
Django Fiction Outlines Documentation

Release 0.4.0

Daniel Andrlik

Apr 02, 2022

Contents

| | | |
|-----------|---|-----------|
| 1 | Introduction | 3 |
| 1.1 | Documentation | 3 |
| 1.2 | Code Repo and Issue Tracker | 3 |
| 1.3 | License | 3 |
| 1.4 | Features | 3 |
| 1.5 | What It Doesn't Do | 4 |
| 1.6 | Running Tests | 4 |
| 1.7 | Credits | 4 |
| 2 | Installation | 5 |
| 2.1 | Quickstart | 5 |
| 3 | Tutorial | 7 |
| 3.1 | Concepts | 7 |
| 3.2 | Intro to <code>fiction_outlines</code> Models | 8 |
| 3.3 | Usage | 9 |
| 4 | API | 11 |
| 4.1 | Models | 11 |
| 5 | Caveats | 19 |
| 6 | Signals | 21 |
| 6.1 | <code>tree_manipulation</code> | 21 |
| 7 | Receivers | 23 |
| 8 | Views | 25 |
| 9 | Forms | 31 |
| 10 | <code>fiction_outlines</code> | 33 |
| 10.1 | <code>fiction_outlines</code> package | 33 |
| 11 | Contributing | 77 |
| 11.1 | Types of Contributions | 77 |
| 11.2 | Get Started! | 78 |
| 11.3 | Pull Request Guidelines | 79 |

| | | |
|-----------|---|-----------|
| 11.4 | Appropriate Conduct | 79 |
| 12 | Contributor Covenant Code of Conduct | 81 |
| 12.1 | Our Pledge | 81 |
| 12.2 | Our Standards | 81 |
| 12.3 | Our Responsibilities | 82 |
| 12.4 | Scope | 82 |
| 12.5 | Enforcement | 82 |
| 12.6 | Attribution | 82 |
| 13 | BSD License | 83 |
| 14 | Credits | 85 |
| 14.1 | Development Lead | 85 |
| 14.2 | Contributors | 85 |
| 15 | History | 87 |
| 15.1 | 0.4.0 (2022-03-17) | 87 |
| 15.2 | 0.3.1 (2018-10-16) | 87 |
| 15.3 | 0.3.0 (2018-08-08) | 87 |
| 15.4 | 0.2.2 (2018-04-19) | 87 |
| 15.5 | 0.2.1 (2018-04-14) | 87 |
| 15.6 | 0.2.0 (2018-04-13) | 88 |
| 15.7 | 0.1.5 (2018-04-09) | 88 |
| 15.8 | 0.1.4 (2018-04-07) | 88 |
| 15.9 | 0.1.3 (2018-04-07) | 88 |
| 15.10 | 0.1.2 (2018-04-02) | 88 |
| 15.11 | 0.1.1 (2018-04-01) | 88 |
| | Python Module Index | 89 |
| | Index | 91 |

Contents:

Welcome to Django Fiction Outlines!

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.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 models for managing series, outlines, characters, locations, and arcs.
- Provides tools for managing multiple arcs within the context of a broader story outline.
- Validates that arcs and outlines follow principles of MACE nesting, and seven point story structure.
- Calculates estimated length of final manuscript based on complexity of outline.
- Objects are associated with users to enable permission management.

- Export outlines to OPML, JSON, or Markdown documents.
 - NOTE: Django Fiction Outlines 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.

1.5 What It Doesn't Do

- Provide a full UI for managing the changes. An API and views are provided, but templates are very basic. It is expected that you will override the templates to match your overall project.
- Outline the whole story for you.
- Write the story for you.
- Do your laundry.

1.6 Running Tests

Does the code actually work?

```
$ pip install -r test_requirements.txt
$ pytest
$ pytest --flake8
```

1.7 Credits

Tools used in rendering this package:

- [Cookiecutter](#)
- [cookiecutter-djangopackage](#)

CHAPTER 2

Installation

At the command line:

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

Or, if you have virtualenvwrapper installed:

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

2.1 Quickstart

Install Django Fiction Outlines:

```
pip install django-fiction-outlines
```

Add it and dependencies to your *INSTALLED_APPS*:

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

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's URL patterns:

```
from fiction_outlines import urls as fiction_outlines_urls

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

When working with Django fiction outlines, it is important to understand that it is an opinionated library, and leans heavily on three concepts.

3.1 Concepts

3.1.1 M.A.C.E. Quotient

The principle that each story thread (or arc) is one of four primary types.

- Milieu: About the journey through a place.
- Answers: A question or mystery that must be answered.
- Character: The journey and change within a character for better or worse.
- Event: An external event that must be dealt with.

This is author [Mary Robinette Kowal](#)'s¹ version of Orson Scott Card's original [M.I.C.E. quotient](#), and I think her version is easier to follow.

3.1.2 Nesting

Again, [from Kowal](#), that story threads, regardless of MACE type, should be resolved in the opposite order in which they were introduced. Last in, first out.²

¹ This project is neither associated with, nor endorsed by, Mary Robinette Kowal or Orson Scott Card.

² Savvy watchers of this lecture will note that `fiction_outlines` also draws its formula for estimating overall manuscript length from Kowal as well.

3.1.3 Seven Point Story Structure

This one comes from author Dan Wells,³ whose [talk on story structure](#) has helped countless writers out there. Essentially, a well-crafted arc consists of seven milestones:

1. Hook: the starting state.
2. Plot Turn 1: What changes that starts this part of the story?
3. Pinch 1: The first **major** road block faced along the path.
4. Midpoint: The halfway mark.
5. Pinch 2: The great challenge. All seems lost.
6. Plot Turn 2: What change or realization occurs to make the resolution possible.
7. Resolution: Completion of the arc. Should be the opposite state of the hook.

Try/Fail cycles should be inserted in-between milestones to direct the pacing and ensure the story earns its milestone moments.

3.2 Intro to `fiction_outlines` Models

3.2.1 First Tier Models

There are four elements in `fiction_outlines` from which everything else descends. Those are:

1. *Outline*: The actual outline container object.
2. *Series*: A collection of related outlines and other objects.
3. *Character*: A single character which can in turn be used in multiple series and outlines.
4. *Location*: Settings/locations which can in turn be used in multiple series and outlines.

The purpose of each should be relatively clear.

3.2.2 Second Tier Models

1. *CharacterInstance*: A related record for a character with an individual outline. Contains additional metadata as to the character's role in the outline.
2. *LocationInstance*: Same as a *CharacterInstance*, but for *Location*.
3. *StoryElementNode*: This model makes up the actual outline elements for the story. It descends from the outline, and represents the structure of the story using a materialized path tree.
4. *Arc*: A story arc, associated with a single MACE type. An outline can have 1 to n arcs. For example, a short story may only have one arc, but a novel will have many. An arc is expected to conform to seven point story structure, and its default state will consist of those milestones.
 - a. *ArcElementNode*: This model represents the nodes of the materialized path tree describing all the points of the arc. One or more character or location instances may be associated with each node. In turn, an arc element node can be associated with a *StoryElementNode* allowing the outliner to visualize the overall story structure of the entire outline.

³ This project is neither associated with, nor endorsed by, Dan Wells.

3.3 Usage

Let's say you want to represent a user who is outlining a new series. We'll call them `user1`.

```
series = Series(
    title='My new franchise',
    description='This is gonna be the next Harry Potter, I just know it.',
    tags='urban fantasy, high hopes',
    user=user1
)
series.save()
my_outline = Outline(
    title='It begins',
    description='A twenty-something discovers that they are the chosen one_
↳to defend the city against all harm.',
    tags='heroine, fae',
    user=user1
)
my_outline.save()
# You now have the series and outline, and can proceed to add arcs or start working_
↳at the overall plot level.
main_arc = my_outline.create_arc(name='Chosen One', mace_type='character',_
↳description='Coming into her own')
# The above command, creates the arc instance and also generates the initial skeleton_
↳of the arc using seven
# point story structure.

# Let's add a character.
samantha = Character(
    name='Samantha Cowler',
    description='A cynical and disaffected young woman destined to be a hero
↳',
    tags='heroine',
    user=user1
)
samantha.save()
samantha_first_book = CharacterInstance(character=samantha, outline=my_outline, pov_
↳character=True, protagonist=True, main_character=True)
samantha_first_book.save()
# Add a location
sam_job = Location(name='The Damn Bar', description='The tavern where Samantha works.
↳', tags='human, normality', user=user1)
sam_job.save()
sam_job_first_book = LocationInstance(location=sam_job, outline=my_outline)
sam_job_first_book.save()
# Want to fetch the arc or story structure?
arc_tree = arc.arc_root_node.get_descendants()
story_tree = my_outline.story_tree_root.get_descendants()
```

For more detail on how to work with these objects, please review the `apiref`.

NOTE: It is almost always better to use `fiction_outlines`' provided views as opposed to manually manipulating the models. The views make working with the objects less complex, and also provide an object-level security model. If you must work with them directly, it is recommended that you subclass the view itself and make your modifications there.

4.1 Models

There are two types of models used in Fiction Outlines, Standard and Tree.

4.1.1 Standard

Standard models are rather typical Django models and so the API is much as you would expect.

```
class fiction_outlines.models.Series(*args, **kwargs)
```

Bases: *fiction_outlines.models.TimeStampedModel*

Container object to hold multiple outline objects if necessary.

Parameters

- **created** (*AutoCreatedField*) – Created
- **modified** (*AutoLastModifiedField*) – Modified
- **id** (*UUIDField*) – Id
- **title** (*CharField*) – Name of the series. You can always change this later.
- **description** (*TextField*) – Jot down a description about your series.
- **user_id** (*ForeignKey*) – The user that created this Series.

This model represents a story series that can hold one or more outlines within it. it is not necessary to define a Series, but it is a useful organizational tool. Standard Django ORM applies.

```
class fiction_outlines.models.Character(*args, **kwargs)
```

Bases: *fiction_outlines.models.TimeStampedModel*

Reusable character definition model.

Parameters

- **created** (*AutoCreatedField*) – Created
- **modified** (*AutoLastModifiedField*) – Modified
- **id** (*UUIDField*) – Id
- **name** (*CharField*) – Name of the character.
- **description** (*TextField*) – Notes about the character to help you remember.
- **user_id** (*ForeignKey*) – The user that created this character.

This model represents a character that may be reused in multiple series, outlines, and story elements.

class `fiction_outlines.models.CharacterInstance` (*args, **kwargs)

Bases: `fiction_outlines.models.TimeStampedModel`

An instance of the character object that can be associated with outlines.

Parameters

- **created** (*AutoCreatedField*) – Created
- **modified** (*AutoLastModifiedField*) – Modified
- **id** (*UUIDField*) – Id
- **main_character** (*BooleanField*) – Is this character the main character for the outline?
- **pov_character** (*BooleanField*) – Is this character a POV character?
- **protagonist** (*BooleanField*) – Does this character serve as the protagonist for this outline?
- **antagonist** (*BooleanField*) – Does this character serve as an antagonist for this outline?
- **obstacle** (*BooleanField*) – Is this character an obstacle in the outline? (not antagonist)
- **villain** (*BooleanField*) – Is the character a straight-out villain?
- **character_id** (*ForeignKey*) – Reference to originating character object.
- **outline_id** (*ForeignKey*) – Outline this instance is associated with.

This model represents a single instance of a character that is associated with an Outline. Contains additional metadata on the character's role in the story.

class `fiction_outlines.models.Location` (*args, **kwargs)

Bases: `fiction_outlines.models.TimeStampedModel`

Reusable location definition model

Parameters

- **created** (*AutoCreatedField*) – Created
- **modified** (*AutoLastModifiedField*) – Modified
- **id** (*UUIDField*) – Id
- **name** (*CharField*) – Name of the location.
- **description** (*TextField*) – Notes about the location to help you remember.
- **user_id** (*ForeignKey*) – The user that created this location.

This model represents a location that may be reused in multiple series, outlines, and story elements.

```
class fiction_outlines.models.LocationInstance (*args, **kwargs)
    Bases: fiction_outlines.models.TimeStampedModel
```

An instance of the given location that can be associated with a given outline.

Parameters

- **created** (*AutoCreatedField*) – Created
- **modified** (*AutoLastModifiedField*) – Modified
- **id** (*UUIDField*) – Id
- **location_id** (*ForeignKey*) – Originating location object.
- **outline_id** (*ForeignKey*) – Outline this object is associated with.

This model represents an instance of a location that is associated with an outline.

```
class fiction_outlines.models.Outline (*args, **kwargs)
    Bases: fiction_outlines.models.TimeStampedModel
```

The typical top of the hierarchy when not enclosed in a series.

Parameters

- **created** (*AutoCreatedField*) – Created
- **modified** (*AutoLastModifiedField*) – Modified
- **id** (*UUIDField*) – Id
- **title** (*CharField*) – Outline title. You can always change this later.
- **description** (*TextField*) – Optionally, describe the story. Or use for notes.
- **series_id** (*ForeignKey*) – Belongs to series.
- **user_id** (*ForeignKey*) – The user that created this outline.

The outline is the manuscript level element that represents and encapsulates all the information about a specific work. It provides a number of convenient convenience features.

`Outline.length_estimate()`

Calculates and estimated word count based on number of characters, locations, and arcs. For reference see: <http://www.writingexcuses.com/2017/07/02/12-27-choosing-a-length/>

This is a cached property that calculates the projected total length of the manuscript.

Example:

```
o1.length_estimate
# Returns estimated total words.
```

`Outline.story_tree_root()`

Fetches the root node for the outline's StoryElementNode tree.

Cached property that returns the root of the outline tree.

Example:

```
root_node = o1.story_tree_root
```

`Outline.refresh_from_db(*args, **kwargs)`

Reload field values from the database.

By default, the reloading happens from the database this instance was loaded from, or by the read router if this instance wasn't loaded from any database. The `using` parameter will override the default.

Fields can be used to specify which fields to reload. The fields should be an iterable of field attnames. If `fields` is `None`, then all non-deferred fields are reloaded.

When accessing deferred fields of an instance, the deferred loading of the field will call this method.

Just like Django's `refresh_from_db()` except that this clears the property cache of the object as well.

`Outline.create_arc(mace_type, name)`

Creates the story arc and initial tree for that arc for the current outline. Returns the resulting `Arc` instance.

Creates an `Arc` object within the outline, and builds the initial tree of `ArcElementNode` objects. Returns the `Arc` object.

Example:

```
arc1 = ol.create_arc(mace_type='event', name='Dragon Invasion')
```

`Outline.validate_nesting()`

Reviews the story tree and validates associated arc elements are nested appropriately. Returns a dict of errors.

Evaluates the tree of `StoryElementNode` objects and returns a dict of errors if any are found. For each error entry, a list of offending nodes will also be included.

Example:

```
error_dict = ol.validate_nesting()
if error_dict:
    # There are errors
    for key, value in error_dict.items():
        print("%s: %s" % (key, value))
```

`class fiction_outlines.models.Arc(*args, **kwargs)`

Bases: `fiction_outlines.models.TimeStampedModel`

A MACE arc for a outline.

Parameters

- **created** (`AutoCreatedField`) – Created
- **modified** (`AutoLastModifiedField`) – Modified
- **id** (`UUIDField`) – Id
- **mace_type** (`CharField`) – The MACE type of the Arc.
- **outline_id** (`ForeignKey`) – Arc belongs to this outline.
- **name** (`CharField`) – Name of this Arc (makes it easier for you to keep track of it.)

The arc represents a story throughline that will be integrated with the outline.

`Arc.current_errors()`

Returns list of errors from `arc_validation`.

A cached property of current structural errors within the Arc.

`Arc.arc_root_node()`

Returns the root node from this object's `ArcElementNode` tree.

A cached property pointing to the root node of the *ArcElementNode* object tree.

`Arc.refresh_from_db(*args, **kwargs)`

Reload field values from the database.

By default, the reloading happens from the database this instance was loaded from, or by the read router if this instance wasn't loaded from any database. The `using` parameter will override the default.

Fields can be used to specify which fields to reload. The fields should be an iterable of field atnames. If fields is `None`, then all non-deferred fields are reloaded.

When accessing deferred fields of an instance, the deferred loading of the field will call this method.

Like the standard Django method, but this also clears cached properties.

`Arc.generate_template_arc_tree()`

Generate a seven point template in this arc. Arc must be empty.

Creates the template arc tree using *Seven Point Story Structure*.

`Arc.fetch_arc_errors()`

Evaluates the current tree of the arc and provides a list of errors that the user should correct.

Evaluates the arc tree for errors the user is recommended to correct.

`Arc.validate_first_element()`

Ensures that the first node for the direct decendents of root is the hook.

Checks that the first child of the root is the Hook.

`Arc.validate_last_element()`

Ensures that the last element of the arc is the resolution.

Checks that the last child of the root is the Resolution.

`Arc.validate_generations()`

Make sure that the descendent depth is valid.

Reviews the structure of the Arc to ensure that node types follow the *allowed_parents* and *allowed_children* properties.

`Arc.validate_milestones()`

Reviews the arc element tree to ensure that milestones appear in the right order.

Verifies that milestones appear in the tree in the Arc tree in the correct sequence.

4.1.2 Tree Models

Tree models are Materialized Path trees descended from the *django-treebeard* provided *MP_Node*.

Note: A discussion of the details of working with MP trees is out of scope for this document. You are recommended to peruse *django-treebeard's excellent documentation*. Make sure to also review *fiction_outlines Caveats* documentation.

class `fiction_outlines.models.ArcElementNode(*args, **kwargs)`

Bases: *fiction_outlines.models.TimeStampedModel*, *treebeard.mp_tree.MP_Node*

Tree nodes for the arc elements.

Parameters

- **created** (*AutoCreatedField*) – Created
- **path** (*CharField*) – Path
- **depth** (*PositiveIntegerField*) – Depth
- **numchild** (*PositiveIntegerField*) – Numchild
- **modified** (*AutoLastModifiedField*) – Modified
- **id** (*UUIDField*) – Id
- **arc_element_type** (*CharField*) – What part of the arc does this represent?
- **arc_id** (*ForeignKey*) – Parent arc.
- **headline** (*CharField*) – Autogenerated from description
- **description** (*TextField*) – Describe what happens at this moment in the story...
- **story_element_node_id** (*ForeignKey*) – Which story node is this element associated with?

This model represents the nodes of the tree that is used as the structure of the *Arc*.

`ArcElementNode.milestone_seq()`

Returns the milestone sequence based off of the arc element definitions.

Cached property retrieving the derived milestone sequence number as it relates to **7PSS_**.

`ArcElementNode.is_milestone()`

Does this node represent an arc milestone?

Cached property returning if this node represents an arc milestone.

`ArcElementNode.parent_outline()`

Private method to fetch parent outline.

Cached property for convenient access to the outline to which this arc tree belongs.

`ArcElementNode.move(target, pos=None)`

Moves the current node and all it's descendants to a new position relative to another node.

Raises PathOverflow – when the library can't make room for the node's new position

Subclass of the `treebeard` method. Fires a *tree_manipulation* signal for your use.

`ArcElementNode.add_child(arc_element_type, description=None, story_element_node=None, **kwargs)`

Overrides the default *treebeard* function, adding additional integrity checks.

Subclasses the `treebeard` method to add required logic for instantiating an Arc Element object.

Example:

```
# Add a try/fail cycle
new_element = ael.add_child('tf', description="Attempting to get into the secret_
↪enclave to get information")
```

`ArcElementNode.add_sibling(pos=None, arc_element_type=None, description=None, story_element_node=None, **kwargs)`

Overrides the default *treebeard* function, adding additional integrity checks.

Subclasses the `treebeard` method to add specific model instantiation requirements.

Example:

```
# Add another beat that followed after another one
beat2 = beat1.add_sibling('beat', description="John discovers an odd item in his_
↪bag.")
```

class `fiction_outlines.models.StoryElementNode` (*args, **kwargs)
 Bases: `fiction_outlines.models.TimeStampedModel`, `treebeard.mp_tree.MP_Node`

Tree nodes for the overall outline of the story.

Parameters

- **created** (`AutoCreatedField`) – Created
- **path** (`CharField`) – Path
- **depth** (`PositiveIntegerField`) – Depth
- **numchild** (`PositiveIntegerField`) – Numchild
- **modified** (`AutoLastModifiedField`) – Modified
- **id** (`UUIDField`) – Id
- **name** (`CharField`) – Optional name/title for this element of the story.
- **description** (`TextField`) – Optional description for this element of the story.
- **outline_id** (`ForeignKey`) – Parent outline.
- **story_element_type** (`CharField`) – What part of the story does this represent? A scene? A chapter?

This class represent the actual structure of the overall outline. Individual arc elements can be associated with a story node, which is how the outline validation tool can verify that *Nesting* is valid.

`StoryElementNode.all_characters()`

Returns a queryset of all characters associated with this node and its descendants, excluding any duplicates.

A property that returns queryset of all the unique character instances associated with this node, and any of its descendant nodes.

`StoryElementNode.all_locations()`

Returns a queryset of all locations associated with this node and its descendants, excluding any duplicates.

A property that returns a queryset of all the unique location instances associated with this node, and any of its descendant nodes.

`StoryElementNode.impact_rating()`

Returns the impact rating for this node. Impact rating is a measure of how powerful this moment in the story is by evaluating how many simultaneous arc elements are associated with it. There is also a generational bleed element, where the impact score creates shockwaves throughout their direct ancestor and descendant nodes. This echo fades fast, but the bigger the impact, the farther it goes.

Currently, the impact bleed does not extend to sibling nodes.

WARNING: Here be dragons.

A property representing the impact/tension rating of this node (expressed as a *float*) in the outline. This rating is derived from associations with arc elements, with extra impact when multiple arcs overlap in the same node. Impact also affects ancestor and descendant nodes with weakening influence the more generations away from the source node. However, impact bleed does not extend to sibling nodes.

`StoryElementNode.move(target, pos=None)`

An override of the treebeard api in order to send a signal in advance.

Subclass of the `treebeard` move method, but also sends signal `tree_manipulation` which you can use with your signal receivers.

`StoryElementNode.add_child`(*story_element_type=None, outline=None, name=None, description=None, **kwargs*)

An override of the `treebeard` `add_child()` method so we can send a signal.

Subclass of the `treebeard` method, but adds instantiation logic for this model.

Example:

```
new_node = node.add_child(story_element_type='chapter', outline=01, name='Chapter_↵
↵1', description='Our story begins.')
```

`StoryElementNode.add_sibling`(*story_element_type=None, outline=None, name=None, description=None, pos=None, **kwargs*)

Override of `treebeard` api to allow us to send a signal.

Subclass of the `treebeard` method, but adds model instantiation logic.

Example:

```
chap2 = new_node.add_sibling(story_element_type='chapter', outline=01, name=
↵'Chapter 2', description='Meanwhile, on the other side of the world')
```

Caveats

Be aware that as tree models are descendents of the `django-treebeard MP_Node` class, the same [Known Caveats](#) apply.

Warning:

- 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.

6.1 tree_manipulation

Fires off a signal on tree manipulation, e.g. a `move()` method. Sends the following:

| Variable | Description | Allowed values |
|------------------|--|---|
| action | What tree manipulation method was called? | add_child add_sibling move update |
| target_node_type | Class of the target node. | If a <code>StoryElementNode</code> , this will be the value of <code>story_element_type</code> . If an <code>ArcElementNode</code> , it will be the value of <code>arc_element_type</code> . |
| target_node | The node to which this is being moved in relation to. | If this is a move action, this will be populated with the target node for the move. |
| pos | Position relative to the target_node that this cell should be added to. Only populated when action is equal to move. | left right first-child last-child first-sibling last-sibling |

CHAPTER 7

Receivers

The following functions are currently tied to the signals generated in `fiction_outlines`. See [Signals](#) for additional information.

`receivers.generate_headline_from_description(instance, *args, **kwargs)`

Auto generate the headline of the node from the first lines of the description.

`receivers.story_root_for_new_outline(instance, created, *args, **kwargs)`

If a new instance of a `Outline` is created, also create the root node of the story tree.

`receivers.arc_node_edit_add_missing_characters_and_locations_to_related_story_node(instance, action, reverse, pk_set, *args, **kwargs)`

If an `arc_element` is modified and it's characters/locations are not already in the story node, add them. We don't assume that removing the arc element would change the characters or locations as of yet. This takes up a little more space in the database, but the additional flexibility for users is worth it.

`receivers.story_node_add_arc_element_update_characters_locations(instance, created, *args, **kwargs)`

If an arc element is added to a story element node, add any missing elements or locations.

`receivers.validate_arc_links_same_outline(instance, *args, **kwargs)`

Evaluates attempts to link an arc to a story node from another outline.

`receivers.validate_character_instance_valid_for_arc(instance, action, reverse, pk_set, *args, **kwargs)`

Evaluate attempts to assign a character instance to ensure it is from same outline.

`receivers.validate_location_instance_valid_for_arc(instance, action, reverse, pk_set, *args, **kwargs)`

Evaluates attempts to add location instances to arc, ensuring they are from same outline.

```
receivers.validate_character_for_story_element(instance, action, reverse, pk_set, *args,  
                                              **kwargs)
```

Validates that character is from the same outline as the story node.

```
receivers.validate_location_for_story_element(instance, action, reverse, pk_set, *args,  
                                              **kwargs)
```

Validates that location is from same outline as story node.

```
receivers.validate_generations_for_story_elements(instance,          action,          tar-  
                                                  get_node_type=None,          tar-  
                                                  get_node=None, pos=None, *args,  
                                                  **kwargs)
```

Unlike arc nodes, for which we just warn about structure, the story tree allowed parent/child rules must be strictly enforced.

Views are provided for the majority of common tasks when working with `fiction_outlines`. Once again, these views are where the object permission model is enforced, so always subclass rather than just replace them.

For the most part, these operate as generic views and all the same functionality applies.

Note: Basic templates for all of these views are provided, but it is expected that you will override them with your own as needed.

```
class fiction_outlines.views.SeriesListView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, django.views.generic.
           list.ListView
```

Generic view for viewing a list of series objects.

```
class fiction_outlines.views.SeriesDetailView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
           PermissionRequiredMixin, braces.views._queries.PrefetchRelatedMixin, django.
           views.generic.detail.DetailView
```

Generic view to see series details.

```
class fiction_outlines.views.SeriesUpdateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
           PermissionRequiredMixin, braces.views._queries.PrefetchRelatedMixin, django.
           views.generic.edit.UpdateView
```

Generic view for updating a series object.

```
class fiction_outlines.views.SeriesCreateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, django.views.generic.
           edit.CreateView
```

Generic view for creating series object.

```
class fiction_outlines.views.SeriesDeleteView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.DeleteView

    Generic view for deleting a series.

class fiction_outlines.views.CharacterListView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, django.views.generic.
    list.ListView

    Generic view for viewing character list.

class fiction_outlines.views.CharacterDetailView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.PrefetchRelatedMixin, django.
    views.generic.detail.DetailView

    Generic view for character details.

class fiction_outlines.views.CharacterUpdateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.PrefetchRelatedMixin, django.
    views.generic.edit.UpdateView

    Generic update view for character.

class fiction_outlines.views.CharacterCreateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, django.views.generic.
    edit.CreateView

    Generic view for creating a character.

class fiction_outlines.views.CharacterDeleteView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.PrefetchRelatedMixin, django.
    views.generic.edit.DeleteView

    Generic view for deleting a character.

class fiction_outlines.views.CharacterInstanceListView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.list.ListView

    Generic view for seeing a list of all character instances for a particular character.

class fiction_outlines.views.CharacterInstanceDetailView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.detail.DetailView

    Generic detail view for character instance.

class fiction_outlines.views.CharacterInstanceUpdateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, django.
    views.generic.edit.UpdateView

    Generic view for updating a character instance.

class fiction_outlines.views.CharacterInstanceCreateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.CreateView
```

Generic create view for a character instance.

```
class fiction_outlines.views.CharacterInstanceDeleteView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.edit.DeleteView
```

Generic view for deleting character instances.

```
class fiction_outlines.views.LocationListView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, django.views.generic.
    list.ListView
```

Generic view for locations.

```
class fiction_outlines.views.LocationDetailView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.PrefetchRelatedMixin, django.
    views.generic.detail.DetailView
```

Generic view for location details.

```
class fiction_outlines.views.LocationUpdateView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.UpdateView
```

Generic view for updating locations.

```
class fiction_outlines.views.LocationCreateView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, django.views.generic.
    edit.CreateView
```

Generic view for creating locations

```
class fiction_outlines.views.LocationDeleteView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.DeleteView
```

Generic view for deleting locations.

```
class fiction_outlines.views.LocationInstanceListView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.list.ListView
```

Generic view for looking at all location instances for a location.

```
class fiction_outlines.views.LocationInstanceDetailView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.detail.DetailView
```

Generic view for a location instance detail view.

```
class fiction_outlines.views.LocationInstanceUpdateView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, django.
    views.generic.edit.UpdateView
```

Generic view for updating a location instance. Not used since there are not details. But it's here if you want to subclass `LocationInstance` and customize it.

```
class fiction_outlines.views.LocationInstanceCreateView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.CreateView
```

Generic view for creating a location instance on a outline.

```
class fiction_outlines.views.LocationInstanceDeleteView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.edit.DeleteView
```

Generic delete view for Location Instance.

```
class fiction_outlines.views.OutlineListView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, braces.views._queries.
    SelectRelatedMixin, braces.views._queries.PrefetchRelatedMixin, django.
    views.generic.list.ListView
```

Generic view for Outline Outline list

```
class fiction_outlines.views.OutlineDetailView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.detail.DetailView
```

Generic view for Outline detail

```
class fiction_outlines.views.OutlineExport (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.detail.DetailView
```

Generic view to get an export of an outline record.

Takes a url kwarg of outline as the pk of the `fiction_outlines.models.Outline` The url kwarg of format determines the type returned. Current supported formats are opml, json, or md.

A view that can return a downloadable export of an outline with structure preserved. Formats supported: OPML, JSON, Markdown For fullest fidelity of data, JSON is the best choice. OPML and Markdown necessarily force the application to strip out quite a bit of nested data.

```
class fiction_outlines.views.OutlineUpdateView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.UpdateView
```

Generic update view for outline details.

```
class fiction_outlines.views.OutlineCreateView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, django.views.generic.
    edit.CreateView
```

Generic view for creating initial outline.

```
class fiction_outlines.views.OutlineDeleteView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.DeleteView
```

Generic delete view for an outline.

```
class fiction_outlines.views.ArcListView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.list.ListView
```


Generic list view for arcs in a outline

```
class fiction_outlines.views.ArcDetailView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.detail.DetailView
```

Generic view for arc details.

```
class fiction_outlines.views.ArcUpdateView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, django.
    views.generic.edit.UpdateView
```

Generic view for updating arc details

```
class fiction_outlines.views.ArcCreateView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.CreateView
```

Generic view for creating an arc.

```
class fiction_outlines.views.ArcDeleteView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.DeleteView
```

Generic view for deleting an arc

```
class fiction_outlines.views.ArcNodeDetailView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.detail.DetailView
```

View for looking at the details of an atomic node as opposed to the whole tree.

```
class fiction_outlines.views.ArcNodeUpdateView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.edit.UpdateView
```

View for editing details of an arc node (but not it's tree position).

```
class fiction_outlines.views.ArcNodeCreateView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.CreateView
```

Create view for an arc node. Assumes that the target position has already been passed to it via kwargs.

```
class fiction_outlines.views.ArcNodeDeleteView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.edit.DeleteView
```

View for deleting an arc node.

Incorporates logic to ensure that if the node represents the Hook or Resolution of the *Seven Point Story Structure*, it cannot be deleted.

```
class fiction_outlines.views.ArcNodeMoveView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.UpdateView
```

View for executing a move method on an arcnode.

```
class fiction_outlines.views.StoryNodeCreateView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.CreateView

    Creation view for a story node. Assumes the target and pos have been passed as kwargs.

class fiction_outlines.views.StoryNodeMoveView (**kwargs)
    Bases: fiction_outlines.views.StoryNodeUpdateView

    View for executing a move method on an arcnode.

class fiction_outlines.views.StoryNodeDetailView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.detail.DetailView

    View for looking at the details of an atomic story node as opposed to the whole tree.

class fiction_outlines.views.StoryNodeUpdateView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.edit.UpdateView

    View for doing basic updates to a story node, but not regarding its position in the tree.

    Will add additional form errors if it is attempted to edit the story_element_type in a manner which would
    break the structure of the outline.

class fiction_outlines.views.StoryNodeCreateView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.CreateView

    Creation view for a story node. Assumes the target and pos have been passed as kwargs.

class fiction_outlines.views.StoryNodeDeleteView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.edit.DeleteView

    Generic view for deleting a story node.
```

Model forms are provided for convenience and can be fed specific kwargs in order to ensure that users are not presented with choices that should not be permitted.

class `fiction_outlines.forms.CharacterInstanceForm(*args, **kwargs)`

Bases: `django.forms.models.ModelForm`

Form for creating character instances

Takes an additional kwarg of `character` which should represent a `Character` instance.

class `fiction_outlines.forms.LocationInstanceForm(*args, **kwargs)`

Bases: `django.forms.models.ModelForm`

Form for creating location instances.

Takes an additional kwarg of `location` which should represent an instance of `Location`.

class `fiction_outlines.forms.CharacterForm(*args, **kwargs)`

Bases: `django.forms.models.ModelForm`

Form for `Character` model.

Takes an additional kwarg of `user` which should represent an instance of `AUTH_USER_MODEL`.

class `fiction_outlines.forms.LocationForm(*args, **kwargs)`

Bases: `django.forms.models.ModelForm`

Form class for `Locations`

Takes an additional kwarg of `user` which should represent an instance of `AUTH_USER_MODEL`.

class `fiction_outlines.forms.OutlineForm(*args, **kwargs)`

Bases: `django.forms.models.ModelForm`

Form class for `Outline` model

Takes an additional kwarg of `user` which should represent an instance of `AUTH_USER_MODEL`.

```
class fiction_outlines.forms.OutlineMoveNodeForm(data=None, files=None,
auto_id='id_%s', prefix=None,
initial=None, error_class=<class
'django.forms.utils.ErrorList'>,
label_suffix=':',
empty_permitted=False, in-
stance=None, **kwargs)
```

Bases: `treebeard.forms.MoveNodeForm`

Subclass of base `treebeard` move node form allowing us to restrict target node options to within a single tree.

It is recommended that you do not subclass or directly call this form, but instead use `treebeard.forms.movenodeform_factory`.

Example:

```
from treebeard.forms import movenodeform_factory
from fiction_outlines import forms
from fiction_outlines.views import ArcNodeMoveView

class SomeFormOrModelView(ArcNodeMoveView):

    model = ArcElementNode # As an example
    form_class = movenodeform_factory(ArcElementNode, form=forms.
    OutlineMoveNodeForm, ...)
    ...
```

10.1 fiction_outlines package

10.1.1 Submodules

10.1.2 fiction_outlines.admin module

10.1.3 fiction_outlines.apps module

```
class fiction_outlines.apps.FictionOutlinesConfig(app_name, app_module)
    Bases: django.apps.config.AppConfig
    name = 'fiction_outlines'
    ready()
        Override this method in subclasses to run code when Django starts.
    verbose_name = 'Fiction Outlines'
```

10.1.4 fiction_outlines.models module

```
class fiction_outlines.models.Arc(*args, **kwargs)
    Bases: fiction_outlines.models.TimeStampedModel
    A MACE arc for a outline.
```

Parameters

- **created**(*AutoCreatedField*) – Created
- **modified**(*AutoLastModifiedField*) – Modified
- **id**(*UUIDField*) – Id
- **mace_type**(*CharField*) – The MACE type of the Arc.

- **outline_id** (*ForeignKey*) – Arc belongs to this outline.
- **name** (*CharField*) – Name of this Arc (makes it easier for you to keep track of it.)

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

arc_root_node

Returns the root node from this object's `ArcElementNode` tree.

arcelementnode_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

current_errors

Returns list of errors from `arc_validation`.

fetch_arc_errors()

Evaluates the current tree of the arc and provides a list of errors that the user should correct.

generate_template_arc_tree()

Generate a seven point template in this arc. Arc must be empty.

get_absolute_url()**get_mace_type_display(*, field=<django.db.models.fields.CharField: mace_type>)****get_next_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)****get_next_by_modified(*, field=<fiction_outlines.models.AutoLastModifiedField: modified>, is_next=True, **kwargs)****get_previous_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs)****get_previous_by_modified(*, field=<fiction_outlines.models.AutoLastModifiedField: modified>, is_next=False, **kwargs)****id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

mace_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

outline

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

outline_id**refresh_from_db** (*args, **kwargs)

Reload field values from the database.

By default, the reloading happens from the database this instance was loaded from, or by the read router if this instance wasn't loaded from any database. The using parameter will override the default.

Fields can be used to specify which fields to reload. The fields should be an iterable of field attnames. If fields is None, then all non-deferred fields are reloaded.

When accessing deferred fields of an instance, the deferred loading of the field will call this method.

validate_first_element ()

Ensures that the first node for the direct decendents of root is the hook.

validate_generations ()

Make sure that the descendent depth is valid.

validate_last_element ()

Ensures that the last element of the arc is the resolution.

validate_milestones ()

Reviews the arc element tree to ensure that milestones appear in the right order.

class fiction_outlines.models.ArcElementNode (*args, **kwargs)

Bases: *fiction_outlines.models.TimeStampedModel*, *treebeard.mp_tree.MP_Node*

Tree nodes for the arc elements.

Parameters

- **created** (*AutoCreatedField*) – Created
- **path** (*CharField*) – Path
- **depth** (*PositiveIntegerField*) – Depth
- **numchild** (*PositiveIntegerField*) – Numchild
- **modified** (*AutoLastModifiedField*) – Modified
- **id** (*UUIDField*) – Id
- **arc_element_type** (*CharField*) – What part of the arc does this represent?
- **arc_id** (*ForeignKey*) – Parent arc.
- **headline** (*CharField*) – Autogenerated from description
- **description** (*TextField*) – Describe what happens at this moment in the story...
- **story_element_node_id** (*ForeignKey*) – Which story node is this element associated with?

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

add_child(*arc_element_type*, *description=None*, *story_element_node=None*, ***kwargs*)

Overrides the default *treebeard* function, adding additional integrity checks.

add_sibling(*pos=None*, *arc_element_type=None*, *description=None*, *story_element_node=None*, ***kwargs*)

Overrides the default *treebeard* function, adding additional integrity checks.

arc

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

arc_element_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

arc_id**assoc_characters**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

assoc_locations

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

description

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_absolute_url()

get_arc_element_type_display(***, *field=<django.db.models.fields.CharField: arc_element_type>*)


```
get_next_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=True,
                    **kwargs)
```

```
get_next_by_modified(*, field=<fiction_outlines.models.AutoLastModifiedField: modified>,
                     is_next=True, **kwargs)
```

```
get_previous_by_created(*, field=<model_utils.fields.AutoCreatedField: created>,
                        is_next=False, **kwargs)
```

```
get_previous_by_modified(*, field=<fiction_outlines.models.AutoLastModifiedField: modified>,
                          is_next=False, **kwargs)
```

headline

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_milestone

Does this node represent an arc milestone?

milestone_seq

Returns the milestone sequence based off of the arc element definitions.

parent_outline

Private method to fetch parent outline.

steplen = 5

story_element_node

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

story_element_node_id

exception fiction_outlines.models.ArcGenerationError

Bases: *fiction_outlines.models.ArcIntegrityError*

Exception for when a non-milestone node is placed at an invalid level of descendency.

exception fiction_outlines.models.ArcIntegrityError

Bases: django.db.utils.IntegrityError

Generic exception for Arc structural warnings.

class fiction_outlines.models.AutoLastModifiedField(*args, **kwargs)

Bases: model_utils.fields.AutoLastModifiedField

Override of the default model_utils behavior to ensure that when an instance is created that the modified and created will be the same.

pre_save(model_instance, add)

Return field's value just before saving.

class fiction_outlines.models.Character(*args, **kwargs)

Bases: *fiction_outlines.models.TimeStampedModel*

Reusable character definition model.

Parameters

- **created** (*AutoCreatedField*) – Created
- **modified** (*AutoLastModifiedField*) – Modified
- **id** (*UUIDField*) – Id
- **name** (*CharField*) – Name of the character.
- **description** (*TextField*) – Notes about the character to help you remember.
- **user_id** (*ForeignKey*) – The user that created this character.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

characterinstance_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

description

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_absolute_url()

get_next_by_created (*, *field*=<*model_utils.fields.AutoCreatedField*: *created*>, *is_next*=True, ***kwargs*)

get_next_by_modified (*, *field*=<*fiction_outlines.models.AutoLastModifiedField*: *modified*>, *is_next*=True, ***kwargs*)

get_previous_by_created (*, *field*=<*model_utils.fields.AutoCreatedField*: *created*>, *is_next*=False, ***kwargs*)

get_previous_by_modified (*, *field*=<*fiction_outlines.models.AutoLastModifiedField*: *modified*>, *is_next*=False, ***kwargs*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <*django.db.models.manager.Manager* object>

series

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

tagged_items

Accessor to the related objects manager on the one-to-many relation created by `GenericRelation`.

In the example:

```
class Post(Model):
    comments = GenericRelation(Comment)
```

`post.comments` is a `ReverseGenericManyToOneDescriptor` instance.

tags = <taggit.managers._TaggableManager object>

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

user_id

class `fiction_outlines.models.CharacterInstance` (*args, **kwargs)

Bases: `fiction_outlines.models.TimeStampedModel`

An instance of the character object that can be associated with outlines.

Parameters

- **created** (`AutoCreatedField`) – Created
- **modified** (`AutoLastModifiedField`) – Modified
- **id** (`UUIDField`) – Id
- **main_character** (`BooleanField`) – Is this character the main character for the outline?
- **pov_character** (`BooleanField`) – Is this character a POV character?
- **protagonist** (`BooleanField`) – Does this character serve as the protagonist for this outline?
- **antagonist** (`BooleanField`) – Does this character serve as an antagonist for this outline?
- **obstacle** (`BooleanField`) – Is this character an obstacle in the outline? (not antagonist)
- **villain** (`BooleanField`) – Is the character a straight-out villain?
- **character_id** (`ForeignKey`) – Reference to originating character object.
- **outline_id** (`ForeignKey`) – Outline this instance is associated with.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

antagonist

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

arcelementnode_set

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

character

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

character_id**get_absolute_url()**

get_next_by_created(* , field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)

get_next_by_modified(* , field=<fiction_outlines.models.AutoLastModifiedField: modified>, is_next=True, **kwargs)

get_previous_by_created(* , field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs)

get_previous_by_modified(* , field=<fiction_outlines.models.AutoLastModifiedField: modified>, is_next=False, **kwargs)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

main_character

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

obstacle

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

outline

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

outline_id**pov_character**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

protagonist

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

storyelementnode_set

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

villain

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

exception `fiction_outlines.models.GenericArcSequenceError`

Bases: `fiction_outlines.models.ArcIntegrityError`

Exception for when a non-milestone element is placed in an invalid sequence.

class `fiction_outlines.models.Location(*args, **kwargs)`

Bases: `fiction_outlines.models.TimeStampedModel`

Reusable location definition model

Parameters

- **created** (`AutoCreatedField`) – Created
- **modified** (`AutoLastModifiedField`) – Modified
- **id** (`UUIDField`) – Id
- **name** (`CharField`) – Name of the location.
- **description** (`TextField`) – Notes about the location to help you remember.
- **user_id** (`ForeignKey`) – The user that created this location.

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

description

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_absolute_url()

get_next_by_created(* , field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)

get_next_by_modified(* , field=<fiction_outlines.models.AutoLastModifiedField: modified>, is_next=True, **kwargs)

get_previous_by_created(* , field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs)

get_previous_by_modified(* , field=<fiction_outlines.models.AutoLastModifiedField: modified>, is_next=False, **kwargs)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

locationinstance_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

series

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

tagged_items

Accessor to the related objects manager on the one-to-many relation created by `GenericRelation`.

In the example:

```
class Post(Model):
    comments = GenericRelation(Comment)
```

`post.comments` is a `ReverseGenericManyToOneDescriptor` instance.

tags = <taggit.managers._TaggableManager object>

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

user_id

class `fiction_outlines.models.LocationInstance(*args, **kwargs)`

Bases: `fiction_outlines.models.TimeStampedModel`

An instance of the given location that can be associated with a given outline.

Parameters

- **created** (`AutoCreatedField`) – Created
- **modified** (`AutoLastModifiedField`) – Modified
- **id** (`UUIDField`) – Id
- **location_id** (`ForeignKey`) – Originating location object.
- **outline_id** (`ForeignKey`) – Outline this object is associated with.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

arcelementnode_set

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

get_absolute_url()

get_next_by_created(*, `field=<model_utils.fields.AutoCreatedField: created>`, `is_next=True`, **kwargs)

get_next_by_modified(*, `field=<fiction_outlines.models.AutoLastModifiedField: modified>`, `is_next=True`, **kwargs)

get_previous_by_created(*, `field=<model_utils.fields.AutoCreatedField: created>`, `is_next=False`, **kwargs)

get_previous_by_modified(*, `field=<fiction_outlines.models.AutoLastModifiedField: modified>`, `is_next=False`, **kwargs)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

location

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

location_id

objects = <django.db.models.manager.Manager object>

outline

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

outline_id**storyelementnode_set**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

Pizza.toppings and Topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

exception fiction_outlines.models.MilestoneDepthError

Bases: *fiction_outlines.models.ArcIntegrityError*

Exception for when an attempt to make a milestone a descendent of any node besides the root node for the tree.

exception fiction_outlines.models.MilestoneSequenceError

Bases: *fiction_outlines.models.ArcIntegrityError*

Exception for when milestone arc elements violate their defined sequence rules.

class fiction_outlines.models.Outline(*args, **kwargs)

Bases: *fiction_outlines.models.TimeStampedModel*

The typical top of the hierarchy when not enclosed in a series.

Parameters

- **created** (AutoCreatedField) – Created
- **modified** (AutoLastModifiedField) – Modified

- **id** (*UUIDField*) – Id
- **title** (*CharField*) – Outline title. You can always change this later.
- **description** (*TextField*) – Optionally, describe the story. Or use for notes.
- **series_id** (*ForeignKey*) – Belongs to series.
- **user_id** (*ForeignKey*) – The user that created this outline.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

arc_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

characterinstance_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

create_arc (*mace_type*, *name*)

Creates the story arc and initial tree for that arc for the current outline. Returns the resulting Arc instance.

description

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_absolute_url ()

```
get_next_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=True,
                    **kwargs)
```

```
get_next_by_modified(*, field=<fiction_outlines.models.AutoLastModifiedField: modified>,
                     is_next=True, **kwargs)
```

```
get_previous_by_created(*, field=<model_utils.fields.AutoCreatedField: created>,
                        is_next=False, **kwargs)
```

```
get_previous_by_modified(*, field=<fiction_outlines.models.AutoLastModifiedField: modified>,
                          is_next=False, **kwargs)
```

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

length_estimate

Calculates and estimated word count based on number of characters, locations, and arcs. For reference see: <http://www.writingexcuses.com/2017/07/02/12-27-choosing-a-length/>

locationinstance_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

objects = <django.db.models.manager.Manager object>

refresh_from_db (*args, **kwargs)

Reload field values from the database.

By default, the reloading happens from the database this instance was loaded from, or by the read router if this instance wasn't loaded from any database. The `using` parameter will override the default.

Fields can be used to specify which fields to reload. The fields should be an iterable of field attnames. If fields is None, then all non-deferred fields are reloaded.

When accessing deferred fields of an instance, the deferred loading of the field will call this method.

series

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

series_id**story_tree_root**

Fetches the root node for the outline's `StoryElementNode` tree.

storyelementnode_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

tagged_items

Accessor to the related objects manager on the one-to-many relation created by GenericRelation.

In the example:

```
class Post(Model):
    comments = GenericRelation(Comment)
```

post.comments is a ReverseGenericManyToOneDescriptor instance.

tags = <taggit.managers._TaggableManager object>

title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

user_id**validate_nesting()**

Reviews the story tree and validates associated arc elements are nested appropriately. Returns a dict of errors.

class fiction_outlines.models.**Series** (*args, **kwargs)

Bases: *fiction_outlines.models.TimeStampedModel*

Container object to hold multiple outline objects if necessary.

Parameters

- **created** (*AutoCreatedField*) – Created
- **modified** (*AutoLastModifiedField*) – Modified
- **id** (*UUIDField*) – Id
- **title** (*CharField*) – Name of the series. You can always change this later.
- **description** (*TextField*) – Jot down a description about your series.
- **user_id** (*ForeignKey*) – The user that created this Series.

exception DoesNotExist

Bases: *django.core.exceptions.ObjectDoesNotExist*

exception MultipleObjectsReturned

Bases: *django.core.exceptions.MultipleObjectsReturned*

character_set

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

description

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_absolute_url()

get_next_by_created(* , field=<model_utils.fields.AutoCreatedField: created>, is_next=True, **kwargs)

get_next_by_modified(* , field=<fiction_outlines.models.AutoLastModifiedField: modified>, is_next=True, **kwargs)

get_previous_by_created(* , field=<model_utils.fields.AutoCreatedField: created>, is_next=False, **kwargs)

get_previous_by_modified(* , field=<fiction_outlines.models.AutoLastModifiedField: modified>, is_next=False, **kwargs)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

location_set

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

objects = <django.db.models.manager.Manager object>

outline_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

tagged_items

Accessor to the related objects manager on the one-to-many relation created by `GenericRelation`.

In the example:

```
class Post(Model):
    comments = GenericRelation(Comment)
```

`post.comments` is a `ReverseGenericManyToOneDescriptor` instance.

tags = <taggit.managers._TaggableManager object>

title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

user_id

class fiction_outlines.models.StoryElementNode(*args, **kwargs)

Bases: *fiction_outlines.models.TimeStampedModel*, *treebeard.mp_tree.MP_Node*

Tree nodes for the overall outline of the story.

Parameters

- **created** (*AutoCreatedField*) – Created
- **path** (*CharField*) – Path
- **depth** (*PositiveIntegerField*) – Depth
- **numchild** (*PositiveIntegerField*) – Numchild
- **modified** (*AutoLastModifiedField*) – Modified
- **id** (*UUIDField*) – Id
- **name** (*CharField*) – Optional name/title for this element of the story.
- **description** (*TextField*) – Optional description for this element of the story.
- **outline_id** (*ForeignKey*) – Parent outline.
- **story_element_type** (*CharField*) – What part of the story does this represent? A scene? A chapter?

exception DoesNotExist

Bases: *django.core.exceptions.ObjectDoesNotExist*

exception MultipleObjectsReturned

Bases: *django.core.exceptions.MultipleObjectsReturned*

add_child (*story_element_type=None, outline=None, name=None, description=None, **kwargs*)

An override of the treebeard add_child() method so we can send a signal.

add_sibling (*story_element_type=None, outline=None, name=None, description=None, pos=None, **kwargs*)

Override of treebeard api to allow us to send a signal.

all_characters

Returns a queryset of all characters associated with this node and its descendants, excluding any duplicates.

all_locations

Returns a queryset of all locations associated with this node and its descendants, excluding any duplicates.

arcelementnode_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

assoc_characters

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

assoc_locations

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

description

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_absolute_url()

```
get_next_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=True,
                    **kwargs)
```

```
get_next_by_modified(*, field=<fiction_outlines.models.AutoLastModifiedField: modified>,
                     is_next=True, **kwargs)
```

```
get_previous_by_created(*, field=<model_utils.fields.AutoCreatedField: created>,
                        is_next=False, **kwargs)
```

```
get_previous_by_modified(*, field=<fiction_outlines.models.AutoLastModifiedField: modified>,
                          is_next=False, **kwargs)
```

```
get_story_element_type_display(*, field=<django.db.models.fields.CharField:
                                     story_element_type>)
```

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

impact_rating

Returns the impact rating for this node. Impact rating is a measure of how powerful this moment in the story is by evaluating how many simultaneous arc elements are associated with it. There is also a generational bleed element, where the impact score creates shockwaves throughout their direct ancestor and descendant nodes. This echo fades fast, but the bigger the impact, the farther it goes.

Currently, the impact bleed does not extend to sibling nodes.

WARNING: Here be dragons.

move (*target*, *pos=None*)

An override of the treebeard api in order to send a signal in advance.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

outline

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

outline_id**steplen = 5****story_element_type**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class fiction_outlines.models.TimeStampedModel(*args, **kwargs)
```

Bases: model_utils.models.TimeStampedModel

Override the model_utils behavior to use our new field.

Parameters

- **created** (*AutoCreatedField*) – Created
- **modified** (*AutoLastModifiedField*) – Modified

class Meta

Bases: object

abstract = False

```
get_next_by_created(*, field=<model_utils.fields.AutoCreatedField: created>, is_next=True,
                    **kwargs)
```

```
get_next_by_modified(*, field=<fiction_outlines.models.AutoLastModifiedField: modified>,
                     is_next=True, **kwargs)
```

```
get_previous_by_created(*, field=<model_utils.fields.AutoCreatedField: created>,
                       is_next=False, **kwargs)
```

```
get_previous_by_modified(*, field=<fiction_outlines.models.AutoLastModifiedField: modified>,
                        is_next=False, **kwargs)
```

```
class fiction_outlines.models.UUIDCharacterTag(*args, **kwargs)
    Bases: taggit.models.GenericUUIDTaggedItemBase, taggit.models.TaggedItemBase
```

Character tags with UUID primary keys

Parameters

- **id** (*AutoField*) – Id
- **tag_id** (*ForeignKey*) – Tag
- **content_type_id** (*ForeignKey*) – Content type
- **object_id** (*UUIDField*) – Object id

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

content_object

Provide a generic many-to-one relation through the `content_type` and `object_id` fields.

This class also doubles as an accessor to the related object (similar to `ForwardManyToOneDescriptor`) by adding itself as a model attribute.

content_type

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = `<django.db.models.manager.Manager object>`

tag

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

```
class fiction_outlines.models.UUIDLocationTag(*args, **kwargs)
```

Bases: `taggit.models.GenericUUIDTaggedItemBase`, `taggit.models.TaggedItemBase`

Location tags with UUID primary keys

Parameters

- **id** (*AutoField*) – Id
- **tag_id** (*ForeignKey*) – Tag

- **content_type_id** (*ForeignKey*) – Content type
- **object_id** (*UUIDField*) – Object id

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

content_object

Provide a generic many-to-one relation through the `content_type` and `object_id` fields.

This class also doubles as an accessor to the related object (similar to `ForwardManyToOneDescriptor`) by adding itself as a model attribute.

content_type

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = `<django.db.models.manager.Manager object>`

tag

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

class `fiction_outlines.models.UUIDOutlineTag(*args, **kwargs)`

Bases: `taggit.models.GenericUUIDTaggedItemBase`, `taggit.models.TaggedItemBase`

Outline tags with UUID primary keys

Parameters

- **id** (*AutoField*) – Id
- **tag_id** (*ForeignKey*) – Tag
- **content_type_id** (*ForeignKey*) – Content type
- **object_id** (*UUIDField*) – Object id

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

content_object

Provide a generic many-to-one relation through the `content_type` and `object_id` fields.

This class also doubles as an accessor to the related object (similar to `ForwardManyToOneDescriptor`) by adding itself as a model attribute.

content_type

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = `<django.db.models.manager.Manager object>`

tag

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

10.1.5 fiction_outlines.receivers module

Receiving methods for signals go here.

`fiction_outlines.receivers.arc_node_edit_add_missing_characters_and_locations_to_related_s`

If an `arc_element` is modified and its characters/locations are not already in the story node, add them. We don't assume that removing the arc element would change the characters or locations as of yet. This takes up a little more space in the database, but the additional flexibility for users is worth it.

`fiction_outlines.receivers.generate_headline_from_description`(*sender, instance, *args, **kwargs*)

Auto generate the headline of the node from the first lines of the description.

`fiction_outlines.receivers.story_node_add_arc_element_update_characters_locations` (*sender*, *instance*, *created*, *args*, *kwargs*)

If an arc element is added to a story element node, add any missing elements or locations.

`fiction_outlines.receivers.story_root_for_new_outline` (*sender*, *instance*, *created*, *args*, *kwargs*)

If a new instance of a Outline is created, also create the root node of the story tree.

`fiction_outlines.receivers.validate_against_prohibited_actions` (*sender*, *instance*, *action*, *target_node_type=None*, *target_node=None*, *pos=None*, *args*, *kwargs*)

`fiction_outlines.receivers.validate_arc_links_same_outline` (*sender*, *instance*, *args*, *kwargs*)

Evaluates attempts to link an arc to a story node from another outline.

`fiction_outlines.receivers.validate_character_for_story_element` (*sender*, *instance*, *action*, *reverse*, *pk_set*, *args*, *kwargs*)

Validates that character is from the same outline as the story node.

`fiction_outlines.receivers.validate_character_instance_valid_for_arc` (*sender*, *instance*, *action*, *reverse*, *pk_set*, *args*, *kwargs*)

Evaluate attempts to assign a character instance to ensure it is from same outline.

`fiction_outlines.receivers.validate_generations_for_story_elements` (*sender*, *instance*, *action*, *target_node_type=None*, *target_node=None*, *pos=None*, *args*, *kwargs*)

Unlike arc nodes, for which we just warn about structure, the story tree allowed parent/child rules must be strictly enforced.

```
fiction_outlines.receivers.validate_location_for_story_element(sender, instance, action, reverse, pk_set, *args, **kwargs)
```

Validates that location is from same outline as story node.

```
fiction_outlines.receivers.validate_location_instance_valid_for_arc(sender, instance, action, reverse, pk_set, *args, **kwargs)
```

Evaluates attempts to add location instances to arc, ensuring they are from same outline.

10.1.6 fiction_outlines.signals module

Custom signals sent by this app.

Current list:

tree_manipulation: Sent when either the ArcElementNode or StoryElementNode trees have their structure manipulated.

10.1.7 fiction_outlines.urls module

URLs for fiction_outlines.

```
fiction_outlines.urls.path(route, view, kwargs=None, name=None, *, Pattern=<class 'django.urls.resolvers.RoutePattern'>)
```

10.1.8 fiction_outlines.views module

Views for fiction_outlines.

```
class fiction_outlines.views.ArcCreateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.PermissionRequiredMixin, django.views.generic.edit.CreateView

    Generic view for creating an arc.

    dispatch(request, *args, **kwargs)

    fields = ['name', 'mace_type']

    form_valid(form)
        If the form is valid, save the associated model.

    get_context_data(**kwargs)
        Insert the form into the context dict.

    get_permission_object()
        Override this method to provide the object to check for permission against. By default uses self.get_object() as provided by SingleObjectMixin. Returns None if there's no get_object method.
```

```
get_success_url()
    Return the URL to redirect to after processing a valid form.

model
    alias of fiction_outlines.models.Arc

permission_required = 'fiction_outlines.edit_outline'

success_url = None

template_name = 'fiction_outlines/arc_create.html'

class fiction_outlines.views.ArcDeleteView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.DeleteView

    Generic view for deleting an arc

    context_object_name = 'arc'

    dispatch(request, *args, **kwargs)

    get_success_url()

    model
        alias of fiction_outlines.models.Arc

    permission_required = 'fiction_outlines.delete_arc'

    pk_url_kwarg = 'arc'

    success_url = None

    template_name = 'fiction_outlines/arc_delete.html'

class fiction_outlines.views.ArcDetailView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.detail.DetailView

    Generic view for arc details.

    context_object_name = 'arc'

    dispatch(request, *args, **kwargs)

    get_context_data(**kwargs)
        Insert the single object into the context dict.

    model
        alias of fiction_outlines.models.Arc

    permission_required = 'fiction_outlines.view_arc'

    pk_url_kwarg = 'arc'

    prefetch_related = ['arcelementnode_set']

    select_related = ['outline']

    template_name = 'fiction_outlines/arc_detail.html'

class fiction_outlines.views.ArcListView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.list.ListView

    Generic list view for arcs in a outline
```

```
context_object_name = 'arc_list'

dispatch(request, *args, **kwargs)

get_context_data(**kwargs)
    Get the context for this view.

get_permission_object()
    Override this method to provide the object to check for permission against. By default uses self.
    get_object() as provided by SingleObjectMixin. Returns None if there's no get_object
    method.

get_queryset()
    Return the list of items for this view.

    The return value must be an iterable and may be an instance of QuerySet in which case QuerySet specific
    behavior will be enabled.

model
    alias of fiction_outlines.models.Arc

permission_required = 'fiction_outlines.view_outline'

template_name = 'fiction_outlines/arc_list.html'

class fiction_outlines.views.ArcNodeCreateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.CreateView

    Create view for an arc node. Assumes that the target position has already been passed to it via kwargs.

    dispatch(request, *args, **kwargs)

    form_class
        alias of fiction_outlines.forms.ArcNodeForm

    form_valid(form)
        If the form is valid, save the associated model.

    get_form_kwargs()
        Return the keyword arguments for instantiating the form.

    get_permission_object()
        Override this method to provide the object to check for permission against. By default uses self.
        get_object() as provided by SingleObjectMixin. Returns None if there's no get_object
        method.

    get_success_url()
        Return the URL to redirect to after processing a valid form.

    model
        alias of fiction_outlines.models.ArcElementNode

    permission_required = 'fiction_outlines.edit_arc'

    success_url = None

    template_name = 'fiction_outlines/arcnode_create.html'

class fiction_outlines.views.ArcNodeDeleteView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.edit.DeleteView

    View for deleting an arc node.
```

```
context_object_name = 'arcnode'

delete(request, *args, **kwargs)
    Call the delete() method on the fetched object and then redirect to the success URL.

dispatch(request, *args, **kwargs)

form_valid(form)

get_success_url()

model
    alias of fiction_outlines.models.ArcElementNode

node_deletion_safe()

permission_required = 'fiction_outlines.delete_arc_node'

pk_url_kwarg = 'arcnode'

prefetch_related = ['assoc_characters', 'assoc_locations']

select_related = ['arc', 'arc__outline', 'story_element_node']

success_url = None

template_name = 'fiction_outlines/arcnode_delete.html'

class fiction_outlines.views.ArcNodeDetailView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.detail.DetailView
    View for looking at the details of an atomic node as opposed to the whole tree.

    context_object_name = 'arcnode'

    model
        alias of fiction_outlines.models.ArcElementNode

    permission_required = 'fiction_outlines.view_arc_node'

    pk_url_kwarg = 'arcnode'

    prefetch_related = ['assoc_characters', 'assoc_locations']

    select_related = ['arc', 'arc__outline', 'story_element_node']

    template_name = 'fiction_outlines/arcnode_detail.html'

class fiction_outlines.views.ArcNodeMoveView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.UpdateView
    View for executing a move method on an arcnode.

    context_object_name = 'arcnode'

    form_class
        alias of django.forms.widgets.ArcElementNodeForm

    form_valid(form)
        If the form is valid, save the associated model.

    get_form_kwargs()
        Return the keyword arguments for instantiating the form.
```

```
get_success_url()
    Return the URL to redirect to after processing a valid form.

model
    alias of fiction_outlines.models.ArcElementNode

permission_required = 'fiction_outlines.edit_arc_node'

pk_url_kwarg = 'arcnode'

success_url = None

template_name = 'fiction_outlines/arcnode_move.html'

class fiction_outlines.views.ArcNodeUpdateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.edit.UpdateView

    View for editing details of an arc node (but not it's tree position).

    context_object_name = 'arcnode'

    form_class
        alias of fiction_outlines.forms.ArcNodeForm

    form_valid(form)
        If the form is valid, save the associated model.

    get_form_kwargs()
        Return the keyword arguments for instantiating the form.

    get_success_url()
        Return the URL to redirect to after processing a valid form.

    model
        alias of fiction_outlines.models.ArcElementNode

    permission_required = 'fiction_outlines.edit_arc_node'

    pk_url_kwarg = 'arcnode'

    prefetch_related = ['assoc_characters', 'assoc_locations']

    select_related = ['arc', 'arc__outline', 'story_element_node']

    success_url = None

    template_name = 'fiction_outlines/arcnode_update.html'

class fiction_outlines.views.ArcUpdateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, django.
    views.generic.edit.UpdateView

    Generic view for updating arc details

    context_object_name = 'arc'

    fields = ['name', 'mace_type']

    get_success_url()
        Return the URL to redirect to after processing a valid form.

    model
        alias of fiction_outlines.models.Arc
```



```
permission_required = 'fiction_outlines.edit_arc'
pk_url_kwarg = 'arc'
select_related = ['outline']
success_url = None
template_name = 'fiction_outlines/arc_update.html'

class fiction_outlines.views.CharacterCreateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, django.views.generic.
           edit.CreateView

    Generic view for creating a character.

    form_class
        alias of fiction_outlines.forms.CharacterForm

    form_valid(form)
        If the form is valid, save the associated model.

    get_form_kwargs()
        Return the keyword arguments for instantiating the form.

    get_success_url()
        Return the URL to redirect to after processing a valid form.

    model
        alias of fiction_outlines.models.Character

    success_url = None
    template_name = 'fiction_outlines/character_create.html'

class fiction_outlines.views.CharacterDeleteView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
           PermissionRequiredMixin, braces.views._queries.PrefetchRelatedMixin, django.
           views.generic.edit.DeleteView

    Generic view for deleting a character.

    context_object_name = 'character'

    model
        alias of fiction_outlines.models.Character

    permission_required = 'fiction_outlines.delete_character'
    pk_url_kwarg = 'character'
    prefetch_related = ['series', 'characterinstance_set']
    success_url = '/fiction-outlines/characters/'
    template_name = 'fiction_outlines/character_delete.html'

class fiction_outlines.views.CharacterDetailView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
           PermissionRequiredMixin, braces.views._queries.PrefetchRelatedMixin, django.
           views.generic.detail.DetailView

    Generic view for character details.

    context_object_name = 'character'
```

```
model
    alias of fiction_outlines.models.Character

permission_required = 'fiction_outlines.view_character'

pk_url_kwarg = 'character'

prefetch_related = ['series', 'characterinstance_set', 'characterinstance_set__outline']

template_name = 'fiction_outlines/character_detail.html'

class fiction_outlines.views.CharacterInstanceCreateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.CreateView

    Generic create view for a character instance.

    dispatch(request, *args, **kwargs)

    form_class
        alias of fiction_outlines.forms.CharacterInstanceForm

    form_valid(form)
        If the form is valid, save the associated model.

    get_context_data(**kwargs)
        Insert the form into the context dict.

    get_form_kwargs()
        Return the keyword arguments for instantiating the form.

    get_success_url()
        Return the URL to redirect to after processing a valid form.

    has_permission()
        Override this method to customize the way permissions are checked.

    model
        alias of fiction_outlines.models.CharacterInstance

    outline = None

    permission_required = None

    success_url = None

    template_name = 'fiction_outlines/character_instance_create.html'

class fiction_outlines.views.CharacterInstanceDeleteView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.edit.DeleteView

    Generic view for deleting character instances.

    context_object_name = 'character_instance'

    dispatch(request, *args, **kwargs)

    get_success_url()

    model
        alias of fiction_outlines.models.CharacterInstance

    permission_required = 'fiction_outlines.delete_character_instance'

    pk_url_kwarg = 'instance'
```

```
prefetch_related = ['arcelementnode_set', 'storyelementnode_set']
select_related = ['character', 'outline']
success_url = None
template_name = 'fiction_outlines/character_instance_delete.html'
```

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

Bases: `django.contrib.auth.mixins.LoginRequiredMixin`, `rules.contrib.views.PermissionRequiredMixin`, `braces.views._queries.SelectRelatedMixin`, `braces.views._queries.PrefetchRelatedMixin`, `django.views.generic.detail.DetailView`

Generic detail view for character instance.

context_object_name = 'character_instance'

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

get_permission_object ()

Override this method to provide the object to check for permission against. By default uses `self.get_object()` as provided by `SingleObjectMixin`. Returns `None` if there's no `get_object` method.

model

alias of `fiction_outlines.models.CharacterInstance`

permission_required = 'fiction_outlines.view_character'

pk_url_kwarg = 'instance'

prefetch_related = ['arcelementnode_set', 'storyelementnode_set']

select_related = ['character', 'outline']

template_name = 'fiction_outlines/character_instance_detail.html'

class `fiction_outlines.views.CharacterInstanceListView` (***kwargs*)

Bases: `django.contrib.auth.mixins.LoginRequiredMixin`, `rules.contrib.views.PermissionRequiredMixin`, `django.views.generic.list.ListView`

Generic view for seeing a list of all character instances for a particular character.

context_object_name = 'character_instance_list'

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

get_context_data (***kwargs*)

Get the context for this view.

get_permission_object ()

Override this method to provide the object to check for permission against. By default uses `self.get_object()` as provided by `SingleObjectMixin`. Returns `None` if there's no `get_object` method.

get_queryset ()

Return the list of items for this view.

The return value must be an iterable and may be an instance of *QuerySet* in which case *QuerySet* specific behavior will be enabled.

model

alias of `fiction_outlines.models.CharacterInstance`

permission_required = 'fiction_outlines.view_character'

```
template_name = 'fiction_outlines/character_instance_list.html'
```

class `fiction_outlines.views.CharacterInstanceUpdateView` (***kwargs*)

Bases: `django.contrib.auth.mixins.LoginRequiredMixin`, `rules.contrib.views.PermissionRequiredMixin`, `braces.views._queries.SelectRelatedMixin`, `django.views.generic.edit.UpdateView`

Generic view for updating a character instance.

context_object_name = `'character_instance'`

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

form_class

alias of `fiction_outlines.forms.CharacterInstanceForm`

form_valid (*form*)

If the form is valid, save the associated model.

get_form_kwargs ()

Return the keyword arguments for instantiating the form.

get_permission_object ()

Override this method to provide the object to check for permission against. By default uses `self.get_object()` as provided by `SingleObjectMixin`. Returns `None` if there's no `get_object` method.

get_success_url ()

Return the URL to redirect to after processing a valid form.

model

alias of `fiction_outlines.models.CharacterInstance`

permission_required = `'fiction_outlines.edit_character'`

pk_url_kwarg = `'instance'`

select_related = `['character', 'outline']`

success_url = `None`

template_name = `'fiction_outlines/character_instance_update.html'`

class `fiction_outlines.views.CharacterListView` (***kwargs*)

Bases: `django.contrib.auth.mixins.LoginRequiredMixin`, `django.views.generic.list.ListView`

Generic view for viewing character list.

context_object_name = `'character_list'`

get_queryset ()

Return the list of items for this view.

The return value must be an iterable and may be an instance of *QuerySet* in which case *QuerySet* specific behavior will be enabled.

model

alias of `fiction_outlines.models.Character`

template_name = `'fiction_outlines/character_list.html'`

```
class fiction_outlines.views.CharacterUpdateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.PrefetchRelatedMixin, django.
    views.generic.edit.UpdateView

    Generic update view for character.

    context_object_name = 'character'

    form_class
        alias of fiction_outlines.forms.CharacterForm

    get_form_kwargs()
        Return the keyword arguments for instantiating the form.

    get_success_url()
        Return the URL to redirect to after processing a valid form.

    model
        alias of fiction_outlines.models.Character

    permission_required = 'fiction_outlines.edit_character'

    pk_url_kwarg = 'character'

    prefetch_related = ['series']

    success_url = None

    template_name = 'fiction_outlines/character_update.html'

class fiction_outlines.views.LocationCreateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, django.views.generic.
    edit.CreateView

    Generic view for creating locations

    form_class
        alias of fiction_outlines.forms.LocationForm

    form_valid(form)
        If the form is valid, save the associated model.

    get_context_data(**kwargs)
        Insert the form into the context dict.

    get_form_kwargs()
        Return the keyword arguments for instantiating the form.

    get_success_url()
        Return the URL to redirect to after processing a valid form.

    model
        alias of fiction_outlines.models.Location

    success_url = None

    template_name = 'fiction_outlines/location_create.html'

class fiction_outlines.views.LocationDeleteView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.DeleteView

    Generic view for deleting locations.

    context_object_name = 'location'
```

```
get_success_url()

model
    alias of fiction_outlines.models.Location

permission_required = 'fiction_outlines.delete_location'

pk_url_kwarg = 'location'

success_url = None

template_name = 'fiction_outlines/location_delete.html'

class fiction_outlines.views.LocationDetailView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.PrefetchRelatedMixin, django.
    views.generic.detail.DetailView

    Generic view for location details.

    context_object_name = 'location'

    model
        alias of fiction_outlines.models.Location

    permission_required = 'fiction_outlines.view_location'

    pk_url_kwarg = 'location'

    prefetch_related = ['series', 'locationinstance_set', 'locationinstance_set__outline']

    template_name = 'fiction_outlines/location_detail.html'

class fiction_outlines.views.LocationInstanceCreateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.CreateView

    Generic view for creating a location instance on a outline.

    dispatch(request, *args, **kwargs)

    form_class
        alias of fiction_outlines.forms.LocationInstanceForm

    form_valid(form)
        If the form is valid, save the associated model.

    get_context_data(**kwargs)
        Insert the form into the context dict.

    get_form_kwargs()
        Return the keyword arguments for instantiating the form.

    get_permission_object()
        Override this method to provide the object to check for permission against. By default uses self.
        get_object() as provided by SingleObjectMixin. Returns None if there's no get_object
        method.

    get_success_url()
        Return the URL to redirect to after processing a valid form.

    model
        alias of fiction_outlines.models.LocationInstance

    permission_required = 'fiction_outlines.edit_location'
```

```
    success_url = None

    template_name = 'fiction_outlines/location_instance_create.html'

class fiction_outlines.views.LocationInstanceDeleteView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.edit.DeleteView

    Generic delete view for Location Instance.

    context_object_name = 'location_instance'

    dispatch(request, *args, **kwargs)

    get_success_url()

    model
        alias of fiction_outlines.models.LocationInstance

    permission_required = 'fiction_outlines.delete_location_instance'

    pk_url_kwarg = 'instance'

    prefetch_related = ['arcelementnode_set', 'storyelementnode_set']

    select_related = ['location', 'outline']

    success_url = None

    template_name = 'fiction_outlines/location_instance_delete.html'

class fiction_outlines.views.LocationInstanceDetailView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.detail.DetailView

    Generic view for a location instance detail view.

    context_object_name = 'location_instance'

    dispatch(request, *args, **kwargs)

    get_permission_object()
        Override this method to provide the object to check for permission against. By default uses self.
        get_object() as provided by SingleObjectMixin. Returns None if there's no get_object
        method.

    model
        alias of fiction_outlines.models.LocationInstance

    permission_required = 'fiction_outlines.view_location'

    pk_url_kwarg = 'instance'

    prefetch_related = ['arcelementnode_set', 'storyelementnode_set']

    select_related = ['location', 'outline']

    template_name = 'fiction_outlines/location_instance_detail.html'

class fiction_outlines.views.LocationInstanceListView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.list.ListView

    Generic view for looking at all location instances for a location.
```

```
context_object_name = 'location_instance_list'

dispatch(request, *args, **kwargs)

get_context_data(**kwargs)
    Get the context for this view.

get_permission_object()
    Override this method to provide the object to check for permission against. By default uses self.get_object() as provided by SingleObjectMixin. Returns None if there's no get_object method.

get_queryset()
    Return the list of items for this view.

    The return value must be an iterable and may be an instance of QuerySet in which case QuerySet specific behavior will be enabled.

model
    alias of fiction_outlines.models.LocationInstance

permission_required = 'fiction_outlines.view_location'

template_name = 'fiction_outlines/location_instance_list.html'

class fiction_outlines.views.LocationInstanceUpdateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, django.views.generic.edit.UpdateView

    Generic view for updating a location instance. Not used since there are not details. But it's here if you want to subclass LocationInstance and customize it.

    context_object_name = 'location_instance'

    dispatch(request, *args, **kwargs)

    form_class
        alias of fiction_outlines.forms.LocationInstanceForm

    form_valid(form)
        If the form is valid, save the associated model.

    get_context_data(**kwargs)
        Insert the form into the context dict.

    get_form_kwargs()
        Return the keyword arguments for instantiating the form.

    get_success_url()
        Return the URL to redirect to after processing a valid form.

    model
        alias of fiction_outlines.models.LocationInstance

    permission_required = 'fiction_outlines.edit_location_instance'

    pk_url_kwarg = 'instance'

    select_related = ['location', 'outline']

    success_url = None

    template_name = 'fiction_outlines/location_instance_update.html'
```



```
class fiction_outlines.views.LocationListView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, django.views.generic.
    list.ListView

    Generic view for locations.

    context_object_name = 'location_list'

    get_queryset()
        Return the list of items for this view.

        The return value must be an iterable and may be an instance of QuerySet in which case QuerySet specific
        behavior will be enabled.

    model
        alias of fiction_outlines.models.Location

    template_name = 'fiction_outlines/location_list.html'

class fiction_outlines.views.LocationUpdateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.UpdateView

    Generic view for updating locations.

    context_object_name = 'location'

    form_class
        alias of fiction_outlines.forms.LocationForm

    get_form_kwargs()
        Return the keyword arguments for instantiating the form.

    get_success_url()
        Return the URL to redirect to after processing a valid form.

    model
        alias of fiction_outlines.models.Location

    permission_required = 'fiction_outlines.edit_location'

    pk_url_kwarg = 'location'

    success_url = None

    template_name = 'fiction_outlines/location_update.html'

class fiction_outlines.views.OutlineCreateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, django.views.generic.
    edit.CreateView

    Generic view for creating initial outline.

    form_class
        alias of fiction_outlines.forms.OutlineForm

    form_valid(form)
        If the form is valid, save the associated model.

    get_form_kwargs()
        Return the keyword arguments for instantiating the form.

    get_success_url()
        Return the URL to redirect to after processing a valid form.
```

```
model
    alias of fiction_outlines.models.Outline

success_url = None

template_name = 'fiction_outlines/outline_create.html'

class fiction_outlines.views.OutlineDeleteView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.DeleteView

    Generic delete view for an outline.

    context_object_name = 'outline'

    model
        alias of fiction_outlines.models.Outline

    permission_required = 'fiction_outlines.delete_outline'

    pk_url_kwarg = 'outline'

    success_url = '/fiction-outlines/outlines/'

    template_name = 'fiction_outlines/outline_delete.html'

class fiction_outlines.views.OutlineDetailView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.detail.DetailView

    Generic view for Outline detail

    context_object_name = 'outline'

    model
        alias of fiction_outlines.models.Outline

    permission_required = 'fiction_outlines.view_outline'

    pk_url_kwarg = 'outline'

    prefetch_related = ['arc_set', 'characterinstance_set', 'locationinstance_set', 'story
    select_related = ['series']

    template_name = 'fiction_outlines/outline_detail.html'

class fiction_outlines.views.OutlineExport(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.detail.DetailView

    Generic view to get an export of an outline record.

    Takes a url kwarg of outline as the pk of the fiction_outlines.models.Outline The url kwarg of
    format determines the type returned. Current supported formats are opml, json, or md.

    context_object_name = 'outline'

    default_format = 'json'

    dispatch(request, *args, **kwargs)

    get_context_data(**kwargs)
        Insert the single object into the context dict.
```

```

model
    alias of fiction_outlines.models.Outline

not_implemented (context, **response_kwargs)
    If DEBUG: raise NotImplemented Exception. If not, raise 404. :raises: 'django.http.Http404' if produc-
    tion environment. :raises: 'NotImplementedError' if settings.DEBUG is True

permission_required = 'fiction_outlines.view_outline'

pk_url_kwarg = 'outline'

prefetch_related = ['arc_set', 'storyelementnode_set', 'characterinstance_set', 'chara

render_to_response (context, **response_kwargs)
    Compares requested format to supported formats and routes the response.

    Attribute switcher A dictionary of format types and their respective response methods.

return_json_response (context, **request_kwargs)
    Returns detailed outline structure as django.http.JsonResponse.

return_md_response (context, **response_kwargs)
    Returns the outline as a single markdown file.

return_opml_response (context, **response_kwargs)
    Returns export data as an opml file.

select_related = ['series', 'user']

template_name = 'fiction_outlines/outline.opml'

class fiction_outlines.views.OutlineListView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, braces.views._queries.
SelectRelatedMixin, braces.views._queries.PrefetchRelatedMixin, django.
views.generic.list.ListView

    Generic view for Outline Outline list

context_object_name = 'outline_list'

get_queryset ()
    Return the list of items for this view.

    The return value must be an iterable and may be an instance of QuerySet in which case QuerySet specific
    behavior will be enabled.

model
    alias of fiction_outlines.models.Outline

prefetch_related = ['arc_set', 'storyelementnode_set', 'characterinstance_set', 'locat

select_related = ['series']

template_name = 'fiction_outlines/outline_list.html'

class fiction_outlines.views.OutlineUpdateView (**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
PermissionRequiredMixin, django.views.generic.edit.UpdateView

    Generic update view for outline details.

context_object_name = 'outline'

form_class
    alias of fiction_outlines.forms.OutlineForm

```

```
get_form_kwargs()
    Return the keyword arguments for instantiating the form.

get_success_url()
    Return the URL to redirect to after processing a valid form.

model
    alias of fiction_outlines.models.Outline

permission_required = 'fiction_outlines.edit_outline'

pk_url_kwarg = 'outline'

success_url = None

template_name = 'fiction_outlines/outline_update.html'

class fiction_outlines.views.SeriesCreateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, django.views.generic.
    edit.CreateView

    Generic view for creating series object.

    fields = ['title', 'description', 'tags']

    form_valid(form)
        Override to ensure we can add the user to the record.

    get_success_url()
        Return the URL to redirect to after processing a valid form.

    model
        alias of fiction_outlines.models.Series

    success_url = None

    template_name = 'fiction_outlines/series_create.html'

class fiction_outlines.views.SeriesDeleteView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.DeleteView

    Generic view for deleting a series.

    context_object_name = 'series'

    model
        alias of fiction_outlines.models.Series

    permission_required = 'fiction_outlines.delete_series'

    pk_url_kwarg = 'series'

    success_url = '/fiction-outlines/series/'

    template_name = 'fiction_outlines/series_delete.html'

class fiction_outlines.views.SeriesDetailView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.PrefetchRelatedMixin, django.
    views.generic.detail.DetailView

    Generic view to see series details.

    context_object_name = 'series'
```

```

model
    alias of fiction_outlines.models.Series

permission_required = 'fiction_outlines.view_series'

pk_url_kwarg = 'series'

prefetch_related = ['character_set', 'location_set', 'outline_set']

template_name = 'fiction_outlines/series_detail.html'

class fiction_outlines.views.SeriesListView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, django.views.generic.
    list.ListView

    Generic view for viewing a list of series objects.

    context_object_name = 'series_list'

    get_queryset()
        Return the list of items for this view.

        The return value must be an iterable and may be an instance of QuerySet in which case QuerySet specific
        behavior will be enabled.

    model
        alias of fiction_outlines.models.Series

    template_name = 'fiction_outlines/series_list.html'

class fiction_outlines.views.SeriesUpdateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.PrefetchRelatedMixin, django.
    views.generic.edit.UpdateView

    Generic view for updating a series object.

    context_object_name = 'series'

    fields = ['title', 'description', 'tags']

    get_success_url()
        Return the URL to redirect to after processing a valid form.

    model
        alias of fiction_outlines.models.Series

    permission_required = 'fiction_outlines.edit_series'

    pk_url_kwarg = 'series'

    prefetch_related = ['character_set', 'location_set', 'outline_set']

    success_url = None

    template_name = 'fiction_outlines/series_update.html'

class fiction_outlines.views.StoryNodeCreateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, django.views.generic.edit.CreateView

    Creation view for a story node. Assumes the target and pos have been passed as kwargs.

    dispatch(request, *args, **kwargs)

    form_class
        alias of fiction_outlines.forms.StoryNodeForm

```

```
form_valid(form)
    If the form is valid, save the associated model.

get_form_kwargs()
    Return the keyword arguments for instantiating the form.

get_permission_object()
    Override this method to provide the object to check for permission against. By default uses self.
    get_object() as provided by SingleObjectMixin. Returns None if there's no get_object
    method.

get_success_url()
    Return the URL to redirect to after processing a valid form.

model
    alias of fiction_outlines.models.StoryElementNode

permission_required = 'fiction_outlines.edit_outline'

success_url = None

template_name = 'fiction_outlines/storynode_create.html'

class fiction_outlines.views.StoryNodeDeleteView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.edit.DeleteView

    Generic view for deleting a story node.

    context_object_name = 'storynode'

    dispatch(request, *args, **kwargs)

    get_success_url()

    model
        alias of fiction_outlines.models.StoryElementNode

    permission_required = 'fiction_outlines.delete_story_node'

    pk_url_kwarg = 'storynode'

    prefetch_related = ['arcelementnode_set', 'assoc_characters', 'assoc_locations']

    select_related = ['outline']

    success_url = None

    template_name = 'fiction_outlines/storynode_delete.html'

class fiction_outlines.views.StoryNodeDetailView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.
    PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.
    views._queries.PrefetchRelatedMixin, django.views.generic.detail.DetailView

    View for looking at the details of an atomic story node as opposed to the whole tree.

    context_object_name = 'storynode'

    model
        alias of fiction_outlines.models.StoryElementNode

    permission_required = 'fiction_outlines.view_story_node'

    pk_url_kwarg = 'storynode'
```

```
prefetch_related = ['arcelementnode_set', 'assoc_characters', 'assoc_locations']
select_related = ['outline']
template_name = 'fiction_outlines/storynode_detail.html'

class fiction_outlines.views.StoryNodeMoveView(**kwargs)
    Bases: fiction_outlines.views.StoryNodeUpdateView
    View for executing a move method on an arcnode.

    form_class
        alias of django.forms.widgets.StoryElementNodeForm

    form_valid(form)
        If the form is valid, save the associated model.

    get_form_kwargs()
        Return the keyword arguments for instantiating the form.

    success_url = None

    template_name = 'fiction_outlines/storynode_move.html'

class fiction_outlines.views.StoryNodeUpdateView(**kwargs)
    Bases: django.contrib.auth.mixins.LoginRequiredMixin, rules.contrib.views.PermissionRequiredMixin, braces.views._queries.SelectRelatedMixin, braces.views._queries.PrefetchRelatedMixin, django.views.generic.edit.UpdateView
    View for doing basic updates to a story node, but not regarding its position in the tree.

    context_object_name = 'storynode'

    form_class
        alias of fiction_outlines.forms.StoryNodeForm

    form_valid(form)
        If the form is valid, save the associated model.

    get_form_kwargs()
        Return the keyword arguments for instantiating the form.

    get_success_url()
        Return the URL to redirect to after processing a valid form.

    model
        alias of fiction_outlines.models.StoryElementNode

    permission_required = 'fiction_outlines.edit_story_node'

    pk_url_kwarg = 'storynode'

    prefetch_related = ['arcelementnode_set', 'assoc_characters', 'assoc_locations']
    select_related = ['outline']
    success_url = None
    template_name = 'fiction_outlines/storynode_update.html'
```

10.1.9 Module contents

fiction_outlines is a reusable Django app for managing manuscript outlines.

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

11.1 Types of Contributions

11.1.1 Report Bugs

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

11.1.2 Fix Bugs

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

11.1.3 Implement Features

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

11.1.4 Write Documentation

Django Fiction Outlines 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.

11.1.5 Submit Feedback

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

11.2 Get Started!

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

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

```
$ git clone git@github.com:your_name_here/django-fiction-outlines.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/  
$ pipenv install --dev
```

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, including testing other Python versions with `tox`:

```
$ flake8 fiction_outlines tests  
$ pytest  
$ tox
```

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.

11.3 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 3.5 and up. Check https://travis-ci.org/maceoutliner/django-fiction-outlines/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`.

11.4 Appropriate Conduct

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

Contributor Covenant Code of Conduct

12.1 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.

12.2 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

12.3 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.

12.4 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.

12.5 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.

12.6 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>

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.

CHAPTER 14

Credits

14.1 Development Lead

- Daniel Andrlik <daniel@andrlik.org>

14.2 Contributors

None yet. Why not be the first?

15.1 0.4.0 (2022-03-17)

- Now requires Django > 3.2. Compatible through Django 4.0
- Compatible with Python 3.9 and 3.10.

15.2 0.3.1 (2018-10-16)

- Now compatible with both Python 3.7 and 3.6

15.3 0.3.0 (2018-08-08)

- Support for Django 2.1

15.4 0.2.2 (2018-04-19)

- Bug fix: override model_utils until my submitted bug fix for created/modified timestamps is merged upstream.

15.5 0.2.1 (2018-04-14)

- Add created and modified auto timestamps to all tree models.

15.6 0.2.0 (2018-04-13)

- Add export functions. A view is provided for users to export outlines as either OPML, JSON, or Markdown documents.

15.7 0.1.5 (2018-04-09)

- Improvements to length estimate calculation.
- Improvements to test coverage.

15.8 0.1.4 (2018-04-07)

- Hotfix release for tag field issue.

15.9 0.1.3 (2018-04-07)

- Bugfix release for migrations

15.10 0.1.2 (2018-04-02)

- Bugfix release.

15.11 0.1.1 (2018-04-01)

- First release on PyPI.

f

- `fiction_outlines`, [75](#)
- `fiction_outlines.apps`, [33](#)
- `fiction_outlines.forms`, [31](#)
- `fiction_outlines.models`, [33](#)
- `fiction_outlines.receivers`, [54](#)
- `fiction_outlines.signals`, [56](#)
- `fiction_outlines.urls`, [56](#)
- `fiction_outlines.views`, [56](#)

A

abstract (*fiction_outlines.models.TimeStampedModel.Meta*
 attribute), 51
 add_child() (*fiction_outlines.models.ArcElementNode*
 method), 16, 36
 add_child() (*fiction_outlines.models.StoryElementNode*
 method), 18, 49
 add_sibling() (*fiction_outlines.models.ArcElementNode*
 method), 16, 36
 add_sibling() (*fiction_outlines.models.StoryElementNode*
 method), 18, 49
 all_characters (*fiction_outlines.models.StoryElementNode*
 attribute), 49
 all_characters() (*fiction_outlines.models.StoryElementNode*
 method), 17
 all_locations (*fiction_outlines.models.StoryElementNode*
 attribute), 49
 all_locations() (*fiction_outlines.models.StoryElementNode*
 method), 17
 antagonist (*fiction_outlines.models.CharacterInstance*
 attribute), 40
 Arc (*class in fiction_outlines.models*), 14, 33
 arc (*fiction_outlines.models.ArcElementNode attribute*),
 36
 Arc.DoesNotExist, 34
 Arc.MultipleObjectsReturned, 34
 arc_element_type (*fiction_outlines.models.ArcElementNode*
 attribute), 36
 arc_id (*fiction_outlines.models.ArcElementNode* *at-*
 tribute), 36
 arc_node_edit_add_missing_characters_and_locations_to_related_story_node()
 (*fiction_outlines.receivers method*), 23
 arc_node_edit_add_missing_characters_and_locations_
 (in module *fiction_outlines.receivers*), 54
 arc_root_node (*fiction_outlines.models.Arc* *at-*
 tribute), 34
 arc_root_node() (*fiction_outlines.models.Arc*
 method), 14
 arc_set (*fiction_outlines.models.Outline attribute*), 45
 ArcCreateView (*class in fiction_outlines.views*), 29,
 56
 ArcDeleteView (*class in fiction_outlines.views*), 29,
 57
 ArcDetailView (*class in fiction_outlines.views*), 29,
 57
 ArcElementNode (*class in fiction_outlines.models*),
 15, 35
 ArcElementNode.DoesNotExist, 35
 ArcElementNode.MultipleObjectsReturned,
 36
 arcelementnode_set (*fiction_outlines.models.Arc*
 attribute), 34
 arcelementnode_set (*fiction_outlines.models.CharacterInstance*
 attribute), 40
 arcelementnode_set (*fiction_outlines.models.LocationInstance* *at-*
 tribute), 43
 arcelementnode_set (*fiction_outlines.models.StoryElementNode*
 attribute), 49
 ArcGenerationError, 37
 ArcIntegrityError, 37
 ArcListView (*class in fiction_outlines.views*), 28, 57
 ArcNodeCreateView (*class in fic-*
 tion_outlines.views), 29, 58
 ArcNodeDeleteView (*class in fic-*
 tion_outlines.views), 29, 58
 ArcNodeDetailView (*class in fic-*
 tion_outlines.views), 29, 59
 ArcNodeMoveView (*class in fiction_outlines.views*),
 29, 59

ArcNodeUpdateView (class in *fiction_outlines.views*), 29, 60

ArcUpdateView (class in *fiction_outlines.views*), 29, 60

assoc_characters (fiction_outlines.models.ArcElementNode attribute), 36

assoc_characters (fiction_outlines.models.StoryElementNode attribute), 50

assoc_locations (fiction_outlines.models.ArcElementNode attribute), 36

assoc_locations (fiction_outlines.models.StoryElementNode attribute), 50

AutoLastModifiedField (class in *fiction_outlines.models*), 37

C

Character (class in *fiction_outlines.models*), 11, 37

character (fiction_outlines.models.CharacterInstance attribute), 40

Character.DoesNotExist, 38

Character.MultipleObjectsReturned, 38

character_id (fiction_outlines.models.CharacterInstance attribute), 40

character_set (fiction_outlines.models.Series attribute), 47

CharacterCreateView (class in *fiction_outlines.views*), 26, 61

CharacterDeleteView (class in *fiction_outlines.views*), 26, 61

CharacterDetailView (class in *fiction_outlines.views*), 26, 61

CharacterForm (class in *fiction_outlines.forms*), 31

CharacterInstance (class in *fiction_outlines.models*), 12, 39

CharacterInstance.DoesNotExist, 39

CharacterInstance.MultipleObjectsReturned, 40

characterinstance_set (fiction_outlines.models.Character attribute), 38

characterinstance_set (fiction_outlines.models.Outline attribute), 45

CharacterInstanceCreateView (class in *fiction_outlines.views*), 26, 62

CharacterInstanceDeleteView (class in *fiction_outlines.views*), 27, 62

CharacterInstanceDetailView (class in *fiction_outlines.views*), 26, 63

CharacterInstanceForm (class in *fiction_outlines.forms*), 31

CharacterInstanceListView (class in *fiction_outlines.views*), 26, 63

CharacterInstanceUpdateView (class in *fiction_outlines.views*), 26, 64

CharacterListView (class in *fiction_outlines.views*), 26, 64

CharacterUpdateView (class in *fiction_outlines.views*), 26, 64

content_object (fiction_outlines.models.UUIDCharacterTag attribute), 52

content_object (fiction_outlines.models.UUIDLocationTag attribute), 53

content_object (fiction_outlines.models.UUIDOutlineTag attribute), 53

content_type (fiction_outlines.models.UUIDCharacterTag attribute), 52

content_type (fiction_outlines.models.UUIDLocationTag attribute), 53

content_type (fiction_outlines.models.UUIDOutlineTag attribute), 54

context_object_name (fiction_outlines.views.ArcDeleteView attribute), 57

context_object_name (fiction_outlines.views.ArcDetailView attribute), 57

context_object_name (fiction_outlines.views.ArcListView attribute), 57

context_object_name (fiction_outlines.views.ArcNodeDeleteView attribute), 58

context_object_name (fiction_outlines.views.ArcNodeDetailView attribute), 59

context_object_name (fiction_outlines.views.ArcNodeMoveView attribute), 59

context_object_name (fiction_outlines.views.ArcNodeUpdateView attribute), 60

context_object_name (fiction_outlines.views.ArcUpdateView attribute), 60

context_object_name (fiction_outlines.views.CharacterDeleteView attribute), 61

context_object_name (fiction_outlines.views.CharacterDetailView attribute), 61

context_object_name (fiction_outlines.views.CharacterInstanceDeleteView attribute), 61

| | |
|---|---|
| <code>tion_outlines.views.CharacterInstanceDeleteView</code> <i>attribute</i>), 62 | <code>tion_outlines.views.OutlineUpdateView</code> <i>attribute</i>), 71 |
| <code>context_object_name</code> (<code>fiction_outlines.views.CharacterInstanceDetailView</code> <i>attribute</i>), 63 | <code>context_object_name</code> (<code>fiction_outlines.views.SeriesDeleteView</code> <i>attribute</i>), 72 |
| <code>context_object_name</code> (<code>fiction_outlines.views.CharacterInstanceListView</code> <i>attribute</i>), 63 | <code>context_object_name</code> (<code>fiction_outlines.views.SeriesDetailView</code> <i>attribute</i>), 72 |
| <code>context_object_name</code> (<code>fiction_outlines.views.CharacterInstanceUpdateView</code> <i>attribute</i>), 64 | <code>context_object_name</code> (<code>fiction_outlines.views.SeriesListView</code> <i>attribute</i>), 73 |
| <code>context_object_name</code> (<code>fiction_outlines.views.CharacterListView</code> <i>attribute</i>), 64 | <code>context_object_name</code> (<code>fiction_outlines.views.SeriesUpdateView</code> <i>attribute</i>), 73 |
| <code>context_object_name</code> (<code>fiction_outlines.views.CharacterUpdateView</code> <i>attribute</i>), 65 | <code>context_object_name</code> (<code>fiction_outlines.views.StoryNodeDeleteView</code> <i>attribute</i>), 74 |
| <code>context_object_name</code> (<code>fiction_outlines.views.LocationDeleteView</code> <i>attribute</i>), 65 | <code>context_object_name</code> (<code>fiction_outlines.views.StoryNodeDetailView</code> <i>attribute</i>), 74 |
| <code>context_object_name</code> (<code>fiction_outlines.views.LocationDetailView</code> <i>attribute</i>), 66 | <code>context_object_name</code> (<code>fiction_outlines.views.StoryNodeUpdateView</code> <i>attribute</i>), 75 |
| <code>context_object_name</code> (<code>fiction_outlines.views.LocationInstanceDeleteView</code> <i>attribute</i>), 67 | <code>create_arc()</code> (<code>fiction_outlines.models.Outline</code> <i>method</i>), 14, 45 |
| <code>context_object_name</code> (<code>fiction_outlines.views.LocationInstanceDetailView</code> <i>attribute</i>), 67 | <code>current_errors</code> (<code>fiction_outlines.models.Arc</code> <i>attribute</i>), 34 |
| <code>context_object_name</code> (<code>fiction_outlines.views.LocationInstanceListView</code> <i>attribute</i>), 67 | <code>current_errors()</code> (<code>fiction_outlines.models.Arc</code> <i>method</i>), 14 |
| <code>context_object_name</code> (<code>fiction_outlines.views.LocationInstanceUpdateView</code> <i>attribute</i>), 68 | D |
| <code>context_object_name</code> (<code>fiction_outlines.views.LocationListView</code> <i>attribute</i>), 69 | <code>default_format</code> (<code>fiction_outlines.views.OutlineExport</code> <i>attribute</i>), 70 |
| <code>context_object_name</code> (<code>fiction_outlines.views.LocationUpdateView</code> <i>attribute</i>), 69 | <code>delete()</code> (<code>fiction_outlines.views.ArcNodeDeleteView</code> <i>method</i>), 59 |
| <code>context_object_name</code> (<code>fiction_outlines.views.OutlineDeleteView</code> <i>attribute</i>), 70 | <code>description</code> (<code>fiction_outlines.models.ArcElementNode</code> <i>attribute</i>), 36 |
| <code>context_object_name</code> (<code>fiction_outlines.views.OutlineDetailView</code> <i>attribute</i>), 70 | <code>description</code> (<code>fiction_outlines.models.Character</code> <i>attribute</i>), 38 |
| <code>context_object_name</code> (<code>fiction_outlines.views.OutlineExport</code> <i>attribute</i>), 70 | <code>description</code> (<code>fiction_outlines.models.Location</code> <i>attribute</i>), 42 |
| <code>context_object_name</code> (<code>fiction_outlines.views.OutlineListView</code> <i>attribute</i>), 71 | <code>description</code> (<code>fiction_outlines.models.Outline</code> <i>attribute</i>), 45 |
| <code>context_object_name</code> (<code>fiction_outlines.views.OutlineUpdateView</code> <i>attribute</i>), 71 | <code>description</code> (<code>fiction_outlines.models.Series</code> <i>attribute</i>), 48 |
| | <code>description</code> (<code>fiction_outlines.models.StoryElementNode</code> <i>attribute</i>), 50 |
| | <code>dispatch()</code> (<code>fiction_outlines.views.ArcCreateView</code> <i>method</i>), 56 |
| | <code>dispatch()</code> (<code>fiction_outlines.views.ArcDeleteView</code> <i>method</i>), 57 |
| | <code>dispatch()</code> (<code>fiction_outlines.views.ArcDetailView</code> <i>method</i>), 57 |

`dispatch()` (`fiction_outlines.views.ArcListView` `form_class` (`fiction_outlines.views.ArcNodeCreateView` `attribute`), 58
`dispatch()` (`fiction_outlines.views.ArcNodeCreateView` `form_class` (`fiction_outlines.views.ArcNodeMoveView` `attribute`), 58
`dispatch()` (`fiction_outlines.views.ArcNodeDeleteView` `form_class` (`fiction_outlines.views.ArcNodeUpdateView` `attribute`), 59
`dispatch()` (`fiction_outlines.views.CharacterInstanceCreateView` `form_class` (`fiction_outlines.views.CharacterCreateView` `attribute`), 62
`dispatch()` (`fiction_outlines.views.CharacterInstanceDeleteView` `form_class` (`fiction_outlines.views.CharacterInstanceCreateView` `attribute`), 62
`dispatch()` (`fiction_outlines.views.CharacterInstanceDetailView` `form_class` (`fiction_outlines.views.CharacterInstanceUpdateView` `attribute`), 63
`dispatch()` (`fiction_outlines.views.CharacterInstanceListView` `form_class` (`fiction_outlines.views.CharacterUpdateView` `attribute`), 63
`dispatch()` (`fiction_outlines.views.CharacterInstanceUpdateView` `form_class` (`fiction_outlines.views.LocationCreateView` `attribute`), 64
`dispatch()` (`fiction_outlines.views.LocationInstanceCreateView` `form_class` (`fiction_outlines.views.LocationInstanceCreateView` `attribute`), 66
`dispatch()` (`fiction_outlines.views.LocationInstanceDeleteView` `form_class` (`fiction_outlines.views.LocationInstanceUpdateView` `attribute`), 67
`dispatch()` (`fiction_outlines.views.LocationInstanceDetailView` `form_class` (`fiction_outlines.views.LocationUpdateView` `attribute`), 67
`dispatch()` (`fiction_outlines.views.LocationInstanceListView` `form_class` (`fiction_outlines.views.OutlineCreateView` `attribute`), 68
`dispatch()` (`fiction_outlines.views.LocationInstanceUpdateView` `form_class` (`fiction_outlines.views.OutlineUpdateView` `attribute`), 68
`dispatch()` (`fiction_outlines.views.OutlineExport` `form_class` (`fiction_outlines.views.StoryNodeCreateView` `attribute`), 70
`dispatch()` (`fiction_outlines.views.StoryNodeCreateView` `form_class` (`fiction_outlines.views.StoryNodeMoveView` `attribute`), 73
`dispatch()` (`fiction_outlines.views.StoryNodeDeleteView` `form_class` (`fiction_outlines.views.StoryNodeUpdateView` `attribute`), 74

F

`fetch_arc_errors()` (`fiction_outlines.models.Arc` `method`), 15, 34
`fiction_outlines` (module), 75
`fiction_outlines.apps` (module), 33
`fiction_outlines.forms` (module), 31
`fiction_outlines.models` (module), 11, 33
`fiction_outlines.receivers` (module), 54
`fiction_outlines.signals` (module), 56
`fiction_outlines.urls` (module), 56
`fiction_outlines.views` (module), 25, 56
`FictionOutlinesConfig` (class in `fiction_outlines.apps`), 33
`fields` (`fiction_outlines.views.ArcCreateView` `attribute`), 56
`fields` (`fiction_outlines.views.ArcUpdateView` `attribute`), 60
`fields` (`fiction_outlines.views.SeriesCreateView` `attribute`), 72
`fields` (`fiction_outlines.views.SeriesUpdateView` `attribute`), 73
`form_valid()` (`fiction_outlines.views.ArcCreateView` `method`), 56
`form_valid()` (`fiction_outlines.views.ArcNodeCreateView` `method`), 58
`form_valid()` (`fiction_outlines.views.ArcNodeDeleteView` `method`), 59
`form_valid()` (`fiction_outlines.views.ArcNodeMoveView` `method`), 59
`form_valid()` (`fiction_outlines.views.ArcNodeUpdateView` `method`), 60
`form_valid()` (`fiction_outlines.views.CharacterCreateView` `method`), 61
`form_valid()` (`fiction_outlines.views.CharacterInstanceCreateView` `method`), 62
`form_valid()` (`fiction_outlines.views.CharacterInstanceUpdateView` `method`), 64
`form_valid()` (`fiction_outlines.views.LocationCreateView` `method`), 65
`form_valid()` (`fiction_outlines.views.LocationInstanceCreateView` `method`), 66
`form_valid()` (`fiction_outlines.views.LocationInstanceUpdateView` `method`), 68

`form_valid()` (`fiction_outlines.views.OutlineCreateView` method), 69
`form_valid()` (`fiction_outlines.views.SeriesCreateView` method), 72
`form_valid()` (`fiction_outlines.views.StoryNodeCreateView` method), 73
`form_valid()` (`fiction_outlines.views.StoryNodeMoveView` method), 75
`form_valid()` (`fiction_outlines.views.StoryNodeUpdateView` method), 75
G
`generate_headline_from_description()` (`fiction_outlines.receivers` method), 23
`generate_headline_from_description()` (in module `fiction_outlines.receivers`), 54
`generate_template_arc_tree()` (`fiction_outlines.models.Arc` method), 15, 34
`GenericArcSequenceError`, 41
`get_absolute_url()` (`fiction_outlines.models.Arc` method), 34
`get_absolute_url()` (`fiction_outlines.models.ArcElementNode` method), 36
`get_absolute_url()` (`fiction_outlines.models.Character` method), 38
`get_absolute_url()` (`fiction_outlines.models.CharacterInstance` method), 40
`get_absolute_url()` (`fiction_outlines.models.Location` method), 42
`get_absolute_url()` (`fiction_outlines.models.LocationInstance` method), 43
`get_absolute_url()` (`fiction_outlines.models.Outline` method), 45
`get_absolute_url()` (`fiction_outlines.models.Series` method), 48
`get_absolute_url()` (`fiction_outlines.models.StoryElementNode` method), 50
`get_arc_element_type_display()` (`fiction_outlines.models.ArcElementNode` method), 36
`get_context_data()` (`fiction_outlines.views.ArcCreateView` method), 56
`get_context_data()` (`fiction_outlines.views.ArcDetailView` method), 57
`get_context_data()` (`fiction_outlines.views.ArcListView` method),
`get_context_data()` (`fiction_outlines.views.CharacterInstanceCreateView` method), 62
`get_context_data()` (`fiction_outlines.views.CharacterInstanceListView` method), 63
`get_context_data()` (`fiction_outlines.views.LocationCreateView` method), 65
`get_context_data()` (`fiction_outlines.views.LocationInstanceCreateView` method), 66
`get_context_data()` (`fiction_outlines.views.LocationInstanceListView` method), 68
`get_context_data()` (`fiction_outlines.views.LocationInstanceUpdateView` method), 68
`get_context_data()` (`fiction_outlines.views.OutlineExport` method), 70
`get_form_kwargs()` (`fiction_outlines.views.ArcNodeCreateView` method), 58
`get_form_kwargs()` (`fiction_outlines.views.ArcNodeMoveView` method), 59
`get_form_kwargs()` (`fiction_outlines.views.ArcNodeUpdateView` method), 60
`get_form_kwargs()` (`fiction_outlines.views.CharacterCreateView` method), 61
`get_form_kwargs()` (`fiction_outlines.views.CharacterInstanceCreateView` method), 62
`get_form_kwargs()` (`fiction_outlines.views.CharacterInstanceUpdateView` method), 64
`get_form_kwargs()` (`fiction_outlines.views.CharacterUpdateView` method), 65
`get_form_kwargs()` (`fiction_outlines.views.LocationCreateView` method), 65
`get_form_kwargs()` (`fiction_outlines.views.LocationInstanceCreateView` method), 66
`get_form_kwargs()` (`fiction_outlines.views.LocationInstanceUpdateView` method), 68
`get_form_kwargs()` (`fiction_outlines.views.LocationUpdateView` method), 68

| | | | |
|--------------------------------------|---|---|---|
| <i>method</i>), 69 | | <i>tion_outlines.models.CharacterInstance</i> | |
| <code>get_form_kwargs()</code> | (fiction_outlines.views.OutlineCreateView | <i>method</i>), 40 | |
| <i>method</i>), 69 | | <code>get_next_by_modified()</code> | (fiction_outlines.models.Location |
| <code>get_form_kwargs()</code> | (fiction_outlines.views.OutlineUpdateView | <i>method</i>), | |
| <i>method</i>), 71 | | 42 | |
| <code>get_form_kwargs()</code> | (fiction_outlines.views.StoryNodeCreateView | <code>get_next_by_modified()</code> | (fiction_outlines.models.LocationInstance |
| <i>method</i>), 74 | | <i>method</i>), 43 | |
| <code>get_form_kwargs()</code> | (fiction_outlines.views.StoryNodeMoveView | <code>get_next_by_modified()</code> | (fiction_outlines.models.Outline |
| <i>method</i>), 75 | | <i>method</i>), 45 | |
| <code>get_form_kwargs()</code> | (fiction_outlines.views.StoryNodeUpdateView | <code>get_next_by_modified()</code> | (fiction_outlines.models.Series |
| <i>method</i>), 75 | | <i>method</i>), 48 | |
| <code>get_mace_type_display()</code> | (fiction_outlines.models.Arc | <code>get_next_by_modified()</code> | (fiction_outlines.models.StoryElementNode |
| <i>method</i>), 34 | | <i>method</i>), 50 | |
| <code>get_next_by_created()</code> | (fiction_outlines.models.Arc | <code>get_next_by_modified()</code> | (fiction_outlines.models.TimeStampedModel |
| <i>method</i>), 34 | | <i>method</i>), 51 | |
| <code>get_next_by_created()</code> | (fiction_outlines.models.ArcElementNode | <code>get_permission_object()</code> | (fiction_outlines.views.ArcCreateView |
| <i>method</i>), 37 | | <i>method</i>), | |
| <code>get_next_by_created()</code> | (fiction_outlines.models.Character | 56 | |
| <i>method</i>), 38 | | <code>get_permission_object()</code> | (fiction_outlines.views.ArcListView |
| <code>get_next_by_created()</code> | (fiction_outlines.models.CharacterInstance | <i>method</i>), | |
| <i>method</i>), 40 | | 58 | |
| <code>get_next_by_created()</code> | (fiction_outlines.models.Location | <code>get_permission_object()</code> | (fiction_outlines.views.ArcNodeCreateView |
| <i>method</i>), 42 | | <i>method</i>), 58 | |
| <code>get_next_by_created()</code> | (fiction_outlines.models.LocationInstance | <code>get_permission_object()</code> | (fiction_outlines.views.CharacterInstanceDetailView |
| <i>method</i>), 43 | | <i>method</i>), 63 | |
| <code>get_next_by_created()</code> | (fiction_outlines.models.Outline | <code>get_permission_object()</code> | (fiction_outlines.views.CharacterInstanceListView |
| <i>method</i>), 45 | | <i>method</i>), 63 | |
| <code>get_next_by_created()</code> | (fiction_outlines.models.Series | <code>get_permission_object()</code> | (fiction_outlines.views.CharacterInstanceUpdateView |
| <i>method</i>), 48 | | <i>method</i>), 64 | |
| <code>get_next_by_created()</code> | (fiction_outlines.models.StoryElementNode | <code>get_permission_object()</code> | (fiction_outlines.views.LocationInstanceCreateView |
| <i>method</i>), 50 | | <i>method</i>), 66 | |
| <code>get_next_by_created()</code> | (fiction_outlines.models.TimeStampedModel | <code>get_permission_object()</code> | (fiction_outlines.views.LocationInstanceDetailView |
| <i>method</i>), 51 | | <i>method</i>), 67 | |
| <code>get_next_by_modified()</code> | (fiction_outlines.models.Arc | <code>get_permission_object()</code> | (fiction_outlines.views.LocationInstanceListView |
| <i>method</i>), 34 | | <i>method</i>), 68 | |
| <code>get_next_by_modified()</code> | (fiction_outlines.models.ArcElementNode | <code>get_permission_object()</code> | (fiction_outlines.views.StoryNodeCreateView |
| <i>method</i>), 37 | | <i>method</i>), 74 | |
| <code>get_next_by_modified()</code> | (fiction_outlines.models.Character | <code>get_previous_by_created()</code> | (fiction_outlines.models.Arc |
| <i>method</i>), 38 | | <i>method</i>), 34 | |
| <code>get_next_by_modified()</code> | (fiction_outlines.models.CharacterInstance | <code>get_previous_by_created()</code> | (fiction_outlines.models.ArcElementNode |
| <i>method</i>), 40 | | <i>method</i>), 37 | |
| <code>get_next_by_modified()</code> | (fiction_outlines.models.CharacterInstance | <code>get_previous_by_created()</code> | (fiction_outlines.models.CharacterInstance |
| <i>method</i>), 40 | | <i>method</i>), 40 | |

| | | |
|---|------------------|--|
| <i>tion_outlines.models.Character</i> 38 | <i>method</i>), | <i>tion_outlines.views.CharacterListView</i> <i>method</i>), 64 |
| <i>get_previous_by_created()</i> <i>tion_outlines.models.CharacterInstance</i> <i>method</i>), 40 | (<i>fic-</i> | <i>get_queryset()</i> (<i>fic-</i> <i>tion_outlines.views.LocationInstanceListView</i> <i>method</i>), 68 |
| <i>get_previous_by_created()</i> <i>tion_outlines.models.Location</i> 42 | <i>method</i>), | <i>get_queryset()</i> (<i>fic-</i> <i>tion_outlines.views.LocationListView</i> <i>method</i>), 69 |
| <i>get_previous_by_created()</i> <i>tion_outlines.models.LocationInstance</i> <i>method</i>), 43 | (<i>fic-</i> | <i>get_queryset()</i> (<i>fic-</i> <i>tion_outlines.views.OutlineListView</i> <i>method</i>), 71 |
| <i>get_previous_by_created()</i> <i>tion_outlines.models.Outline</i> <i>method</i>), 45 | (<i>fic-</i> | <i>get_queryset()</i> (<i>fic-</i> <i>tion_outlines.views.SeriesListView</i> <i>method</i>), 73 |
| <i>get_previous_by_created()</i> <i>tion_outlines.models.Series</i> <i>method</i>), 48 | (<i>fic-</i> | <i>get_story_element_type_display()</i> (<i>fic-</i> <i>tion_outlines.models.StoryElementNode</i> <i>method</i>), 50 |
| <i>get_previous_by_created()</i> <i>tion_outlines.models.StoryElementNode</i> <i>method</i>), 50 | (<i>fic-</i> | <i>get_success_url()</i> (<i>fic-</i> <i>tion_outlines.views.ArcCreateView</i> <i>method</i>), 56 |
| <i>get_previous_by_created()</i> <i>tion_outlines.models.TimeStampedModel</i> <i>method</i>), 51 | (<i>fic-</i> | <i>get_success_url()</i> (<i>fic-</i> <i>tion_outlines.views.ArcDeleteView</i> <i>method</i>), 57 |
| <i>get_previous_by_modified()</i> <i>tion_outlines.models.Arc</i> <i>method</i>), 34 | (<i>fic-</i> | <i>get_success_url()</i> (<i>fic-</i> <i>tion_outlines.views.ArcNodeCreateView</i> <i>method</i>), 58 |
| <i>get_previous_by_modified()</i> <i>tion_outlines.models.ArcElementNode</i> <i>method</i>), 37 | (<i>fic-</i> | <i>get_success_url()</i> (<i>fic-</i> <i>tion_outlines.views.ArcNodeDeleteView</i> <i>method</i>), 59 |
| <i>get_previous_by_modified()</i> <i>tion_outlines.models.Character</i> 38 | <i>method</i>), | <i>get_success_url()</i> (<i>fic-</i> <i>tion_outlines.views.ArcNodeMoveView</i> <i>method</i>), 59 |
| <i>get_previous_by_modified()</i> <i>tion_outlines.models.CharacterInstance</i> <i>method</i>), 40 | (<i>fic-</i> | <i>get_success_url()</i> (<i>fic-</i> <i>tion_outlines.views.ArcNodeUpdateView</i> <i>method</i>), 60 |
| <i>get_previous_by_modified()</i> <i>tion_outlines.models.Location</i> 42 | <i>method</i>), | <i>get_success_url()</i> (<i>fic-</i> <i>tion_outlines.views.ArcUpdateView</i> <i>method</i>), 60 |
| <i>get_previous_by_modified()</i> <i>tion_outlines.models.LocationInstance</i> <i>method</i>), 43 | (<i>fic-</i> | <i>get_success_url()</i> (<i>fic-</i> <i>tion_outlines.views.CharacterCreateView</i> <i>method</i>), 61 |
| <i>get_previous_by_modified()</i> <i>tion_outlines.models.Outline</i> <i>method</i>), 45 | (<i>fic-</i> | <i>get_success_url()</i> (<i>fic-</i> <i>tion_outlines.views.CharacterInstanceCreateView</i> <i>method</i>), 62 |
| <i>get_previous_by_modified()</i> <i>tion_outlines.models.Series</i> <i>method</i>), 48 | (<i>fic-</i> | <i>get_success_url()</i> (<i>fic-</i> <i>tion_outlines.views.CharacterInstanceDeleteView</i> <i>method</i>), 62 |
| <i>get_previous_by_modified()</i> <i>tion_outlines.models.StoryElementNode</i> <i>method</i>), 50 | (<i>fic-</i> | <i>get_success_url()</i> (<i>fic-</i> <i>tion_outlines.views.CharacterInstanceUpdateView</i> <i>method</i>), 64 |
| <i>get_previous_by_modified()</i> <i>tion_outlines.models.TimeStampedModel</i> <i>method</i>), 51 | (<i>fic-</i> | <i>get_success_url()</i> (<i>fic-</i> <i>tion_outlines.views.CharacterUpdateView</i> <i>method</i>), 65 |
| <i>get_queryset()</i> (<i>fiction_outlines.views.ArcListView</i> <i>method</i>), 58 | (<i>fic-</i> | <i>get_success_url()</i> (<i>fic-</i> |
| <i>get_queryset()</i> (<i>fiction_outlines.views.CharacterInstanceListView</i> <i>method</i>), 63 | (<i>fic-</i> | <i>get_success_url()</i> (<i>fic-</i> |
| <i>get_queryset()</i> | (<i>fic-</i> | <i>get_success_url()</i> (<i>fic-</i> |

| | |
|--|---|
| <code>tion_outlines.views.LocationCreateView</code> <code>method</code>), 65 | <code>id</code> (<code>fiction_outlines.models.LocationInstance</code> attribute), 43 |
| <code>get_success_url()</code> <code>tion_outlines.views.LocationDeleteView</code> <code>method</code>), 65 | (<code>fiction_outlines.models.Outline</code> attribute), 45 <code>id</code> (<code>fiction_outlines.models.Series</code> attribute), 48 <code>id</code> (<code>fiction_outlines.models.StoryElementNode</code> attribute), 50 |
| <code>get_success_url()</code> <code>tion_outlines.views.LocationInstanceCreateView</code> <code>method</code>), 66 | <code>id</code> (<code>fiction_outlines.models.UUIDCharacterTag</code> attribute), 52 |
| <code>get_success_url()</code> <code>tion_outlines.views.LocationInstanceDeleteView</code> <code>method</code>), 67 | <code>id</code> (<code>fiction_outlines.models.UUIDLocationTag</code> attribute), 53 <code>id</code> (<code>fiction_outlines.models.UUIDOutlineTag</code> attribute), 54 |
| <code>get_success_url()</code> <code>tion_outlines.views.LocationInstanceUpdateView</code> <code>method</code>), 68 | <code>impact_rating</code> (<code>fiction_outlines.models.StoryElementNode</code> attribute), 50 |
| <code>get_success_url()</code> <code>tion_outlines.views.LocationUpdateView</code> <code>method</code>), 69 | <code>impact_rating()</code> (<code>fiction_outlines.models.StoryElementNode</code> method), 17 |
| <code>get_success_url()</code> <code>tion_outlines.views.OutlineCreateView</code> <code>method</code>), 69 | <code>is_milestone</code> (<code>fiction_outlines.models.ArcElementNode</code> attribute), 37 |
| <code>get_success_url()</code> <code>tion_outlines.views.OutlineUpdateView</code> <code>method</code>), 72 | <code>is_milestone()</code> (<code>fiction_outlines.models.ArcElementNode</code> method), 16 |
| <code>get_success_url()</code> <code>tion_outlines.views.SeriesCreateView</code> method), 72 | L |
| <code>get_success_url()</code> <code>tion_outlines.views.SeriesUpdateView</code> method), 73 | <code>length_estimate</code> (<code>fiction_outlines.models.Outline</code> attribute), 46 <code>length_estimate()</code> (<code>fiction_outlines.models.Outline</code> method), 13 |
| <code>get_success_url()</code> <code>tion_outlines.views.StoryNodeCreateView</code> <code>method</code>), 74 | <code>Location</code> (class in <code>fiction_outlines.models</code>), 12, 41 <code>location</code> (<code>fiction_outlines.models.LocationInstance</code> attribute), 44 |
| <code>get_success_url()</code> <code>tion_outlines.views.StoryNodeDeleteView</code> <code>method</code>), 74 | <code>Location.DoesNotExist</code> , 41 <code>Location.MultipleObjectsReturned</code> , 41 <code>location_id</code> (<code>fiction_outlines.models.LocationInstance</code> attribute), 44 |
| <code>get_success_url()</code> <code>tion_outlines.views.StoryNodeUpdateView</code> <code>method</code>), 75 | <code>location_set</code> (<code>fiction_outlines.models.Series</code> attribute), 48 <code>LocationCreateView</code> (class in <code>fiction_outlines.views</code>), 27, 65 <code>LocationDeleteView</code> (class in <code>fiction_outlines.views</code>), 27, 65 <code>LocationDetailView</code> (class in <code>fiction_outlines.views</code>), 27, 66 <code>LocationForm</code> (class in <code>fiction_outlines.forms</code>), 31 <code>LocationInstance</code> (class in <code>fiction_outlines.models</code>), 13, 43 <code>LocationInstance.DoesNotExist</code> , 43 <code>LocationInstance.MultipleObjectsReturned</code> , 43 <code>locationinstance_set</code> (<code>fiction_outlines.models.Location</code> attribute), 42 |
| H | |
| <code>has_permission()</code> <code>tion_outlines.views.CharacterInstanceCreateView</code> <code>method</code>), 62 | |
| <code>headline</code> (<code>fiction_outlines.models.ArcElementNode</code> attribute), 37 | |
| I | |
| <code>id</code> (<code>fiction_outlines.models.Arc</code> attribute), 34 | |
| <code>id</code> (<code>fiction_outlines.models.ArcElementNode</code> attribute), 37 | |
| <code>id</code> (<code>fiction_outlines.models.Character</code> attribute), 38 | |
| <code>id</code> (<code>fiction_outlines.models.CharacterInstance</code> attribute), 40 | |
| <code>id</code> (<code>fiction_outlines.models.Location</code> attribute), 42 | |

[locationinstance_set](#) (*fiction_outlines.models.Outline* attribute), [46](#)
[LocationInstanceCreateView](#) (class in *fiction_outlines.views*), [27](#), [66](#)
[LocationInstanceDeleteView](#) (class in *fiction_outlines.views*), [28](#), [67](#)
[LocationInstanceDetailView](#) (class in *fiction_outlines.views*), [27](#), [67](#)
[LocationInstanceForm](#) (class in *fiction_outlines.forms*), [31](#)
[LocationInstanceListView](#) (class in *fiction_outlines.views*), [27](#), [67](#)
[LocationInstanceUpdateView](#) (class in *fiction_outlines.views*), [27](#), [68](#)
[LocationListView](#) (class in *fiction_outlines.views*), [27](#), [68](#)
[LocationUpdateView](#) (class in *fiction_outlines.views*), [27](#), [69](#)

M

[mace_type](#) (*fiction_outlines.models.Arc* attribute), [34](#)
[main_character](#) (*fiction_outlines.models.CharacterInstance* attribute), [40](#)
[milestone_seq](#) (*fiction_outlines.models.ArcElementNode* attribute), [37](#)
[milestone_seq\(\)](#) (*fiction_outlines.models.ArcElementNode* method), [16](#)
[MilestoneDepthError](#), [44](#)
[MilestoneSequenceError](#), [44](#)
[model](#) (*fiction_outlines.views.ArcCreateView* attribute), [57](#)
[model](#) (*fiction_outlines.views.ArcDeleteView* attribute), [57](#)
[model](#) (*fiction_outlines.views.ArcDetailView* attribute), [57](#)
[model](#) (*fiction_outlines.views.ArcListView* attribute), [58](#)
[model](#) (*fiction_outlines.views.ArcNodeCreateView* attribute), [58](#)
[model](#) (*fiction_outlines.views.ArcNodeDeleteView* attribute), [59](#)
[model](#) (*fiction_outlines.views.ArcNodeDetailView* attribute), [59](#)
[model](#) (*fiction_outlines.views.ArcNodeMoveView* attribute), [60](#)
[model](#) (*fiction_outlines.views.ArcNodeUpdateView* attribute), [60](#)
[model](#) (*fiction_outlines.views.ArcUpdateView* attribute), [60](#)
[model](#) (*fiction_outlines.views.CharacterCreateView* attribute), [61](#)
[model](#) (*fiction_outlines.views.CharacterDeleteView* attribute), [61](#)
[model](#) (*fiction_outlines.views.CharacterDetailView* attribute), [61](#)
[model](#) (*fiction_outlines.views.CharacterInstanceCreateView* attribute), [62](#)
[model](#) (*fiction_outlines.views.CharacterInstanceDeleteView* attribute), [62](#)
[model](#) (*fiction_outlines.views.CharacterInstanceDetailView* attribute), [63](#)
[model](#) (*fiction_outlines.views.CharacterInstanceListView* attribute), [63](#)
[model](#) (*fiction_outlines.views.CharacterInstanceUpdateView* attribute), [64](#)
[model](#) (*fiction_outlines.views.CharacterListView* attribute), [64](#)
[model](#) (*fiction_outlines.views.CharacterUpdateView* attribute), [65](#)
[model](#) (*fiction_outlines.views.LocationCreateView* attribute), [65](#)
[model](#) (*fiction_outlines.views.LocationDeleteView* attribute), [66](#)
[model](#) (*fiction_outlines.views.LocationDetailView* attribute), [66](#)
[model](#) (*fiction_outlines.views.LocationInstanceCreateView* attribute), [66](#)
[model](#) (*fiction_outlines.views.LocationInstanceDeleteView* attribute), [67](#)
[model](#) (*fiction_outlines.views.LocationInstanceDetailView* attribute), [67](#)
[model](#) (*fiction_outlines.views.LocationInstanceListView* attribute), [68](#)
[model](#) (*fiction_outlines.views.LocationInstanceUpdateView* attribute), [68](#)
[model](#) (*fiction_outlines.views.LocationListView* attribute), [69](#)
[model](#) (*fiction_outlines.views.LocationUpdateView* attribute), [69](#)
[model](#) (*fiction_outlines.views.OutlineCreateView* attribute), [69](#)
[model](#) (*fiction_outlines.views.OutlineDeleteView* attribute), [70](#)
[model](#) (*fiction_outlines.views.OutlineDetailView* attribute), [70](#)
[model](#) (*fiction_outlines.views.OutlineExport* attribute), [70](#)
[model](#) (*fiction_outlines.views.OutlineListView* attribute), [71](#)
[model](#) (*fiction_outlines.views.OutlineUpdateView* attribute), [72](#)
[model](#) (*fiction_outlines.views.SeriesCreateView* attribute), [72](#)
[model](#) (*fiction_outlines.views.SeriesDeleteView* attribute), [72](#)

`model` (`fiction_outlines.views.SeriesDetailView` attribute), 72
`model` (`fiction_outlines.views.SeriesListView` attribute), 73
`model` (`fiction_outlines.views.SeriesUpdateView` attribute), 73
`model` (`fiction_outlines.views.StoryNodeCreateView` attribute), 74
`model` (`fiction_outlines.views.StoryNodeDeleteView` attribute), 74
`model` (`fiction_outlines.views.StoryNodeDetailView` attribute), 74
`model` (`fiction_outlines.views.StoryNodeUpdateView` attribute), 75
`move()` (`fiction_outlines.models.ArcElementNode` method), 16
`move()` (`fiction_outlines.models.StoryElementNode` method), 17, 51

N

`name` (`fiction_outlines.apps.FictionOutlinesConfig` attribute), 33
`name` (`fiction_outlines.models.Arc` attribute), 34
`name` (`fiction_outlines.models.Character` attribute), 38
`name` (`fiction_outlines.models.Location` attribute), 42
`name` (`fiction_outlines.models.StoryElementNode` attribute), 51
`node_deletion_safe()` (`fiction_outlines.views.ArcNodeDeleteView` method), 59
`not_implemented()` (`fiction_outlines.views.OutlineExport` method), 71

O

`objects` (`fiction_outlines.models.Arc` attribute), 34
`objects` (`fiction_outlines.models.Character` attribute), 38
`objects` (`fiction_outlines.models.CharacterInstance` attribute), 40
`objects` (`fiction_outlines.models.Location` attribute), 42
`objects` (`fiction_outlines.models.LocationInstance` attribute), 44
`objects` (`fiction_outlines.models.Outline` attribute), 46
`objects` (`fiction_outlines.models.Series` attribute), 48
`objects` (`fiction_outlines.models.UUIDCharacterTag` attribute), 52
`objects` (`fiction_outlines.models.UUIDLocationTag` attribute), 53
`objects` (`fiction_outlines.models.UUIDOutlineTag` attribute), 54
`obstacle` (`fiction_outlines.models.CharacterInstance` attribute), 40

`Outline` (class in `fiction_outlines.models`), 13, 44
`outline` (`fiction_outlines.models.Arc` attribute), 34
`outline` (`fiction_outlines.models.CharacterInstance` attribute), 40
`outline` (`fiction_outlines.models.LocationInstance` attribute), 44
`outline` (`fiction_outlines.models.StoryElementNode` attribute), 51
`outline` (`fiction_outlines.views.CharacterInstanceCreateView` attribute), 62
`Outline.DoesNotExist`, 45
`Outline.MultipleObjectsReturned`, 45
`outline_id` (`fiction_outlines.models.Arc` attribute), 35
`outline_id` (`fiction_outlines.models.CharacterInstance` attribute), 41
`outline_id` (`fiction_outlines.models.LocationInstance` attribute), 44
`outline_id` (`fiction_outlines.models.StoryElementNode` attribute), 51
`outline_set` (`fiction_outlines.models.Series` attribute), 48
`OutlineCreateView` (class in `fiction_outlines.views`), 28, 69
`OutlineDeleteView` (class in `fiction_outlines.views`), 28, 70
`OutlineDetailView` (class in `fiction_outlines.views`), 28, 70
`OutlineExport` (class in `fiction_outlines.views`), 28, 70
`OutlineForm` (class in `fiction_outlines.forms`), 31
`OutlineListView` (class in `fiction_outlines.views`), 28, 71
`OutlineMoveNodeForm` (class in `fiction_outlines.forms`), 31
`OutlineUpdateView` (class in `fiction_outlines.views`), 28, 71

P

`parent_outline` (`fiction_outlines.models.ArcElementNode` attribute), 37
`parent_outline()` (`fiction_outlines.models.ArcElementNode` method), 16
`path()` (in module `fiction_outlines.urls`), 56
`permission_required` (`fiction_outlines.views.ArcCreateView` attribute), 57
`permission_required` (`fiction_outlines.views.ArcDeleteView` attribute), 57
`permission_required` (`fiction_outlines.views.ArcDetailView` attribute), 57

| | | | |
|---|---|--|---|
| permission_required <i>tion_outlines.views.ArcListView</i> 58 | (fiction_outlines.views.ArcListView attribute), | permission_required <i>tion_outlines.views.LocationInstanceDeleteView</i> attribute), 67 | (fiction_outlines.views.LocationInstanceDeleteView attribute), 67 |
| permission_required <i>tion_outlines.views.ArcNodeCreateView</i> attribute), 58 | (fiction_outlines.views.ArcNodeCreateView attribute), | permission_required <i>tion_outlines.views.LocationInstanceDetailView</i> attribute), 67 | (fiction_outlines.views.LocationInstanceDetailView attribute), 67 |
| permission_required <i>tion_outlines.views.ArcNodeDeleteView</i> tribute), 59 | (fiction_outlines.views.ArcNodeDeleteView at-tribute), 59 | permission_required <i>tion_outlines.views.LocationInstanceListView</i> attribute), 68 | (fiction_outlines.views.LocationInstanceListView attribute), 68 |
| permission_required <i>tion_outlines.views.ArcNodeDetailView</i> tribute), 59 | (fiction_outlines.views.ArcNodeDetailView at-tribute), 59 | permission_required <i>tion_outlines.views.LocationInstanceUpdateView</i> attribute), 68 | (fiction_outlines.views.LocationInstanceUpdateView attribute), 68 |
| permission_required <i>tion_outlines.views.ArcNodeMoveView</i> tribute), 60 | (fiction_outlines.views.ArcNodeMoveView at-tribute), 60 | permission_required <i>tion_outlines.views.LocationUpdateView</i> attribute), 69 | (fiction_outlines.views.LocationUpdateView attribute), 69 |
| permission_required <i>tion_outlines.views.ArcNodeUpdateView</i> attribute), 60 | (fiction_outlines.views.ArcNodeUpdateView attribute), | permission_required <i>tion_outlines.views.OutlineDeleteView</i> tribute), 70 | (fiction_outlines.views.OutlineDeleteView at-tribute), 70 |
| permission_required <i>tion_outlines.views.ArcUpdateView</i> 60 | (fiction_outlines.views.ArcUpdateView attribute), 60 | permission_required <i>tion_outlines.views.OutlineDetailView</i> tribute), 70 | (fiction_outlines.views.OutlineDetailView at-tribute), 70 |
| permission_required <i>tion_outlines.views.CharacterDeleteView</i> attribute), 61 | (fiction_outlines.views.CharacterDeleteView attribute), | permission_required <i>tion_outlines.views.OutlineExport</i> 71 | (fiction_outlines.views.OutlineExport attribute), 71 |
| permission_required <i>tion_outlines.views.CharacterDetailView</i> attribute), 62 | (fiction_outlines.views.CharacterDetailView attribute), | permission_required <i>tion_outlines.views.OutlineUpdateView</i> tribute), 72 | (fiction_outlines.views.OutlineUpdateView at-tribute), 72 |
| permission_required <i>tion_outlines.views.CharacterInstanceCreateView</i> attribute), 62 | (fiction_outlines.views.CharacterInstanceCreateView attribute), | permission_required <i>tion_outlines.views.SeriesDeleteView</i> tribute), 72 | (fiction_outlines.views.SeriesDeleteView at-tribute), 72 |
| permission_required <i>tion_outlines.views.CharacterInstanceDeleteView</i> attribute), 62 | (fiction_outlines.views.CharacterInstanceDeleteView attribute), | permission_required <i>tion_outlines.views.SeriesDetailView</i> 73 | (fiction_outlines.views.SeriesDetailView attribute), 73 |
| permission_required <i>tion_outlines.views.CharacterInstanceDetailView</i> attribute), 63 | (fiction_outlines.views.CharacterInstanceDetailView attribute), | permission_required <i>tion_outlines.views.SeriesUpdateView</i> tribute), 73 | (fiction_outlines.views.SeriesUpdateView at-tribute), 73 |
| permission_required <i>tion_outlines.views.CharacterInstanceListView</i> attribute), 63 | (fiction_outlines.views.CharacterInstanceListView attribute), | permission_required <i>tion_outlines.views.StoryNodeCreateView</i> attribute), 74 | (fiction_outlines.views.StoryNodeCreateView attribute), 74 |
| permission_required <i>tion_outlines.views.CharacterInstanceUpdateView</i> attribute), 64 | (fiction_outlines.views.CharacterInstanceUpdateView attribute), | permission_required <i>tion_outlines.views.StoryNodeDeleteView</i> attribute), 74 | (fiction_outlines.views.StoryNodeDeleteView attribute), 74 |
| permission_required <i>tion_outlines.views.CharacterUpdateView</i> attribute), 65 | (fiction_outlines.views.CharacterUpdateView attribute), | permission_required <i>tion_outlines.views.StoryNodeDetailView</i> attribute), 74 | (fiction_outlines.views.StoryNodeDetailView attribute), 74 |
| permission_required <i>tion_outlines.views.LocationDeleteView</i> attribute), 66 | (fiction_outlines.views.LocationDeleteView attribute), | permission_required <i>tion_outlines.views.StoryNodeUpdateView</i> attribute), 75 | (fiction_outlines.views.StoryNodeUpdateView attribute), 75 |
| permission_required <i>tion_outlines.views.LocationDetailView</i> tribute), 66 | (fiction_outlines.views.LocationDetailView at-tribute), 66 | pk_url_kwarg (<i>fiction_outlines.views.ArcDeleteView</i> attribute), 57 | (fiction_outlines.views.ArcDeleteView attribute), 57 |
| permission_required <i>tion_outlines.views.LocationInstanceCreateView</i> attribute), 66 | (fiction_outlines.views.LocationInstanceCreateView attribute), | pk_url_kwarg (<i>fiction_outlines.views.ArcDetailView</i> attribute), 57 | (fiction_outlines.views.ArcDetailView attribute), 57 |
| | | pk_url_kwarg (<i>fiction_outlines.views.ArcNodeDeleteView</i> attribute), 59 | (fiction_outlines.views.ArcNodeDeleteView attribute), 59 |

`pk_url_kwarg` (`fiction_outlines.views.ArcNodeDetailView` `attribute`), 41
`attribute`), 59 `pre_save()` (`fiction_outlines.models.AutoLastModifiedField`
`pk_url_kwarg` (`fiction_outlines.views.ArcNodeMoveView` `method`), 37
`attribute`), 60 `prefetch_related` (`fic-`
`pk_url_kwarg` (`fiction_outlines.views.ArcNodeUpdateView` `tion_outlines.views.ArcDetailView` `attribute`),
`attribute`), 60 57
`pk_url_kwarg` (`fiction_outlines.views.ArcUpdateView` `prefetch_related` (`fic-`
`attribute`), 61 `tion_outlines.views.ArcNodeDeleteView` `at-`
`pk_url_kwarg` (`fiction_outlines.views.CharacterDeleteView` `tribute`), 59
`attribute`), 61 `prefetch_related` (`fic-`
`pk_url_kwarg` (`fiction_outlines.views.CharacterDetailView` `tion_outlines.views.ArcNodeDetailView` `at-`
`attribute`), 62 `tribute`), 59
`pk_url_kwarg` (`fiction_outlines.views.CharacterInstanceDeleteView` `prefetch_related` (`fic-`
`attribute`), 62 `tion_outlines.views.ArcNodeUpdateView`
`pk_url_kwarg` (`fiction_outlines.views.CharacterInstanceDetailView` `attribute`), 60
`attribute`), 63 `prefetch_related` (`fic-`
`pk_url_kwarg` (`fiction_outlines.views.CharacterInstanceUpdateView` `tion_outlines.views.CharacterDeleteView`
`attribute`), 64 `attribute`), 61
`pk_url_kwarg` (`fiction_outlines.views.CharacterUpdateView` `prefetch_related` (`fic-`
`attribute`), 65 `tion_outlines.views.CharacterDetailView`
`pk_url_kwarg` (`fiction_outlines.views.LocationDeleteView` `attribute`), 62
`attribute`), 66 `prefetch_related` (`fic-`
`pk_url_kwarg` (`fiction_outlines.views.LocationDetailView` `tion_outlines.views.CharacterInstanceDeleteView`
`attribute`), 66 `attribute`), 62
`pk_url_kwarg` (`fiction_outlines.views.LocationInstanceDeleteView` `prefetch_related` (`fic-`
`attribute`), 67 `tion_outlines.views.CharacterInstanceDetailView`
`pk_url_kwarg` (`fiction_outlines.views.LocationInstanceDetailView` `attribute`), 63
`attribute`), 67 `prefetch_related` (`fic-`
`pk_url_kwarg` (`fiction_outlines.views.LocationInstanceUpdateView` `tion_outlines.views.CharacterUpdateView`
`attribute`), 68 `attribute`), 65
`pk_url_kwarg` (`fiction_outlines.views.LocationUpdateView` `prefetch_related` (`fic-`
`attribute`), 69 `tion_outlines.views.LocationDetailView` `at-`
`pk_url_kwarg` (`fiction_outlines.views.OutlineDeleteView` `tribute`), 66
`attribute`), 70 `prefetch_related` (`fic-`
`pk_url_kwarg` (`fiction_outlines.views.OutlineDetailView` `tion_outlines.views.LocationInstanceDeleteView`
`attribute`), 70 `attribute`), 67
`pk_url_kwarg` (`fiction_outlines.views.OutlineExport` `prefetch_related` (`fic-`
`attribute`), 71 `tion_outlines.views.LocationInstanceDetailView`
`pk_url_kwarg` (`fiction_outlines.views.OutlineUpdateView` `attribute`), 67
`attribute`), 72 `prefetch_related` (`fic-`
`pk_url_kwarg` (`fiction_outlines.views.SeriesDeleteView` `tion_outlines.views.OutlineDetailView` `at-`
`attribute`), 72 `tribute`), 70
`pk_url_kwarg` (`fiction_outlines.views.SeriesDetailView` `prefetch_related` (`fic-`
`attribute`), 73 `tion_outlines.views.OutlineExport` `attribute`),
`pk_url_kwarg` (`fiction_outlines.views.SeriesUpdateView` 71
`attribute`), 73 `prefetch_related` (`fic-`
`pk_url_kwarg` (`fiction_outlines.views.StoryNodeDeleteView` `tion_outlines.views.OutlineListView` `attribute`),
`attribute`), 74 71
`pk_url_kwarg` (`fiction_outlines.views.StoryNodeDetailView` `prefetch_related` (`fic-`
`attribute`), 74 `tion_outlines.views.SeriesDetailView` `attribute`),
`pk_url_kwarg` (`fiction_outlines.views.StoryNodeUpdateView` 73
`attribute`), 75 `prefetch_related` (`fic-`
`pov_character` (`fiction_outlines.models.CharacterInstance` `tion_outlines.views.SeriesUpdateView` `at-`
`tribute`), 73

[prefetch_related](#) ([fiction_outlines.views.StoryNodeDeleteView](#) attribute), [74](#)
[prefetch_related](#) ([fiction_outlines.views.StoryNodeDetailView](#) attribute), [74](#)
[prefetch_related](#) ([fiction_outlines.views.StoryNodeUpdateView](#) attribute), [75](#)
[protagonist](#) ([fiction_outlines.models.CharacterInstance](#) attribute), [41](#)

R

[ready\(\)](#) ([fiction_outlines.apps.FictionOutlinesConfig](#) method), [33](#)
[refresh_from_db\(\)](#) ([fiction_outlines.models.Arc](#) method), [15](#), [35](#)
[refresh_from_db\(\)](#) ([fiction_outlines.models.Outline](#) method), [13](#), [46](#)
[render_to_response\(\)](#) ([fiction_outlines.views.OutlineExport](#) method), [71](#)
[return_json_response\(\)](#) ([fiction_outlines.views.OutlineExport](#) method), [71](#)
[return_md_response\(\)](#) ([fiction_outlines.views.OutlineExport](#) method), [71](#)
[return_opml_response\(\)](#) ([fiction_outlines.views.OutlineExport](#) method), [71](#)

S

[select_related](#) ([fiction_outlines.views.ArcDetailView](#) attribute), [57](#)
[select_related](#) ([fiction_outlines.views.ArcNodeDeleteView](#) attribute), [59](#)
[select_related](#) ([fiction_outlines.views.ArcNodeDetailView](#) attribute), [59](#)
[select_related](#) ([fiction_outlines.views.ArcNodeUpdateView](#) attribute), [60](#)
[select_related](#) ([fiction_outlines.views.ArcUpdateView](#) attribute), [61](#)
[select_related](#) ([fiction_outlines.views.CharacterInstanceDeleteView](#) attribute), [63](#)
[select_related](#) ([fiction_outlines.views.CharacterInstanceDetailView](#) attribute), [63](#)
[select_related](#) ([fiction_outlines.views.CharacterInstanceUpdateView](#) attribute), [64](#)
[select_related](#) ([fiction_outlines.views.LocationInstanceDetailView](#) attribute), [67](#)
[select_related](#) ([fiction_outlines.views.LocationInstanceUpdateView](#) attribute), [68](#)
[select_related](#) ([fiction_outlines.views.OutlineDetailView](#) attribute), [70](#)
[select_related](#) ([fiction_outlines.views.OutlineExport](#) attribute), [71](#)
[select_related](#) ([fiction_outlines.views.OutlineListView](#) attribute), [71](#)
[select_related](#) ([fiction_outlines.views.StoryNodeDeleteView](#) attribute), [74](#)
[select_related](#) ([fiction_outlines.views.StoryNodeDetailView](#) attribute), [75](#)
[select_related](#) ([fiction_outlines.views.StoryNodeUpdateView](#) attribute), [75](#)
[select_related](#) ([fiction_outlines.views.LocationInstanceDeleteView](#) attribute), [67](#)
[Series](#) (class in [fiction_outlines.models](#)), [11](#), [47](#)
[series](#) ([fiction_outlines.models.Character](#) attribute), [38](#)
[series](#) ([fiction_outlines.models.Location](#) attribute), [42](#)
[series](#) ([fiction_outlines.models.Outline](#) attribute), [46](#)
[Series.DoesNotExist](#), [47](#)
[Series.MultipleObjectsReturned](#), [47](#)
[series_id](#) ([fiction_outlines.models.Outline](#) attribute), [46](#)
[SeriesCreateView](#) (class in [fiction_outlines.views](#)), [25](#), [72](#)
[SeriesDeleteView](#) (class in [fiction_outlines.views](#)), [25](#), [72](#)
[SeriesDetailView](#) (class in [fiction_outlines.views](#)), [25](#), [72](#)
[SeriesListView](#) (class in [fiction_outlines.views](#)), [25](#), [73](#)
[SeriesUpdateView](#) (class in [fiction_outlines.views](#)), [25](#), [73](#)
[steplen](#) ([fiction_outlines.models.ArcElementNode](#) attribute), [37](#)
[steplen](#) ([fiction_outlines.models.StoryElementNode](#) attribute), [51](#)

| | | |
|--|---|---|
| <code>story_element_node</code> | (fiction_outlines.models.ArcElementNode attribute), 37 | <code>success_url</code> (fiction_outlines.views.ArcNodeUpdateView attribute), 60 |
| <code>story_element_node_id</code> | (fiction_outlines.models.ArcElementNode attribute), 37 | <code>success_url</code> (fiction_outlines.views.ArcUpdateView attribute), 61 |
| <code>story_element_type</code> | (fiction_outlines.models.StoryElementNode attribute), 51 | <code>success_url</code> (fiction_outlines.views.CharacterCreateView attribute), 61 |
| <code>story_node_add_arc_element_update_characters_location</code> | (fiction_outlines.receivers method), 23 | <code>success_url</code> (fiction_outlines.views.CharacterDeleteView attribute), 61 |
| <code>story_node_add_arc_element_update_characters_location</code> | (in module fiction_outlines.receivers), 54 | <code>success_url</code> (fiction_outlines.views.CharacterInstanceCreateView attribute), 62 |
| <code>story_root_for_new_outline()</code> | (fiction_outlines.receivers method), 23 | <code>success_url</code> (fiction_outlines.views.CharacterInstanceDeleteView attribute), 63 |
| <code>story_root_for_new_outline()</code> | (in module fiction_outlines.receivers), 55 | <code>success_url</code> (fiction_outlines.views.CharacterInstanceUpdateView attribute), 64 |
| <code>story_tree_root</code> | (fiction_outlines.models.Outline attribute), 46 | <code>success_url</code> (fiction_outlines.views.CharacterUpdateView attribute), 65 |
| <code>story_tree_root()</code> | (fiction_outlines.models.Outline method), 13 | <code>success_url</code> (fiction_outlines.views.LocationCreateView attribute), 65 |
| <code>StoryElementNode</code> | (class in fiction_outlines.models), 17, 49 | <code>success_url</code> (fiction_outlines.views.LocationDeleteView attribute), 66 |
| <code>StoryElementNode.DoesNotExist</code> | 49 | <code>success_url</code> (fiction_outlines.views.LocationInstanceCreateView attribute), 66 |
| <code>StoryElementNode.MultipleObjectsReturned</code> | 49 | <code>success_url</code> (fiction_outlines.views.LocationInstanceDeleteView attribute), 67 |
| <code>storyelementnode_set</code> | (fiction_outlines.models.CharacterInstance attribute), 41 | <code>success_url</code> (fiction_outlines.views.LocationInstanceUpdateView attribute), 68 |
| <code>storyelementnode_set</code> | (fiction_outlines.models.LocationInstance attribute), 44 | <code>success_url</code> (fiction_outlines.views.LocationUpdateView attribute), 69 |
| <code>storyelementnode_set</code> | (fiction_outlines.models.Outline attribute), 46 | <code>success_url</code> (fiction_outlines.views.OutlineCreateView attribute), 70 |
| <code>StoryNodeCreateView</code> | (class in fiction_outlines.views), 29, 30, 73 | <code>success_url</code> (fiction_outlines.views.OutlineDeleteView attribute), 70 |
| <code>StoryNodeDeleteView</code> | (class in fiction_outlines.views), 30, 74 | <code>success_url</code> (fiction_outlines.views.OutlineUpdateView attribute), 72 |
| <code>StoryNodeDetailView</code> | (class in fiction_outlines.views), 30, 74 | <code>success_url</code> (fiction_outlines.views.SeriesCreateView attribute), 72 |
| <code>StoryNodeMoveView</code> | (class in fiction_outlines.views), 30, 75 | <code>success_url</code> (fiction_outlines.views.SeriesDeleteView attribute), 72 |
| <code>StoryNodeUpdateView</code> | (class in fiction_outlines.views), 30, 75 | <code>success_url</code> (fiction_outlines.views.SeriesUpdateView attribute), 73 |
| <code>success_url</code> | (fiction_outlines.views.ArcCreateView attribute), 57 | <code>success_url</code> (fiction_outlines.views.StoryNodeCreateView attribute), 74 |
| <code>success_url</code> | (fiction_outlines.views.ArcDeleteView attribute), 57 | <code>success_url</code> (fiction_outlines.views.StoryNodeDeleteView attribute), 74 |
| <code>success_url</code> | (fiction_outlines.views.ArcNodeCreateView attribute), 58 | <code>success_url</code> (fiction_outlines.views.StoryNodeMoveView attribute), 75 |
| <code>success_url</code> | (fiction_outlines.views.ArcNodeDeleteView attribute), 59 | <code>success_url</code> (fiction_outlines.views.StoryNodeUpdateView attribute), 75 |
| <code>success_url</code> | (fiction_outlines.views.ArcNodeMoveView attribute), 60 | <code>success_url</code> (fiction_outlines.views.UUIDCharacterTag attribute), 52 |
| | | <code>success_url</code> (fiction_outlines.views.UUIDLocationTag attribute), 53 |

tag (*fiction_outlines.models.UUIDOutlineTag* attribute), 54
 tagged_items (*fiction_outlines.models.Character* attribute), 39
 tagged_items (*fiction_outlines.models.Location* attribute), 42
 tagged_items (*fiction_outlines.models.Outline* attribute), 46
 tagged_items (*fiction_outlines.models.Series* attribute), 48
 tags (*fiction_outlines.models.Character* attribute), 39
 tags (*fiction_outlines.models.Location* attribute), 43
 tags (*fiction_outlines.models.Outline* attribute), 47
 tags (*fiction_outlines.models.Series* attribute), 48
 template_name (*fiction_outlines.views.ArcCreateView* attribute), 57
 template_name (*fiction_outlines.views.ArcDeleteView* attribute), 57
 template_name (*fiction_outlines.views.ArcDetailView* attribute), 57
 template_name (*fiction_outlines.views.ArcListView* attribute), 58
 template_name (*fiction_outlines.views.ArcNodeCreateView* attribute), 58
 template_name (*fiction_outlines.views.ArcNodeDeleteView* attribute), 59
 template_name (*fiction_outlines.views.ArcNodeDetailView* attribute), 59
 template_name (*fiction_outlines.views.ArcNodeMoveView* attribute), 60
 template_name (*fiction_outlines.views.ArcNodeUpdateView* attribute), 60
 template_name (*fiction_outlines.views.ArcUpdateView* attribute), 61
 template_name (*fiction_outlines.views.CharacterCreateView* attribute), 61
 template_name (*fiction_outlines.views.CharacterDeleteView* attribute), 61
 template_name (*fiction_outlines.views.CharacterDetailView* attribute), 62
 template_name (*fiction_outlines.views.CharacterInstanceCreateView* attribute), 62
 template_name (*fiction_outlines.views.CharacterInstanceDeleteView* attribute), 63
 template_name (*fiction_outlines.views.CharacterInstanceDetailView* attribute), 63
 template_name (*fiction_outlines.views.CharacterInstanceListView* attribute), 63
 template_name (*fiction_outlines.views.CharacterInstanceUpdateView* attribute), 64
 template_name (*fiction_outlines.views.CharacterListView* attribute), 64
 template_name (*fiction_outlines.views.CharacterUpdateView* attribute), 65
 template_name (*fiction_outlines.views.LocationCreateView* attribute), 65
 template_name (*fiction_outlines.views.LocationDeleteView* attribute), 66
 template_name (*fiction_outlines.views.LocationDetailView* attribute), 66
 template_name (*fiction_outlines.views.LocationInstanceCreateView* attribute), 67
 template_name (*fiction_outlines.views.LocationInstanceDeleteView* attribute), 67
 template_name (*fiction_outlines.views.LocationInstanceDetailView* attribute), 67
 template_name (*fiction_outlines.views.LocationInstanceListView* attribute), 68
 template_name (*fiction_outlines.views.LocationInstanceUpdateView* attribute), 68
 template_name (*fiction_outlines.views.LocationListView* attribute), 69
 template_name (*fiction_outlines.views.LocationUpdateView* attribute), 69
 template_name (*fiction_outlines.views.OutlineCreateView* attribute), 70
 template_name (*fiction_outlines.views.OutlineDeleteView* attribute), 70

tribute), 70
template_name (fiction_outlines.views.OutlineDetailView attribute), 70
template_name (fiction_outlines.views.OutlineExport attribute), 71
template_name (fiction_outlines.views.OutlineListView attribute), 71
template_name (fiction_outlines.views.OutlineUpdateView attribute), 72
template_name (fiction_outlines.views.SeriesCreateView attribute), 72
template_name (fiction_outlines.views.SeriesDeleteView attribute), 72
template_name (fiction_outlines.views.SeriesDetailView attribute), 73
template_name (fiction_outlines.views.SeriesListView attribute), 73
template_name (fiction_outlines.views.SeriesUpdateView attribute), 73
template_name (fiction_outlines.views.StoryNodeCreateView attribute), 74
template_name (fiction_outlines.views.StoryNodeDeleteView attribute), 74
template_name (fiction_outlines.views.StoryNodeDetailView attribute), 75
template_name (fiction_outlines.views.StoryNodeMoveView attribute), 75
template_name (fiction_outlines.views.StoryNodeUpdateView attribute), 75
TimeStampedModel (class in fiction_outlines.models), 51
TimeStampedModel.Meta (class in fiction_outlines.models), 51
title (fiction_outlines.models.Outline attribute), 47
title (fiction_outlines.models.Series attribute), 49

U

user (fiction_outlines.models.Character attribute), 39
user (fiction_outlines.models.Location attribute), 43
user (fiction_outlines.models.Outline attribute), 47
user (fiction_outlines.models.Series attribute), 49

user_id (fiction_outlines.models.Character attribute), 39
user_id (fiction_outlines.models.Location attribute), 43
user_id (fiction_outlines.models.Outline attribute), 47
user_id (fiction_outlines.models.Series attribute), 49
UUIDCharacterTag (class in fiction_outlines.models), 51
UUIDCharacterTag.DoesNotExist, 52
UUIDCharacterTag.MultipleObjectsReturned, 52
UUIDLocationTag (class in fiction_outlines.models), 52
UUIDLocationTag.DoesNotExist, 53
UUIDLocationTag.MultipleObjectsReturned, 53
UUIDOutlineTag (class in fiction_outlines.models), 53
UUIDOutlineTag.DoesNotExist, 53
UUIDOutlineTag.MultipleObjectsReturned, 53

V

validate_against_prohibited_actions() (in module fiction_outlines.receivers), 55
validate_arc_links_same_outline() (fiction_outlines.receivers method), 23
validate_arc_links_same_outline() (in module fiction_outlines.receivers), 55
validate_character_for_story_element() (fiction_outlines.receivers method), 23
validate_character_for_story_element() (in module fiction_outlines.receivers), 55
validate_character_instance_valid_for_arc() (fiction_outlines.receivers method), 23
validate_character_instance_valid_for_arc() (in module fiction_outlines.receivers), 55
validate_first_element() (fiction_outlines.models.Arc method), 15, 35
validate_generations() (fiction_outlines.models.Arc method), 15, 35
validate_generations_for_story_elements() (fiction_outlines.receivers method), 24
validate_generations_for_story_elements() (in module fiction_outlines.receivers), 55
validate_last_element() (fiction_outlines.models.Arc method), 15, 35
validate_location_for_story_element() (fiction_outlines.receivers method), 24
validate_location_for_story_element() (in module fiction_outlines.receivers), 55
validate_location_instance_valid_for_arc() (fiction_outlines.receivers method), 23

`validate_location_instance_valid_for_arc()`
 (in module *fiction_outlines.receivers*), [56](#)
`validate_milestones()` (fiction_outlines.models.Arc method), [15](#), [35](#)
`validate_nesting()` (fiction_outlines.models.Outline method), [14](#),
 [47](#)
`verbose_name` (fiction_outlines.apps.FictionOutlinesConfig
 attribute), [33](#)
`villain` (fiction_outlines.models.CharacterInstance
 attribute), [41](#)