeMoveMixin`

Bases: `object`

API mixin for move method for nodes.

get_object (*pkval*)

Fetches an individual object and verifies that the user has the appropriate permissions to move it.

Parameters **pkval** – The primary key value to use in the search.

Returns `Object`

Raises

- **Http404** – if object cannot be found
- **PermissionDenied** – if user does not have the required permissions.

perform_move ()

Validates the proposed move, and if valid, will manipulate the tree accordingly.

Raises

- **ValidationError** – if the move request violates tree structure.
- **TreeUnavailable** – if the DB tree is out of available nodes.

Returns A serializer representing the updated node.

pos = `None`

post (*request, *args, **kwargs*)

related_key = `None`

source_node = `None`

target_node = `None`

target_node_type_fieldname = `None`

4.1.5 `fiction_outlines_api.models` module

All models come from `fiction_outlines` so there is nothing to do here.

4.1.6 fiction_outlines_api.rules module

Rules definitions for api. We just import in the default rules from fiction_outlines.

4.1.7 fiction_outlines_api.serializers module

```
class fiction_outlines_api.serializers.ArcCreateSerializer (instance=None,  
                                                         data=<class  
                                                         'rest_framework.fields.empty'>,  
                                                         **kwargs)
```

Bases: `rest_framework.serializers.Serializer`

Serializer used for arc creation. You can't create an Arc using it's primary serializer.

```
class fiction_outlines_api.serializers.ArcElementNodeSerializer (instance=None,  
                                                         data=<class  
                                                         'rest_framework.fields.empty'>,  
                                                         **kwargs)
```

Bases: `rest_framework.serializers.ModelSerializer`

Serializer for ArcElementNode.

Special read-only attributes:

Attribute is_milestone Does this element represent a milestone in the Arc

Attribute milestone_seq If a milestone, what is the proscribed sequence where it should appear in the context of the other milestones.

Attribute parent_outline Convenience reference to the parent outline object.

Attribute headline Automatically generated from the first line of the description.

```
class Meta
```

Bases: `object`

fields = ('id', 'arc', 'arc_element_type', 'headline', 'description', 'story_element')

model

alias of `fiction_outlines.models.ArcElementNode`

read_only_fields = ('id', 'arc', 'headline', 'milestone_seq', 'is_milestone', 'parent_outline')

```
class fiction_outlines_api.serializers.ArcSerializer (instance=None, data=<class  
                                                         'rest_framework.fields.empty'>,  
                                                         **kwargs)
```

Bases: `rest_framework.serializers.ModelSerializer`

Serializer for Arc model.

To create an Arc, make sure to instead use the `ArcCreateSerializer` and post it to `fiction_outlines_api.views.ArcCreateView`

```
class Meta
```

Bases: `object`

fields = ('id', 'name', 'mace_type', 'outline', 'name', 'current_errors', 'arc_structure')

model

alias of `fiction_outlines.models.Arc`

read_only_fields = ('id', 'outline', 'current_errors', 'arc_structure')

get_arc_structure (*obj*)

Fetches all the `fiction_outlines.models.ArcElementNode` objects associated with this arc as an annotated list.

Parameters *obj* – The arc this serializer represents.

Returns An annotated list ready to be serialized or an empty list.

```
class fiction_outlines_api.serializers.CharacterInstanceSerializer (instance=None,  
                                                                data=<class  
                                                                'rest_framework.fields.empty'>,  
                                                                **kwargs)
```

Bases: `rest_framework.serializers.ModelSerializer`

Serializer for character instance.

class Meta

Bases: `object`

extra_kwargs = {'field': {'required': False}}

field = 'name'

fields = ('id', 'outline', 'character', 'name', 'main_character', 'pov_character',

model

alias of `fiction_outlines.models.CharacterInstance`

read_only_fields = ('id', 'outline', 'character', 'name')

```
class fiction_outlines_api.serializers.CharacterSerializer (instance=None,  
                                                         data=<class  
                                                         'rest_framework.fields.empty'>,  
                                                         **kwargs)
```

Bases: `taggit_serializer.serializers.TaggitSerializer`, `rest_framework.serializers.ModelSerializer`

Serializer for Character model.

class Meta

Bases: `object`

fields = ('id', 'name', 'description', 'series', 'tags', 'character_instances')

model

alias of `fiction_outlines.models.Character`

read_only_fields = ('id', 'character_instances')

```
class fiction_outlines_api.serializers.LocationInstanceSerializer (instance=None,  
                                                                data=<class  
                                                                'rest_framework.fields.empty'>,  
                                                                **kwargs)
```

Bases: `rest_framework.serializers.ModelSerializer`

Serializer for location instance.

class Meta

Bases: `object`

extra_kwargs = {'field': {'required': False}}

field = 'outline'

fields = ('id', 'location', 'name', 'outline')

```
model
    alias of fiction_outlines.models.LocationInstance

read_only_fields = ('id', 'location', 'name', 'outline')
```

class fiction_outlines_api.serializers.**LocationSerializer** (*instance=None*,
data=<class
'rest_framework.fields.empty'>,
**kwargs)

Bases: taggit_serializer.serializers.TaggitSerializer, rest_framework.
serializers.ModelSerializer

Serializer for Location model.

```
class Meta
    Bases: object

    fields = ('id', 'name', 'description', 'series', 'tags', 'location_instances')

    model
        alias of fiction_outlines.models.Location

    read_only_fields = ('id', 'location_instances')
```

class fiction_outlines_api.serializers.**OutlineSerializer** (*instance=None*,
data=<class
'rest_framework.fields.empty'>,
**kwargs)

Bases: taggit_serializer.serializers.TaggitSerializer, rest_framework.
serializers.ModelSerializer

Serializer for Outline model. Also provides `outline_structure`, which is a treebeard annotated list of story element nodes.

Special read-only fields:

- Attribute** `length_estimate` Auto calculated length (in words) based on the number of characters, locations, and arcs.
- Attribute** `arc_set` The `fiction_outlines.models.Arc` objects associated with this outline.
- Attribute** `characterinstance_set` Character instances associated with this outline.
- Attribute** `locationinstance_set` Location instances associated with this outline.
- Attribute** `outline_structure` An annotated list of `fiction_outlines.models.StoryElementNode` objects for visualizing the tree structure.

```
class Meta
    Bases: object

    fields = ('id', 'title', 'description', 'series', 'tags', 'length_estimate', 'arc_s

    model
        alias of fiction_outlines.models.Outline

    read_only_fields = ('id', 'length_estimate', 'arc_set', 'characterinstance_set', 'l
```

get_outline_structure (*obj*)

Retrieves all `fiction_outlines.StoryElementNode` objects associated with this outline, and returns it as an annotated list that can be serialized.

- Parameters** `obj` – The outline object this serializer represents.
- Returns** An annotated list for serialization or an empty list.

```
class fiction_outlines_api.serializers.SeriesSerializer (instance=None,  
                                                    data=<class  
                                                    'rest_framework.fields.empty'>,  
                                                    **kwargs)
```

Bases: taggit_serializer.serializers.TaggitSerializer, rest_framework.
serializers.ModelSerializer

Serializer for Series model.

```
class Meta
```

Bases: object

```
fields = ('id', 'title', 'description', 'tags')
```

```
model
```

alias of fiction_outlines.models.Series

```
read_only_fields = ('id',)
```

```
class fiction_outlines_api.serializers.StoryElementNodeSerializer (instance=None,  
                                                                data=<class  
                                                                'rest_framework.fields.empty'>,  
                                                                **kwargs)
```

Bases: rest_framework.serializers.ModelSerializer

Serializer for StoryElementNode

Two special fields here:

Attribute all_characters All characters associated with this node or its descendants.

Attribute all_locations All locations associated with this node of its descendants.

```
class Meta
```

Bases: object

```
fields = ('id', 'story_element_type', 'name', 'description', 'outline', 'assoc_char
```

```
model
```

alias of fiction_outlines.models.StoryElementNode

```
read_only_fields = ('all_characters', 'all_locations', 'impact_rating', 'outline',
```

```
fiction_outlines_api.serializers.convert_annotated_list (annotated_list,    serial-  
                                                         izer_class)
```

Takes an annotated list and a serializer class and returns a version that is suitable for serialization.

4.1.8 fiction_outlines_api.urls module

4.1.9 fiction_outlines_api.views module

```
class fiction_outlines_api.views.ArcCreateView (**kwargs)
```

Bases: rest_framework_rules.mixins.PermissionRequiredMixin, rest_framework.
generics.CreateAPIView

API for creating arcs. Uses a custom serializer, as Arcs are generated via special methods inside a transaction.

Provides HTTP method:

- POST: Accepts data compatible with `fiction_outlines_api.serializers.ArcCreateSerializer`

```
create (request, *args, **kwargs)
```

dispatch (*request*, **args*, ***kwargs*)

.dispatch() is pretty much the same as Django's regular dispatch, but with extra hooks for startup, finalize, and exception handling.

object_permission_required = 'fiction_outlines.edit_outline'

perform_create (*serializer*)

permission_required = 'fiction_outlines_api.valid_user'

post (*request*, **args*, ***kwargs*)

serializer_class

alias of *fiction_outlines_api.serializers.ArcCreateSerializer*

class *fiction_outlines_api.views.ArcDetailView* (***kwargs*)

Bases: *rest_framework_rules.mixins.PermissionRequiredMixin*, *rest_framework.generics.RetrieveUpdateDestroyAPIView*

API for non-create object operations for Arc model.

Provides HTTP methods:

- GET: Retrieve an individual object.
- PUT: Update the object with data compatible with a *fiction_outlines_api.serializers.ArcSerializer*
- PATCH: Update the object with partial data that corresponds to keys in the serializer.
- DELETE: Delete the object.

delete (*request*, **args*, ***kwargs*)

get_queryset ()

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using *self.queryset*.

This method should always be used rather than accessing *self.queryset* directly, as *self.queryset* gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

lookup_url_kwarg = 'arc'

object_permission_required = 'fiction_outlines.view_arc'

patch (*request*, **args*, ***kwargs*)

permission_required = 'fiction_outlines_api.valid_user'

put (*request*, **args*, ***kwargs*)

serializer_class

alias of *fiction_outlines_api.serializers.ArcSerializer*

class *fiction_outlines_api.views.ArcNodeCreateView* (***kwargs*)

Bases: *rest_framework_rules.mixins.PermissionRequiredMixin*, *fiction_outlines_api.mixins.NodeAddMixin*, *rest_framework.generics.CreateAPIView*

API view for add_child and add_sibling. You can only create tree objects in relation to another object in the tree. See *mixins.NodeAddMixin* for more details.

```
fields_required_for_add = ('arc_element_type', 'description', 'story_element_node')
```

```
get_queryset()
```

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using *self.queryset*.

This method should always be used rather than accessing *self.queryset* directly, as *self.queryset* gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

```
lookup_url_kwarg = 'arcnode'
```

```
object_permission_required = 'fiction_outlines.edit_arc_node'
```

```
permission_required = 'fiction_outlines_api.valid_user'
```

```
serializer_class
```

alias of *fiction_outlines_api.serializers.ArcElementNodeSerializer*

```
class fiction_outlines_api.views.ArcNodeDetailView(**kwargs)
```

Bases: *rest_framework_rules.mixins.PermissionRequiredMixin*, *rest_framework.generics.RetrieveUpdateDestroyAPIView*

API for viewing a tree of *fiction_outlines.models.ArcElementNode*, as well as some basic updates.

Provides HTTP methods:

- GET: Retrieve an individual object.
- PUT: Update the object with data compatible with a *fiction_outlines_api.serializers.ArcElementNodeSerializer*
- PATCH: Update the object with partial data that corresponds to keys in the serializer.
- DELETE: Delete the object.

```
delete(request, *args, **kwargs)
```

```
destroy(request, *args, **kwargs)
```

Destroys the object, but first checks that the element doesn't represent either the Hook or Resolution of an Arc. These types may not be deleted unless you are deleting the whole arc.

Raises *ParseError* – if the object represents the Hook or Resolution.

```
get_queryset()
```

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using *self.queryset*.

This method should always be used rather than accessing *self.queryset* directly, as *self.queryset* gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

```
lookup_url_kwarg = 'arcnode'
```

```
object_permission_required = 'fiction_outlines.view_arc_node'
```

```
patch(request, *args, **kwargs)
```

```
permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,)
permission_required = 'fiction_outlines_api.valid_user'
put (request, *args, **kwargs)

serializer_class
    alias of fiction_outlines_api.serializers.ArcElementNodeSerializer

update (request, *args, **kwargs)
```

class `fiction_outlines_api.views.ArcNodeMoveView` (***kwargs*)

Bases: `rest_framework_rules.mixins.PermissionRequiredMixin`, `fiction_outlines_api.mixins.NodeMoveMixin`, `rest_framework.generics.GenericAPIView`

View for moving arc nodes. See `mixins.NodeMoveMixin` for more details.

get_queryset ()

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using `self.queryset`.

This method should always be used rather than accessing `self.queryset` directly, as `self.queryset` gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

```
object_permission_required = 'fiction_outlines.edit_arc_node'
permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,)
permission_required = 'fiction_outlines_api.valid_user'
related_key = 'arc'

serializer_class
    alias of fiction_outlines_api.serializers.ArcElementNodeSerializer

target_node_type_fieldname = 'arc_element_type'
```

class `fiction_outlines_api.views.CharacterDetail` (***kwargs*)

Bases: `rest_framework_rules.mixins.PermissionRequiredMixin`, `rest_framework.generics.RetrieveUpdateDestroyAPIView`

API view for all single item character operations besides create.

Provides HTTP methods:

- GET: Retrieve an individual character record.
- PUT: Update a character with data compatible with a `fiction_outlines_api.serializers.CharacterSerializer`
- PATCH: Update a character with partial data that corresponds to keys in `fiction_outlines_api.serializers.CharacterSerializer`

delete (*request, *args, **kwargs*)

get_queryset ()

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using `self.queryset`.

This method should always be used rather than accessing *self.queryset* directly, as *self.queryset* gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different queriesets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

```
lookup_url_kwarg = 'character'
```

```
object_permission_required = 'fiction_outlines.view_character'
```

```
patch(request, *args, **kwargs)
```

```
perform_update(serializer)
```

Validates that if a series is specified, that the user has rights to it as well.

Raises PermissionDenied – if the user does not have the required permissions.

```
permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,) )
```

```
permission_required = 'fiction_outlines_api.valid_user'
```

```
put(request, *args, **kwargs)
```

```
serializer_class
```

alias of *fiction_outlines_api.serializers.CharacterSerializer*

```
class fiction_outlines_api.views.CharacterInstanceCreateView(**kwargs)
```

Bases: *fiction_outlines_api.mixins.MultiObjectPermissionsMixin*,
rest_framework_rules.mixins.PermissionRequiredMixin, *rest_framework*.
generics.CreateAPIView

API view for creating a character instance. Expects kwargs in url for character and outline. All other data comes from the serializer.

Provides HTTP methods:

- POST: Accepts data compatible with a *fiction_outlines_api.serializers.CharacterInstanceSerializer*

```
object_class_permission_dict = {'character': {'obj_class': <class 'fiction_outlines_api.views.CharacterInstanceCreateView'>}}
```

```
perform_create(serializer)
```

Attempts the actual creation of the instance.

Raises ParseError – if the character is already associated with the outline.

```
permission_required = 'fiction_outlines_api.valid_user'
```

```
serializer_class
```

alias of *fiction_outlines_api.serializers.CharacterInstanceSerializer*

```
class fiction_outlines_api.views.CharacterInstanceDetailView(**kwargs)
```

Bases: *rest_framework_rules.mixins.PermissionRequiredMixin*, *rest_framework*.
generics.RetrieveUpdateDestroyAPIView

API view for non-creation actions on CharacterInstance objects.

Provides HTTP methods:

- GET: Retrieve an individual object.
- PUT: Update the object with data compatible with a *fiction_outlines_api.serializers.CharacterInstanceSerializer*
- PATCH: Update the object with partial data that corresponds to keys in the serializer.

- DELETE: Delete the object.

delete (*request*, **args*, ***kwargs*)

get_queryset ()

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using *self.queryset*.

This method should always be used rather than accessing *self.queryset* directly, as *self.queryset* gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

lookup_url_kwarg = 'instance'

object_permission_required = 'fiction_outlines.view_character_instance'

patch (*request*, **args*, ***kwargs*)

permission_required = 'fiction_outlines_api.valid_user'

put (*request*, **args*, ***kwargs*)

serializer_class

alias of *fiction_outlines_api.serializers.CharacterInstanceSerializer*

class *fiction_outlines_api.views.CharacterList* (***kwargs*)

Bases: *rest_framework.generics.ListCreateAPIView*

API view for character list

Provides HTTP methods:

- GET: Retrieve a list of *fiction_outlines.models.Character* objects for the current user.
- POST: Create a character. Takes data compatible with *fiction_outlines_api.serializers.CharacterSerializer*

get_queryset ()

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using *self.queryset*.

This method should always be used rather than accessing *self.queryset* directly, as *self.queryset* gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

perform_create (*serializer*)

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,))

serializer_class

alias of *fiction_outlines_api.serializers.CharacterSerializer*

class *fiction_outlines_api.views.LocationDetail* (***kwargs*)

Bases: *rest_framework_rules.mixins.PermissionRequiredMixin*, *rest_framework.generics.RetrieveUpdateDestroyAPIView*

API view for all single item location operations besides create.

Provides HTTP methods:

- GET: Retrieve an individual object.
- PUT: Update the object with data compatible with a `fiction_outlines_api.serializers.LocationSerializer`
- PATCH: Update the object with partial data that corresponds to keys in the serializer.
- DELETE: Delete the object.

`delete (request, *args, **kwargs)`

`get_queryset ()`

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using `self.queryset`.

This method should always be used rather than accessing `self.queryset` directly, as `self.queryset` gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

`lookup_url_kwarg = 'location'`

`object_permission_required = 'fiction_outlines.view_location'`

`patch (request, *args, **kwargs)`

`perform_update (serializer)`

Performs update actions. If a series is specified, it verifies the user has rights to it as well.

Raises `PermissionDenied` – if the user does not have the correct permissions.

`permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,)`

`permission_required = 'fiction_outlines_api.valid_user'`

`put (request, *args, **kwargs)`

`serializer_class`

alias of `fiction_outlines_api.serializers.LocationSerializer`

`class fiction_outlines_api.views.LocationInstanceCreateView (**kwargs)`

Bases: `fiction_outlines_api.mixins.MultiObjectPermissionsMixin`,
`rest_framework_rules.mixins.PermissionRequiredMixin`, `rest_framework`.
`generics.CreateAPIView`

API view for creating a location instance. Expects kwargs in url for location and outline. All other data comes from the serializer.

Provides HTTP method:

- POST: accepts data compatible with `fiction_outlines_api.serializers.LocationInstanceSerializer`

`object_class_permission_dict = {'location': {'obj_class': <class 'fiction_outlines.m`

`perform_create (serializer)`

Creates the location instance.

Raises `ParseError` – if the location is already associated with the outline.

`permission_required = 'fiction_outlines_api.valid_user'`

`serializer_class`

alias of `fiction_outlines_api.serializers.LocationInstanceSerializer`

```
class fiction_outlines_api.views.LocationInstanceDetailView(**kwargs)
```

Bases: `rest_framework_rules.mixins.PermissionRequiredMixin`, `rest_framework.generics.RetrieveDestroyAPIView`

API view for non-creation actions on `LocationInstance` objects. As locations instances don't have editable data, only retrieval and destroy are supported at this time.

Provides HTTP methods:

- GET: Retrieve an individual object.
- DELETE: Delete the object.

```
delete (request, *args, **kwargs)
```

```
get_queryset ()
```

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using `self.queryset`.

This method should always be used rather than accessing `self.queryset` directly, as `self.queryset` gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

```
lookup_url_kwarg = 'instance'
```

```
object_permission_required = 'fiction_outlines.view_location_instance'
```

```
permission_required = 'fiction_outlines_api.valid_user'
```

```
serializer_class
```

alias of `fiction_outlines_api.serializers.LocationInstanceSerializer`

```
class fiction_outlines_api.views.LocationList(**kwargs)
```

Bases: `rest_framework.generics.ListCreateAPIView`

API view for location list

Provides HTTP methods:

- GET: Retrieve a list of Locations for the current user.
- POST: Create a location with data compatible with `fiction_outlines_api.serializers.LocationSerializer`

```
get_queryset ()
```

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using `self.queryset`.

This method should always be used rather than accessing `self.queryset` directly, as `self.queryset` gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

```
perform_create (serializer)
```

Creates the object. If a series is specified, verify the user has rights to it as well.

Raises `PermissionDenied` – if the user lackst he required permissions.

```
permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,)
```

serializer_classalias of *fiction_outlines_api.serializers.LocationSerializer***class** *fiction_outlines_api.views.OutlineDetail* (**kwargs)Bases: *rest_framework_rules.mixins.PermissionRequiredMixin*, *rest_framework.generics.RetrieveUpdateDestroyAPIView*

API view for all single item outline operations besides create.

Provides HTTP methods:

- GET: Retrieve an individual object.
- PUT: Update the object with data compatible with a *fiction_outlines_api.serializers.OutlineSerializer*
- PATCH: Update the object with partial data that corresponds to keys in the serializer.
- DELETE: Delete the object.

delete (*request*, *args, **kwargs)**get_queryset** ()Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using *self.queryset*.This method should always be used rather than accessing *self.queryset* directly, as *self.queryset* gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

lookup_url_kwarg = 'outline'**object_permission_required** = 'fiction_outlines.view_outline'**patch** (*request*, *args, **kwargs)**perform_update** (*serializer*)

Updates the object. If a series is specified, it verifies the user has permissions for that object as well.

Raises *PermissionDenied* – if the user lacks the needed permissions.**permission_classes** = (<class 'rest_framework.permissions.IsAuthenticated'>,))**permission_required** = 'fiction_outlines_api.valid_user'**put** (*request*, *args, **kwargs)**serializer_class**alias of *fiction_outlines_api.serializers.OutlineSerializer***class** *fiction_outlines_api.views.OutlineList* (**kwargs)Bases: *rest_framework.generics.ListCreateAPIView*

API view for Outline list

Provides HTTP methods:

- GET: Retrived a list of outlines for the current user.
- POST: Create a new Outline. Accepts data compatible with *fiction_outlines_api.serializers.OutlineSerializer*

get_queryset ()

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using *self.queryset*.

This method should always be used rather than accessing *self.queryset* directly, as *self.queryset* gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

perform_create (serializer)

Creates the object. If a series is specified, it verifies that the user has rights to it as well.

Raises `PermissionDenied` – if the user does not have the required permissions.

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,))

serializer_class

alias of *fiction_outlines_api.serializers.OutlineSerializer*

class fiction_outlines_api.views.**SeriesDetail** (**kwargs)

Bases: rest_framework_rules.mixins.PermissionRequiredMixin, rest_framework.generics.RetrieveUpdateDestroyAPIView

Retrieves details of a series, and enables editing of the object.

Provides HTTP methods:

- GET: retrieve the individual object.
- PUT: Update the object. Takes data compatible with a *fiction_outlines_api.serializers.SeriesSerializer*
- PATCH: Update the object. Takes partial data compatible with keys from *fiction_outlines_api.serializers.SeriesSerializer*
- DELETE: Delete the object.

delete (request, *args, **kwargs)

get_queryset ()

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using *self.queryset*.

This method should always be used rather than accessing *self.queryset* directly, as *self.queryset* gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

lookup_url_kwarg = 'series'

object_permission_required = 'fiction_outlines.view_series'

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,))

permission_required = 'fiction_outlines_api.valid_user'

put (request, *args, **kwargs)

serializer_class

alias of *fiction_outlines_api.serializers.SeriesSerializer*

```
class fiction_outlines_api.views.SeriesList (**kwargs)
    Bases: rest_framework.generics.ListCreateAPIView

    API view for series list.

    Provides HTTP methods:

        • GET: Retrive fiction_outlines.models.Series objects for the current user.

        • POST: Accepts data compatible with a fiction_outlines_api.serializers.
              SeriesSerializer

get_queryset ()
    Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using
    self.queryset.

    This method should always be used rather than accessing self.queryset directly, as self.queryset gets eval-
    uated only once, and those results are cached for all subsequent requests.

    You may want to override this if you need to provide different querysets depending on the incoming
    request.

    (Eg. return a list of items that is specific to the user)

perform_create (serializer)

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,)

serializer_class
    alias of fiction_outlines_api.serializers.SeriesSerializer

class fiction_outlines_api.views.StoryNodeCreateView (**kwargs)
    Bases: rest_framework_rules.mixins.PermissionRequiredMixin,
           fiction_outlines_api.mixins.NodeAddMixin, rest_framework.generics.
           CreateAPIView

    View for adding story nodes. You can only create tree elements in relation to other objects in the same tree. See
    mixins.NodeAddMixin for more details.

fields_required_for_add = ('story_element_type', 'name', 'description')

get_queryset ()
    Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using
    self.queryset.

    This method should always be used rather than accessing self.queryset directly, as self.queryset gets eval-
    uated only once, and those results are cached for all subsequent requests.

    You may want to override this if you need to provide different querysets depending on the incoming
    request.

    (Eg. return a list of items that is specific to the user)

lookup_url_kwarg = 'storynode'

object_permission_required = 'fiction_outlines.edit_story_node'

permission_required = 'fiction_outlines_api.valid_user'

serializer_class
    alias of fiction_outlines_api.serializers.StoryElementNodeSerializer

class fiction_outlines_api.views.StoryNodeDetailView (**kwargs)
    Bases: rest_framework_rules.mixins.PermissionRequiredMixin, rest_framework.
           generics.RetrieveUpdateDestroyAPIView
```

API for viewing and editing a story node.

Provides HTTP methods:

- GET: Retrieve an individual object.
- PUT: Update the object with data compatible with a `fiction_outlines_api.serializers.StoryElementNodeSerializer`
- PATCH: Update the object with partial data that corresponds to keys in the serializer.
- DELETE: Delete the object.

delete (*request*, *args, **kwargs)

get_queryset ()

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using *self.queryset*.

This method should always be used rather than accessing *self.queryset* directly, as *self.queryset* gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

lookup_url_kwarg = 'storynode'

object_permission_required = 'fiction_outlines.view_story_node'

patch (*request*, *args, **kwargs)

permission_classes = (<class 'rest_framework.permissions.IsAuthenticated'>,))

permission_required = 'fiction_outlines_api.valid_user'

put (*request*, *args, **kwargs)

serializer_class

alias of `fiction_outlines_api.serializers.StoryElementNodeSerializer`

update (*request*, *args, **kwargs)

Updates the node, after first running it through validation to ensure it won't violate the tree structure.

Raises `ParseError` – if the edit would violate the tree structure.

class `fiction_outlines_api.views.StoryNodeMoveView` (**kwargs)

Bases: `rest_framework_rules.mixins.PermissionRequiredMixin`,
`fiction_outlines_api.mixins.NodeMoveMixin`, `rest_framework.generics.GenericAPIView`

View for moving story nodes. See `mixins.NodeMoveMixin` for more details.

get_queryset ()

Get the list of items for this view. This must be an iterable, and may be a queryset. Defaults to using *self.queryset*.

This method should always be used rather than accessing *self.queryset* directly, as *self.queryset* gets evaluated only once, and those results are cached for all subsequent requests.

You may want to override this if you need to provide different querysets depending on the incoming request.

(Eg. return a list of items that is specific to the user)

object_permission_required = 'fiction_outlines.edit_story_node'

```
permission_required = 'fiction_outlines_api.valid_user'
related_key = 'outline'
serializer_class
    alias of fiction_outlines_api.serializers.StoryElementNodeSerializer
target_node_type_fieldname = 'story_element_type'
```

4.1.10 Module contents

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at <https://github.com/maceoutliner/django-fiction-outlines-api/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

5.1.4 Write Documentation

Django Fiction Outlines API could always use more documentation, whether as part of the official Django Fiction Outlines docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/maceoutliner/django-fiction-outlines-api/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up `django-fiction-outlines-api` for local development.

1. Fork the `django-fiction-outlines-api` repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/django-fiction-outlines-api.git
```

3. Install your local copy into a virtualenv. Assuming you have `pipenv` installed, this is how you set up your fork for local development:

```
$ cd django-fiction-outlines-api/  
$ pipenv install --dev
```

Alternatively:

```
$ pip install test_requirements.txt
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

It is preferred and appreciated if you follow [this style guide](#) for your commit messages.

5. When you're done making changes, check that your changes pass `flake8` and the tests.:

```
$ flake8 fiction_outlines_api tests  
$ pytest
```

To get `flake8` and `tox`, just `pip` install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .  
$ git commit -m "Your detailed description of your changes."  
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python for Python 3.6 and up. Check https://travis-ci.org/maceoutliner/django-fiction-outlines-api/pull_requests and make sure that the tests pass for all supported Python versions. (Don't submit issues or PR to enable Python 2 support. They will not be merged.)
4. Where appropriate, squash your commits using `git rebase -i`.

5.4 Appropriate Conduct

All contributors and project participants are expected to follow our *Contributor Covenant Code of Conduct*.

Contributor Covenant Code of Conduct

6.1 Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

6.2 Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

6.3 Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

6.4 Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

6.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at coc@maceoutliner.org. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

6.6 Attribution

This Code of Conduct is adapted from the [Contributor Covenant](https://www.contributor-covenant.org/version/1/4/code-of-conduct.html), version 1.4, available at <https://www.contributor-covenant.org/version/1/4/code-of-conduct.html>

CHAPTER 7

BSD License

The Clear BSD License

Copyright (c) 2018, Daniel Andrlik All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted (subject to the limitations in the disclaimer below) provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

NO EXPRESS OR IMPLIED LICENSES TO ANY PARTY'S PATENT RIGHTS ARE GRANTED BY THIS LICENSE. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

8.1 Development Lead

- Daniel Andrlik <daniel@andrlik.org>

8.2 Contributors

None yet. Why not be the first?

9.1 0.1.3 (2018-06-21)

- Move dependency back to official django-taggit-serializer package now that our pull request has been merged.

9.2 0.1.0 (2018-04-07)

- First release on PyPI.

f

- `fiction_outlines_api`, [27](#)
- `fiction_outlines_api.apps`, [9](#)
- `fiction_outlines_api.exceptions`, [9](#)
- `fiction_outlines_api.mixins`, [10](#)
- `fiction_outlines_api.models`, [11](#)
- `fiction_outlines_api.rules`, [12](#)
- `fiction_outlines_api.serializers`, [12](#)
- `fiction_outlines_api.urls`, [15](#)
- `fiction_outlines_api.views`, [15](#)

A

ArcCreateSerializer (class in fiction_outlines_api.serializers), 12

ArcCreateView (class in fiction_outlines_api.views), 15

ArcDetailView (class in fiction_outlines_api.views), 16

ArcElementNodeSerializer (class in fiction_outlines_api.serializers), 12

ArcElementNodeSerializer.Meta (class in fiction_outlines_api.serializers), 12

ArcNodeCreateView (class in fiction_outlines_api.views), 16

ArcNodeDetailView (class in fiction_outlines_api.views), 17

ArcNodeMoveView (class in fiction_outlines_api.views), 18

ArcSerializer (class in fiction_outlines_api.serializers), 12

ArcSerializer.Meta (class in fiction_outlines_api.serializers), 12

C

CharacterDetail (class in fiction_outlines_api.views), 18

CharacterInstanceCreateView (class in fiction_outlines_api.views), 19

CharacterInstanceDetailView (class in fiction_outlines_api.views), 19

CharacterInstanceSerializer (class in fiction_outlines_api.serializers), 13

CharacterInstanceSerializer.Meta (class in fiction_outlines_api.serializers), 13

CharacterList (class in fiction_outlines_api.views), 20

CharacterSerializer (class in fiction_outlines_api.serializers), 13

CharacterSerializer.Meta (class in fiction_outlines_api.serializers), 13

check_object_permissions() (fiction_outlines_api.mixins.MultiObjectPermissionsMixin method), 10

convert_annotated_list() (in module fic-

tion_outlines_api.serializers), 15

create() (fiction_outlines_api.mixins.NodeAddMixin method), 10

create() (fiction_outlines_api.views.ArcCreateView method), 15

D

default_code (fiction_outlines_api.exceptions.ServiceUnavailable attribute), 9

default_code (fiction_outlines_api.exceptions.TreeUnavailable attribute), 9

default_detail (fiction_outlines_api.exceptions.ServiceUnavailable attribute), 9

default_detail (fiction_outlines_api.exceptions.TreeUnavailable attribute), 9

delete() (fiction_outlines_api.views.ArcDetailView method), 16

delete() (fiction_outlines_api.views.ArcNodeDetailView method), 17

delete() (fiction_outlines_api.views.CharacterDetail method), 18

delete() (fiction_outlines_api.views.CharacterInstanceDetailView method), 20

delete() (fiction_outlines_api.views.LocationDetail method), 21

delete() (fiction_outlines_api.views.LocationInstanceDetailView method), 22

delete() (fiction_outlines_api.views.OutlineDetail method), 23

delete() (fiction_outlines_api.views.SeriesDetail method), 24

delete() (fiction_outlines_api.views.StoryNodeDetailView method), 26

destroy() (fiction_outlines_api.views.ArcNodeDetailView method), 17

dispatch() (fiction_outlines_api.views.ArcCreateView method), 15

E

extra_kwargs (fiction_outlines_api.serializers.CharacterInstanceSerializer.M

- attribute), 13
- extra_kwargs (fiction_outlines_api.serializers.LocationInstanceSerializer.Meta attribute), 13
- ## F
- fiction_outlines_api (module), 27
- fiction_outlines_api.apps (module), 9
- fiction_outlines_api.exceptions (module), 9
- fiction_outlines_api.mixins (module), 10
- fiction_outlines_api.models (module), 11
- fiction_outlines_api.rules (module), 12
- fiction_outlines_api.serializers (module), 12
- fiction_outlines_api.urls (module), 15
- fiction_outlines_api.views (module), 15
- FictionOutlinesApiConfig (class in fiction_outlines_api.apps), 9
- field (fiction_outlines_api.serializers.CharacterInstanceSerializer.Meta attribute), 13
- field (fiction_outlines_api.serializers.LocationInstanceSerializer.Meta attribute), 13
- fields (fiction_outlines_api.serializers.ArcElementNodeSerializer.Meta attribute), 12
- fields (fiction_outlines_api.serializers.ArcSerializer.Meta attribute), 12
- fields (fiction_outlines_api.serializers.CharacterInstanceSerializer.Meta attribute), 13
- fields (fiction_outlines_api.serializers.CharacterSerializer.Meta attribute), 13
- fields (fiction_outlines_api.serializers.LocationInstanceSerializer.Meta attribute), 13
- fields (fiction_outlines_api.serializers.LocationSerializer.Meta attribute), 14
- fields (fiction_outlines_api.serializers.OutlineSerializer.Meta attribute), 14
- fields (fiction_outlines_api.serializers.SeriesSerializer.Meta attribute), 15
- fields (fiction_outlines_api.serializers.StoryElementNodeSerializer.Meta attribute), 15
- fields_required_for_add (fiction_outlines_api.mixins.NodeAddMixin attribute), 10
- fields_required_for_add (fiction_outlines_api.views.ArcNodeCreateView attribute), 16
- fields_required_for_add (fiction_outlines_api.views.StoryNodeCreateView attribute), 25
- ## G
- get_arc_structure() (fiction_outlines_api.serializers.ArcSerializer method), 12
- get_object() (fiction_outlines_api.mixins.NodeMoveMixin method), 11
- get_object_permission_queryset() (fiction_outlines_api.mixins.MultiObjectPermissionsMixin method), 10
- get_outline_structure() (fiction_outlines_api.serializers.OutlineSerializer method), 14
- get_permission_object() (fiction_outlines_api.mixins.MultiObjectPermissionsMixin method), 10
- get_queryset() (fiction_outlines_api.views.ArcDetailView method), 16
- get_queryset() (fiction_outlines_api.views.ArcNodeCreateView method), 17
- get_queryset() (fiction_outlines_api.views.ArcNodeDetailView method), 17
- get_queryset() (fiction_outlines_api.views.ArcNodeMoveView method), 18
- get_queryset() (fiction_outlines_api.views.CharacterDetail method), 18
- get_queryset() (fiction_outlines_api.views.CharacterInstanceDetailView method), 20
- get_queryset() (fiction_outlines_api.views.CharacterList method), 20
- get_queryset() (fiction_outlines_api.views.LocationDetail method), 21
- get_queryset() (fiction_outlines_api.views.LocationInstanceDetailView method), 22
- get_queryset() (fiction_outlines_api.views.LocationList method), 22
- get_queryset() (fiction_outlines_api.views.OutlineDetail method), 23
- get_queryset() (fiction_outlines_api.views.OutlineList method), 23
- get_queryset() (fiction_outlines_api.views.SeriesDetail method), 24
- get_queryset() (fiction_outlines_api.views.SeriesList method), 25
- get_queryset() (fiction_outlines_api.views.StoryNodeCreateView method), 25
- get_queryset() (fiction_outlines_api.views.StoryNodeDetailView method), 26
- get_queryset() (fiction_outlines_api.views.StoryNodeMoveView method), 26
- ## L
- LocationDetail (class in fiction_outlines_api.views), 20
- LocationInstanceCreateView (class in fiction_outlines_api.views), 21
- LocationInstanceDetailView (class in fiction_outlines_api.views), 21
- LocationInstanceSerializer (class in fiction_outlines_api.serializers), 13
- LocationInstanceSerializer.Meta (class in fiction_outlines_api.serializers), 13

| | | |
|--|--|--|
| attribute), 26 | permission_classes | (fiction_outlines_api.views.CharacterDetail attribute), 19 |
| OutlineDetail (class in fiction_outlines_api.views), 23 | permission_classes | (fiction_outlines_api.views.CharacterList attribute), 20 |
| OutlineList (class in fiction_outlines_api.views), 23 | permission_classes | (fiction_outlines_api.views.LocationDetail attribute), 21 |
| OutlineSerializer (class in fiction_outlines_api.serializers), 14 | permission_classes | (fiction_outlines_api.views.LocationList attribute), 22 |
| OutlineSerializer.Meta (class in fiction_outlines_api.serializers), 14 | permission_classes | (fiction_outlines_api.views.OutlineDetail attribute), 23 |
| P | permission_classes | (fiction_outlines_api.views.OutlineList attribute), 24 |
| patch() (fiction_outlines_api.views.ArcDetailView method), 16 | permission_classes | (fiction_outlines_api.views.SeriesDetail attribute), 24 |
| patch() (fiction_outlines_api.views.ArcNodeDetailView method), 17 | permission_classes (fiction_outlines_api.views.SeriesList attribute), 25 | |
| patch() (fiction_outlines_api.views.CharacterDetail method), 19 | permission_classes | (fiction_outlines_api.views.StoryNodeDetailView attribute), 26 |
| patch() (fiction_outlines_api.views.CharacterInstanceDetailView method), 20 | permission_classes | (fiction_outlines_api.views.StoryNodeDetailView attribute), 26 |
| patch() (fiction_outlines_api.views.LocationDetail method), 21 | permission_required | (fiction_outlines_api.views.ArcCreateView attribute), 16 |
| patch() (fiction_outlines_api.views.OutlineDetail method), 23 | permission_required | (fiction_outlines_api.views.ArcDetailView attribute), 16 |
| patch() (fiction_outlines_api.views.StoryNodeDetailView method), 26 | permission_required | (fiction_outlines_api.views.ArcNodeCreateView attribute), 17 |
| perform_create() (fiction_outlines_api.mixins.NodeAddMixin method), 10 | permission_required | (fiction_outlines_api.views.ArcNodeDetailView attribute), 18 |
| perform_create() (fiction_outlines_api.views.ArcCreateView method), 16 | permission_required | (fiction_outlines_api.views.ArcNodeMoveView attribute), 18 |
| perform_create() (fiction_outlines_api.views.CharacterInstanceCreateView method), 19 | permission_required | (fiction_outlines_api.views.CharacterDetail attribute), 19 |
| perform_create() (fiction_outlines_api.views.CharacterList method), 20 | permission_required | (fiction_outlines_api.views.CharacterInstanceCreateView attribute), 19 |
| perform_create() (fiction_outlines_api.views.LocationInstanceCreateView method), 21 | permission_required | (fiction_outlines_api.views.CharacterInstanceDetailView attribute), 20 |
| perform_create() (fiction_outlines_api.views.LocationList method), 22 | permission_required | (fiction_outlines_api.views.LocationDetail attribute), 21 |
| perform_create() (fiction_outlines_api.views.OutlineList method), 24 | permission_required | (fiction_outlines_api.views.LocationList attribute), 22 |
| perform_create() (fiction_outlines_api.views.SeriesList method), 25 | permission_required | (fiction_outlines_api.views.OutlineDetail attribute), 23 |
| perform_move() (fiction_outlines_api.mixins.NodeMoveMixin method), 11 | permission_required | (fiction_outlines_api.views.OutlineList attribute), 24 |
| perform_update() (fiction_outlines_api.views.CharacterDetail method), 19 | permission_required | (fiction_outlines_api.views.SeriesList attribute), 25 |
| perform_update() (fiction_outlines_api.views.LocationDetail method), 21 | permission_required | (fiction_outlines_api.views.StoryNodeDetailView attribute), 26 |
| perform_update() (fiction_outlines_api.views.OutlineDetail method), 23 | permission_required | (fiction_outlines_api.views.StoryNodeDetailView attribute), 26 |
| permission_classes | permission_required | (fiction_outlines_api.views.ArcCreateView attribute), 16 |
| permission_classes | permission_required | (fiction_outlines_api.views.ArcDetailView attribute), 16 |
| permission_classes | permission_required | (fiction_outlines_api.views.ArcNodeCreateView attribute), 17 |
| permission_classes | permission_required | (fiction_outlines_api.views.ArcNodeDetailView attribute), 18 |
| permission_classes | permission_required | (fiction_outlines_api.views.ArcNodeMoveView attribute), 18 |
| permission_classes | permission_required | (fiction_outlines_api.views.CharacterDetail attribute), 19 |
| permission_classes | permission_required | (fiction_outlines_api.views.CharacterInstanceCreateView attribute), 19 |
| permission_classes | permission_required | (fiction_outlines_api.views.CharacterInstanceDetailView attribute), 20 |
| permission_classes | permission_required | (fiction_outlines_api.views.LocationDetail attribute), 21 |
| permission_classes | permission_required | (fiction_outlines_api.views.LocationList attribute), 22 |
| permission_classes | permission_required | (fiction_outlines_api.views.OutlineDetail attribute), 23 |
| permission_classes | permission_required | (fiction_outlines_api.views.OutlineList attribute), 24 |
| permission_classes | permission_required | (fiction_outlines_api.views.SeriesList attribute), 25 |
| permission_classes | permission_required | (fiction_outlines_api.views.StoryNodeDetailView attribute), 26 |

tion_outlines_api.views.LocationInstanceCreateView (fiction_outlines_api.serializers.CharacterSerializer.Meta attribute), 21
 permission_required (fiction_outlines_api.views.LocationInstanceDetailView attribute), 22
 permission_required (fiction_outlines_api.views.OutlineDetailView attribute), 23
 permission_required (fiction_outlines_api.views.SeriesDetailView attribute), 24
 permission_required (fiction_outlines_api.views.StoryNodeCreateView attribute), 25
 permission_required (fiction_outlines_api.views.StoryNodeDetailView attribute), 26
 permission_required (fiction_outlines_api.views.StoryNodeMoveView attribute), 26
 pos (fiction_outlines_api.mixins.NodeMoveMixin attribute), 11
 post() (fiction_outlines_api.mixins.MultiObjectPermissionsMixin method), 10
 post() (fiction_outlines_api.mixins.NodeAddMixin method), 11
 post() (fiction_outlines_api.mixins.NodeMoveMixin method), 11
 post() (fiction_outlines_api.views.ArcCreateView method), 16
 put() (fiction_outlines_api.views.ArcDetailView method), 16
 put() (fiction_outlines_api.views.ArcNodeDetailView method), 18
 put() (fiction_outlines_api.views.CharacterDetailView method), 19
 put() (fiction_outlines_api.views.CharacterInstanceDetailView method), 20
 put() (fiction_outlines_api.views.LocationDetailView method), 21
 put() (fiction_outlines_api.views.OutlineDetailView method), 23
 put() (fiction_outlines_api.views.SeriesDetailView method), 24
 put() (fiction_outlines_api.views.StoryNodeDetailView method), 26
 read_only_fields (fiction_outlines_api.serializers.ArcElementNodeSerializer.Meta attribute), 12
 read_only_fields (fiction_outlines_api.serializers.ArcSerializer.Meta attribute), 12
 read_only_fields (fiction_outlines_api.serializers.CharacterInstanceSerializer.Meta attribute), 13
 read_only_fields (fiction_outlines_api.serializers.CharacterSerializer.Meta attribute), 13
 read_only_fields (fiction_outlines_api.serializers.LocationInstanceSerializer attribute), 14
 read_only_fields (fiction_outlines_api.serializers.LocationSerializer.Meta attribute), 14
 read_only_fields (fiction_outlines_api.serializers.OutlineSerializer.Meta attribute), 14
 read_only_fields (fiction_outlines_api.serializers.SeriesSerializer.Meta attribute), 15
 read_only_fields (fiction_outlines_api.serializers.StoryElementNodeSerializer attribute), 15
 related_key (fiction_outlines_api.mixins.NodeMoveMixin attribute), 11
 related_key (fiction_outlines_api.views.ArcNodeMoveView attribute), 18
 related_key (fiction_outlines_api.views.StoryNodeMoveView attribute), 27
 S
 serializer_class (fiction_outlines_api.views.ArcCreateView attribute), 16
 serializer_class (fiction_outlines_api.views.ArcDetailView attribute), 16
 serializer_class (fiction_outlines_api.views.ArcNodeCreateView attribute), 17
 serializer_class (fiction_outlines_api.views.ArcNodeDetailView attribute), 18
 serializer_class (fiction_outlines_api.views.ArcNodeMoveView attribute), 18
 serializer_class (fiction_outlines_api.views.CharacterDetailView attribute), 19
 serializer_class (fiction_outlines_api.views.CharacterInstanceCreateView attribute), 19
 serializer_class (fiction_outlines_api.views.CharacterInstanceDetailView attribute), 20
 serializer_class (fiction_outlines_api.views.CharacterList attribute), 20
 serializer_class (fiction_outlines_api.views.LocationDetailView attribute), 21
 serializer_class (fiction_outlines_api.views.LocationInstanceCreateView attribute), 21
 serializer_class (fiction_outlines_api.views.LocationInstanceDetailView attribute), 22
 serializer_class (fiction_outlines_api.views.LocationList attribute), 22
 serializer_class (fiction_outlines_api.views.OutlineDetailView attribute), 23
 serializer_class (fiction_outlines_api.views.OutlineList attribute), 24
 serializer_class (fiction_outlines_api.views.SeriesDetailView attribute), 24
 serializer_class (fiction_outlines_api.views.SeriesList attribute), 25

serializer_class (fiction_outlines_api.views.StoryNodeCreateView attribute), [25](#)
serializer_class (fiction_outlines_api.views.StoryNodeDetailView attribute), [26](#)
serializer_class (fiction_outlines_api.views.StoryNodeMoveView attribute), [27](#)
SeriesDetail (class in fiction_outlines_api.views), [24](#)
SeriesList (class in fiction_outlines_api.views), [24](#)
SeriesSerializer (class in fiction_outlines_api.serializers), [14](#)
SeriesSerializer.Meta (class in fiction_outlines_api.serializers), [15](#)
ServiceUnavailable, [9](#)
source_node (fiction_outlines_api.mixins.NodeMoveMixin attribute), [11](#)
status_code (fiction_outlines_api.exceptions.ServiceUnavailable attribute), [9](#)
status_code (fiction_outlines_api.exceptions.TreeUnavailable attribute), [9](#)
StoryElementNodeSerializer (class in fiction_outlines_api.serializers), [15](#)
StoryElementNodeSerializer.Meta (class in fiction_outlines_api.serializers), [15](#)
StoryNodeCreateView (class in fiction_outlines_api.views), [25](#)
StoryNodeDetailView (class in fiction_outlines_api.views), [25](#)
StoryNodeMoveView (class in fiction_outlines_api.views), [26](#)

T

target_node (fiction_outlines_api.mixins.NodeMoveMixin attribute), [11](#)
target_node_type_fieldname (fiction_outlines_api.mixins.NodeMoveMixin attribute), [11](#)
target_node_type_fieldname (fiction_outlines_api.views.ArcNodeMoveView attribute), [18](#)
target_node_type_fieldname (fiction_outlines_api.views.StoryNodeMoveView attribute), [27](#)
TreeUnavailable, [9](#)

U

update() (fiction_outlines_api.views.ArcNodeDetailView method), [18](#)
update() (fiction_outlines_api.views.StoryNodeDetailView method), [26](#)

V

verbose_name (fiction_outlines_api.apps.FictionOutlinesApiConfig attribute), [9](#)