
elanmanweb

Release 0.1

Mar 01, 2020

Contents:

1	eLanman web site documentation	1
1.1	Introduction	1
1.2	Getting Started with eLanman	2
1.3	website source code	3
1.4	Contributing to Open Source Projects	24
2	Indices and tables	29
3	Credits	31
	Python Module Index	33
	Index	35

1.1 Introduction

1.1.1 About

Placeholder for Intro section This document provides a set of best practices for open source contributions - bug reports, code submissions / pull requests, etc.

For the most part, these guidelines apply equally to *any* project regardless of programming language or topic. Where applicable, we outline where individual projects/languages may have additional requirements.

Naturally, this document is itself open source, and we encourage feedback & suggestions for improvement.

Sources

Currently this document draws from the contribution documentation for a handful of related Python open source projects: [Fabric](#), [Invoke](#), [Paramiko](#), etc.

It's expected that over time we will incorporate additional material from related attempts at consolidating this type of info. We'll update with a list here when that happens.

1.1.2 Submitting bugs

Due diligence

Before submitting a bug, please do the following:

- Perform **basic troubleshooting** steps:
 - **Make sure you're on the latest version.** If you're not on the most recent version, your problem may have been solved already! Upgrading is always the best first step.

- **Try older versions.** If you're already *on* the latest release, try rolling back a few minor versions (e.g. if on 1.7, try 1.5 or 1.6) and see if the problem goes away. This will help the devs narrow down when the problem first arose in the commit log.
- **Try switching up dependency versions.** If the software in question has dependencies (other libraries, etc) try upgrading/downgrading those as well.
- **Search the project's bug/issue tracker** to make sure it's not a known issue.
- If you don't find a pre-existing issue, consider **checking with the mailing list and/or IRC channel** in case the problem is non-bug-related.

1.2 Getting Started with eLanman

1.2.1 About

Place holder for Getting Started section This document provides a set of best practices for open source contributions - bug reports, code submissions / pull requests, etc.

For the most part, these guidelines apply equally to *any* project regardless of programming language or topic. Where applicable, we outline where individual projects/languages may have additional requirements.

Naturally, this document is itself open source, and we encourage feedback & suggestions for improvement.

Sources

Currently this document draws from the contribution documentation for a handful of related Python open source projects: [Fabric](#), [Invoke](#), [Paramiko](#), etc.

It's expected that over time we will incorporate additional material from related attempts at consolidating this type of info. We'll update with a list here when that happens.

1.2.2 Submitting bugs

Due diligence

Before submitting a bug, please do the following:

- Perform **basic troubleshooting** steps:
 - **Make sure you're on the latest version.** If you're not on the most recent version, your problem may have been solved already! Upgrading is always the best first step.
 - **Try older versions.** If you're already *on* the latest release, try rolling back a few minor versions (e.g. if on 1.7, try 1.5 or 1.6) and see if the problem goes away. This will help the devs narrow down when the problem first arose in the commit log.
 - **Try switching up dependency versions.** If the software in question has dependencies (other libraries, etc) try upgrading/downgrading those as well.
- **Search the project's bug/issue tracker** to make sure it's not a known issue.
- If you don't find a pre-existing issue, consider **checking with the mailing list and/or IRC channel** in case the problem is non-bug-related.

What to put in your bug report

Make sure your report gets the attention it deserves: bug reports with missing information may be ignored or punted back to you, delaying a fix. The below constitutes a bare minimum; more info is almost always better:

- **What version of the core programming language interpreter/compiler are you using?** For example, if it's a Python project, are you using Python 2.7.3? Python 3.3.1? PyPy 2.0?
- **What operating system are you on?** Windows? (Vista? 7? 32-bit? 64-bit?) Mac OS X? (10.7.4? 10.9.0?) Linux? (Which distro? Which version of that distro? 32 or 64 bits?) Again, more detail is better.
- **Which version or versions of the software are you using?** Ideally, you followed the advice above and have ruled out (or verified that the problem exists in) a few different versions.
- **How can the developers recreate the bug on their end?** If possible, include a copy of your code, the command you used to invoke it, and the full output of your run (if applicable.)
 - A common tactic is to pare down your code until a simple (but still bug-causing) “base case” remains. Not only can this help you identify problems which aren't real bugs, but it means the developer can get to fixing the bug faster.

1.3 website source code

1.3.1 extender application

Submodules

extender.forms

```
class extender.forms.RoleCreateForm(owner=None, *args, **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of extender.models.RoleObject

        fields = ['shortname', 'desc', 'exid']

        labels = {'desc': 'description', 'exid': 'External ID', 'shortname': 'Short Name'}

        help_texts = {'exid': 'Key to link yaml vars', 'shortname': 'Directory name for t'}

        error_messages = {'exid': {'max_length': 'ID is too long.'}, 'shortname': {'max_

        base_fields = {'desc': <django.forms.fields.CharField object>, 'exid': <django.forms

        declared_fields = {}

        media

class extender.forms.LogicalGroupCreateForm(owner=None, *args, **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of extender.models.LogicalGroupObject
```

```
fields = ['shortname', 'exid']
labels = {'exid': 'External ID', 'shortname': 'Short Name'}
help_texts = {'exid': 'Key to link yaml vars', 'shortname': 'File name for that C
error_messages = {'exid': {'max_length': 'ID is too long.'}, 'shortname': {'max_
base_fields = {'exid': <django.forms.fields.CharField object>, 'shortname': <django.
declared_fields = {}
media

class extender.forms.ConfigSectionCreateForm(owner=None, *args, **kwargs)
Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    model
        alias of extender.models.ConfigSectionObject

    fields = ['shortname', 'exid', 'desc', 'listobject']
    labels = {'desc': 'Description', 'exid': 'External ID', 'listobject': 'Is this s
    help_texts = {'desc': 'Description', 'exid': 'Key to link yaml vars', 'listobject
    error_messages = {'exid': {'max_length': 'ID is too long.'}, 'shortname': {'max_
base_fields = {'desc': <django.forms.fields.CharField object>, 'exid': <django.forms
declared_fields = {}
media

class extender.forms.RoleTaskCreateForm(owner=None, *args, **kwargs)
Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    model
        alias of extender.models.RoleTaskObject

    fields = ['shortname', 'role', 'desc']
    labels = {'desc': 'Description', 'role': 'Assigned to role', 'shortname': 'Task
    help_texts = {'desc': 'Task description', 'role': 'Choose a role that this task f
    error_messages = {'exid': {'max_length': 'ID is too long.'}, 'shortname': {'max_
base_fields = {'desc': <django.forms.fields.CharField object>, 'role': <django.forms
declared_fields = {}
media

class extender.forms.ApiCreateForm(owner=None, *args, **kwargs)
Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    model
        alias of extender.models.ApiObject
```

```

        fields = ['shortname', 'desc']
        labels = {'desc': 'Description', 'shortname': 'Short Name'}
        help_texts = {'desc': 'Description for that API', 'shortname': 'API name '}
        error_messages = {'desc': {'max_length': 'ID is too long.'}, 'shortname': {'max_
base_fields = {'desc': <django.forms.fields.CharField object>, 'shortname': <django.
declared_fields = {}
media

class extender.forms.ApiSectionCreateForm(owner=None, *args, **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of extender.models.ApiSectionObject

        fields = ['shortname', 'exid', 'api', 'desc', 'docurl']
        labels = {'api': 'API', 'desc': 'Description', 'docurl': 'API Documentation link
        help_texts = {'api': 'Parent API', 'desc': 'Section description', 'docurl': 'Pro
        error_messages = {'desc': {'max_length': 'ID is too long.'}, 'shortname': {'max_
base_fields = {'api': <django.forms.models.ModelChoiceField object>, 'desc': <django
declared_fields = {}
media

class extender.forms.ApiSubCreateForm(owner=None, *args, **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of extender.models.ApiSubObject

        fields = ['shortname', 'exid', 'section', 'desc', 'trailngspace']
        labels = {'desc': 'Description', 'exid': 'External ID', 'section': 'API Section
        help_texts = {'desc': 'Sub Object description', 'exid': 'Key to link yaml vars',
        error_messages = {'desc': {'max_length': 'ID is too long.'}, 'name': {'max_lengt
base_fields = {'desc': <django.forms.fields.CharField object>, 'exid': <django.forms
declared_fields = {}
media

class extender.forms.RoleTemplateCreateForm(owner=None, *args, **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of extender.models.RoleTemplateObject

```

```
    fields = ['shortname', 'role', 'desc', 'istemplate']
    labels = {'desc': 'Description', 'istemplate': 'Parse', 'role': 'Role', 'shortname': 'Shortname'}
    help_texts = {'desc': 'Template description', 'istemplate': 'Process Jinja2 or jinja2'}
    error_messages = {'desc': {'max_length': 'ID is too long.'}, 'shortname': {'max_length': 'Shortname is too long.'}}
    base_fields = {'desc': <django.forms.fields.CharField object>, 'istemplate': <django.forms.fields.BooleanField object>}
    declared_fields = {}
    media

class extender.forms.ConfigSubCreateForm(owner=None, *args, **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of extender.models.ConfigSubObject

        fields = ['shortname', 'exid', 'section', 'desc']
        labels = {'desc': 'Description', 'exid': 'External ID', 'section': 'Config Section', 'shortname': 'Shortname'}
        help_texts = {'desc': 'Config Object description', 'exid': 'Key to link yaml vars', 'section': 'Config Section'}
        error_messages = {'desc': {'max_length': 'ID is too long.'}, 'name': {'max_length': 'Name is too long.'}}
        base_fields = {'desc': <django.forms.fields.CharField object>, 'exid': <django.forms.fields.CharField object>, 'section': <django.forms.fields.CharField object>}
        declared_fields = {}
        media

class extender.forms.ConfigValueCreateForm(owner=None, *args, **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of extender.models.ConfigValueObject

        fields = ['exid', 'subobject', 'value', 'desc']
        labels = {'desc': 'Description', 'exid': 'External ID', 'subobject': 'Sub object', 'value': 'Value'}
        help_texts = {'desc': 'short Description', 'exid': 'Key to link yaml vars', 'subobject': 'Sub object', 'value': 'Value'}
        error_messages = {'desc': {'max_length': 'ID is too long.'}, 'name': {'max_length': 'Name is too long.'}}
        base_fields = {'desc': <django.forms.fields.CharField object>, 'exid': <django.forms.fields.CharField object>, 'subobject': <django.forms.fields.CharField object>}
        declared_fields = {}
        media

class extender.forms.ApiValueCreateForm(owner=None, *args, **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of extender.models.ApiValueObject
```

```

    fields = ['exid', 'subobject', 'value', 'desc']
    labels = {'desc': 'Description', 'exid': 'External ID', 'subobject': 'Sub object'}
    help_texts = {'desc': 'short Description', 'exid': 'Key to link yaml vars', 'subobject': 'Sub object'}
    error_messages = {'desc': {'max_length': 'ID is too long.'}, 'name': {'max_length': 'Name is too long.'}}
    base_fields = {'desc': <django.forms.fields.CharField object>, 'exid': <django.forms.fields.CharField object>}
    declared_fields = {}
    media

```

extender.models

```

class extender.models.RoleObjectQuerySet (model=None, query=None, using=None, hints=None)
    Bases: django.db.models.query.QuerySet
    search (query)

class extender.models.RoleObjectManager
    Bases: django.db.models.manager.Manager
    get_queryset ()
        Return a new QuerySet object. Subclasses can override this method to customize the behavior of the Manager.
    search (query)

```

```

class extender.models.RoleObject (id, owner, shortname, exid, desc, addon, slug, ispublic, popularity)
    Bases: django.db.models.base.Model

```

owner

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.

shortname

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

exid

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

desc

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

addon

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

slug

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

ispublic

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

popularity

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

objects = <extender.models.RoleObjectManager object>

get_absolute_url()

get_absolute_url_edit()

title

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

id

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

owner_id

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

roletaskobject_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.

roletemplateobject_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.

class extender.models.LogicalGroupObjectQuerySet (*model=None, query=None, using=None, hints=None*)

Bases: django.db.models.query.QuerySet

search (*query*)

class `extender.models.LogicalGroupObjectManager`

Bases: `django.db.models.manager.Manager`

get_queryset ()

Return a new QuerySet object. Subclasses can override this method to customize the behavior of the Manager.

search (*query*)

class `extender.models.LogicalGroupObject` (*id, owner, shortname, exid, slug, ispublic, popularity*)

Bases: `django.db.models.base.Model`

owner

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.

shortname

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

exid

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

slug

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

ispublic

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

popularity

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

objects = `<extender.models.LogicalGroupObjectManager object>`

get_absolute_url ()

get_absolute_url_edit ()

title

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

id

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

owner_id

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

```
class extender.models.ConfigSectionObjectQuerySet (model=None, query=None, using=None, hints=None)
```

Bases: django.db.models.query.QuerySet

search (*query*)

```
class extender.models.ConfigSectionObjectManager
```

Bases: django.db.models.manager.Manager

get_queryset ()

Return a new QuerySet object. Subclasses can override this method to customize the behavior of the Manager.

search (*query*)

```
class extender.models.ConfigSectionObject (id, owner, shortname, exid, desc, listobject, slug, ispublic, popularity)
```

Bases: django.db.models.base.Model

owner

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.

shortname

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

exid

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

desc

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

listobject

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

slug

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

ispublic

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

popularity

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

```
objects = <extender.models.ConfigSectionObjectManager object>
```

`get_absolute_url()`

`get_absolute_url_edit()`

`title`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

configsubobject_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.

id

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

owner_id

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

class `extender.models.RoleTaskObjectQuerySet` (*model=None, query=None, using=None, hints=None*)

Bases: `django.db.models.query.QuerySet`

search (*query*)

class `extender.models.RoleTaskObjectManager`

Bases: `django.db.models.manager.Manager`

get_queryset ()

Return a new `QuerySet` object. Subclasses can override this method to customize the behavior of the `Manager`.

search (*query*)

class `extender.models.RoleTaskObject` (*id, owner, shortname, desc, role, slug, ispublic, popularity, isdev, ispublished*)

Bases: `django.db.models.base.Model`

owner

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.

shortname

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

desc

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

role

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.

slug

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

ispublic

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

popularity

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

isdev

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

ispublished

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

objects = <extender.models.RoleTaskObjectManager object>

get_absolute_url()

get_absolute_url_edit()

title

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

id

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

owner_id

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

role_id

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

```
class extender.models.ApiObjectQuerySet (model=None, query=None, using=None,  
                                           hints=None)
```

Bases: django.db.models.query.QuerySet

search (*query*)

```
class extender.models.ApiObjectManager
```

Bases: django.db.models.manager.Manager

get_queryset ()

Return a new QuerySet object. Subclasses can override this method to customize the behavior of the Manager.

search (*query*)

```
class extender.models.ApiObject (id, owner, shortname, desc, slug, ispublic, popularity, isdev,  
                                   ispublished)
```

Bases: django.db.models.base.Model

owner

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.

shortname

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

desc

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

slug

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

ispublic

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

popularity

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

isdev

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

ispublished

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

objects = <extender.models.ApiObjectManager object>

get_absolute_url()

get_absolute_url_edit()

title

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

apisectionobject_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.

id

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

owner_id

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

```
class extender.models.ApiSectionObjectQuerySet (model=None, query=None, us-
                                              ing=None, hints=None)
```

Bases: `django.db.models.query.QuerySet`

search(query)

class extender.models.ApiSectionObjectManager

Bases: `django.db.models.manager.Manager`

get_queryset()

Return a new `QuerySet` object. Subclasses can override this method to customize the behavior of the `Manager`.

search(query)

```
class extender.models.ApiSectionObject (id, owner, shortname, exid, desc, api, slug, ispublic,
                                         popularity, docurl)
```

Bases: `django.db.models.base.Model`

owner

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.

shortname

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

exid

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

desc

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

api

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.

slug

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

ispublic

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

popularity

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

docurl

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

objects = <extender.models.ApiSectionObjectManager object>

get_absolute_url()

get_absolute_url_edit()

title

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

api_id

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

apisubobject_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.

id

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

owner_id

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

```
class extender.models.RoleTemplateObjectQuerySet(model=None, query=None, using=None, hints=None)
```

Bases: `django.db.models.query.QuerySet`

search (*query*)

```
class extender.models.RoleTemplateObjectManager
```

Bases: `django.db.models.manager.Manager`

get_queryset ()

Return a new `QuerySet` object. Subclasses can override this method to customize the behavior of the `Manager`.

search (*query*)

```
class extender.models.RoleTemplateObject(id, owner, shortname, slug, desc, istemplate, role, ispublic, popularity, isdev, ispublished)
```

Bases: `django.db.models.base.Model`

owner

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.

shortname

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

slug

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

desc

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

istemplate

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

role

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.

ispublic

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

popularity

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

isdev

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

ispublished

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

objects = <extender.models.RoleTemplateObjectManager object>

get_absolute_url()

get_absolute_url_edit()

title

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

id

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

owner_id

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

role_id

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

class extender.models.ConfigSubObjectQuerySet (*model=None, query=None, using=None, hints=None*)

Bases: django.db.models.query.QuerySet

search (*query*)

class extender.models.ConfigSubObjectManager

Bases: django.db.models.manager.Manager

get_queryset()

Return a new QuerySet object. Subclasses can override this method to customize the behavior of the Manager.

search(query)

class `extender.models.ConfigSubObject` (*id, owner, shortname, exid, desc, section, slug, ispublic, popularity*)

Bases: `django.db.models.base.Model`

owner

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.

shortname

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

exid

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

desc

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

section

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.

slug

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

ispublic

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

popularity

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

objects = `<extender.models.ConfigSubObjectManager object>`

get_absolute_url()

get_absolute_url_edit()

title

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

configvalueobject_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.

id

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

owner_id

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

section_id

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

class extender.models.**ApiSubObjectQuerySet** (*model=None, query=None, using=None, hints=None*)

Bases: django.db.models.query.QuerySet

search (*query*)

class extender.models.**ApiSubObjectManager**

Bases: django.db.models.manager.Manager

get_queryset ()

Return a new QuerySet object. Subclasses can override this method to customize the behavior of the Manager.

search (*query*)

class extender.models.**ApiSubObject** (*id, owner, shortname, exid, desc, section, slug, ispublic, popularity, trailngspace*)

Bases: django.db.models.base.Model

owner

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.

shortname

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

exid

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

desc

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

section

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.

slug

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

ispublic

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

popularity

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

trailingspace

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

objects = <extender.models.ApiSubObjectManager object>

get_absolute_url()

get_absolute_url_edit()

title

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

apivalueobject_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.

id

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

owner_id

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

section_id

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

```
class extender.models.ConfigValueObjectQuerySet(model=None, query=None, using=None, hints=None)
```

Bases: `django.db.models.query.QuerySet`

search (*query*)

```
class extender.models.ConfigValueObjectManager
```

Bases: `django.db.models.manager.Manager`

get_queryset ()

Return a new `QuerySet` object. Subclasses can override this method to customize the behavior of the Manager.

search (*query*)

```
class extender.models.ConfigValueObject(id, owner, exid, subobject, value, desc, slug, ispublic, popularity)
```

Bases: `django.db.models.base.Model`

owner

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.

exid

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

subobject

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.

value

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

desc

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

slug

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

ispublic

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

popularity

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

objects = <extender.models.ConfigValueObjectManager object>

get_absolute_url()

get_absolute_url_edit()

title**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

id

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

owner_id

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

subobject_id

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

class extender.models.**ApiValueObjectQuerySet** (*model=None, query=None, using=None, hints=None*)

Bases: django.db.models.query.QuerySet

search (*query*)

class extender.models.**ApiValueObjectManager**

Bases: django.db.models.manager.Manager

get_queryset()

Return a new QuerySet object. Subclasses can override this method to customize the behavior of the Manager.

search (*query*)

class extender.models.**ApiValueObject** (*id, owner, exid, subobject, value, desc, slug, ispublic, popularity*)

Bases: django.db.models.base.Model

owner

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.

exid

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

subobject

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.

value

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

desc

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

slug

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

ispublic

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

popularity

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

objects = <extender.models.ApiValueObjectManager object>

get_absolute_url()

get_absolute_url_edit()

title

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

id

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

owner_id

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

subobject_id

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

```
extender.models.ro_pre_save_receiver(sender, instance, *args, **kwargs)
extender.models.lgo_pre_save_receiver(sender, instance, *args, **kwargs)
extender.models.cs_pre_save_receiver(sender, instance, *args, **kwargs)
extender.models.rt_pre_save_receiver(sender, instance, *args, **kwargs)
extender.models.ao_pre_save_receiver(sender, instance, *args, **kwargs)
extender.models.as_pre_save_receiver(sender, instance, *args, **kwargs)
extender.models.rtm_pre_save_receiver(sender, instance, *args, **kwargs)
extender.models.co_pre_save_receiver(sender, instance, *args, **kwargs)
extender.models.aps_pre_save_receiver(sender, instance, *args, **kwargs)
extender.models.cv_pre_save_receiver(sender, instance, *args, **kwargs)
extender.models.av_pre_save_receiver(sender, instance, *args, **kwargs)
```

extender.utils

```
extender.utils.random_string_generator(size=10, chars='abcdefghijklmnopqrstuvwxyz0123456789')
extender.utils.unique_slug_generator(instance, new_slug=None)
    This is for a Django project and it assumes your instance has a slug field and a title character (char)
    field.
```

extender.views

1.3.2 runnow application

Submodules**runnow.forms****runnow.views**

1.4 Contributing to Open Source Projects

1.4.1 About

This document provides a set of best practices for open source contributions - bug reports, code submissions / pull requests, etc.

For the most part, these guidelines apply equally to *any* project regardless of programming language or topic. Where applicable, we outline where individual projects/languages may have additional requirements.

Naturally, this document is itself open source, and we encourage feedback & suggestions for improvement.

Sources

Currently this document draws from the contribution documentation for a handful of related Python open source projects: [Fabric](#), [Invoke](#), [Paramiko](#), etc.

It's expected that over time we will incorporate additional material from related attempts at consolidating this type of info. We'll update with a list here when that happens.

1.4.2 Submitting bugs

Due diligence

Before submitting a bug, please do the following:

- Perform **basic troubleshooting** steps:
 - **Make sure you're on the latest version.** If you're not on the most recent version, your problem may have been solved already! Upgrading is always the best first step.
 - **Try older versions.** If you're already *on* the latest release, try rolling back a few minor versions (e.g. if on 1.7, try 1.5 or 1.6) and see if the problem goes away. This will help the devs narrow down when the problem first arose in the commit log.
 - **Try switching up dependency versions.** If the software in question has dependencies (other libraries, etc) try upgrading/downgrading those as well.
- **Search the project's bug/issue tracker** to make sure it's not a known issue.
- If you don't find a pre-existing issue, consider **checking with the mailing list and/or IRC channel** in case the problem is non-bug-related.

What to put in your bug report

Make sure your report gets the attention it deserves: bug reports with missing information may be ignored or punted back to you, delaying a fix. The below constitutes a bare minimum; more info is almost always better:

- **What version of the core programming language interpreter/compiler are you using?** For example, if it's a Python project, are you using Python 2.7.3? Python 3.3.1? PyPy 2.0?
- **What operating system are you on?** Windows? (Vista? 7? 32-bit? 64-bit?) Mac OS X? (10.7.4? 10.9.0?) Linux? (Which distro? Which version of that distro? 32 or 64 bits?) Again, more detail is better.
- **Which version or versions of the software are you using?** Ideally, you followed the advice above and have ruled out (or verified that the problem exists in) a few different versions.
- **How can the developers recreate the bug on their end?** If possible, include a copy of your code, the command you used to invoke it, and the full output of your run (if applicable.)
 - A common tactic is to pare down your code until a simple (but still bug-causing) “base case” remains. Not only can this help you identify problems which aren't real bugs, but it means the developer can get to fixing the bug faster.

1.4.3 Contributing changes

Version control branching

- Always **make a new branch** for your work, no matter how small. This makes it easy for others to take just that one set of changes from your repository, in case you have multiple unrelated changes floating around.
 - A corollary: **don't submit unrelated changes in the same branch/pull request!** The maintainer shouldn't have to reject your awesome bugfix because the feature you put in with it needs more review.
- **Base your new branch off of the appropriate branch** on the main repository:
 - **Bug fixes** should be based on the branch named after the **oldest supported release line** the bug affects.
 - * E.g. if a feature was introduced in 1.1, the latest release line is 1.3, and a bug is found in that feature - make your branch based on 1.1. The maintainer will then forward-port it to 1.3 and master.
 - * Bug fixes requiring large changes to the code or which have a chance of being otherwise disruptive, may need to base off of **master** instead. This is a judgement call – ask the devs!
 - **New features** should branch off of the **'master' branch**.
 - * Note that depending on how long it takes for the dev team to merge your patch, the copy of `master` you worked off of may get out of date! If you find yourself 'bumping' a pull request that's been side-lined for a while, **make sure you rebase or merge to latest master** to ensure a speedier resolution.

Code formatting

- **Follow the style you see used in the primary repository!** Consistency with the rest of the project always trumps other considerations. It doesn't matter if you have your own style or if the rest of the code breaks with the greater community - just follow along.
- Python projects usually follow the [PEP-8](#) guidelines (though many have minor deviations depending on the lead maintainers' preferences.)

Documentation isn't optional

It's not! Patches without documentation will be returned to sender. By "documentation" we mean:

- **Docstrings** (for Python; or API-doc-friendly comments for other languages) must be created or updated for public API functions/methods/etc. (This step is optional for some bugfixes.)
 - Don't forget to include `versionadded`/`versionchanged` ReST directives at the bottom of any new or changed Python docstrings!
 - * Use `versionadded` for truly new API members – new methods, functions, classes or modules.
 - * Use `versionchanged` when adding/removing new function/method arguments, or whenever behavior changes.
- New features should ideally include updates to **prose documentation**, including useful example code snippets.
- All submissions should have a **changelog entry** crediting the contributor and/or any individuals instrumental in identifying the problem.

Tests aren't optional

Any bugfix that doesn't include a test proving the existence of the bug being fixed, may be suspect. Ditto for new features that can't prove they actually work.

We’ve found that test-first development really helps make features better architected and identifies potential edge cases earlier instead of later. Writing tests before the implementation is strongly encouraged.

Full example

Here’s an example workflow for a project `theproject` hosted on Github, which is currently in version 1.3.x. Your username is `yourname` and you’re submitting a basic bugfix. (This workflow only changes slightly if the project is hosted at Bitbucket, self-hosted, or etc.)

Preparing your Fork

1. Hit ‘fork’ on Github, creating e.g. `yourname/theproject`.
2. Clone your project: `git clone git@github.com:yourname/theproject`.
3. Create a branch: `cd theproject; git checkout -b foo-the-bars 1.3`.

Making your Changes

1. Add changelog entry crediting yourself.
2. Write tests expecting the correct/fixed functionality; make sure they fail.
3. Hack, hack, hack.
4. Run tests again, making sure they pass.
5. Commit your changes: `git commit -m "Foo the bars"`

Creating Pull Requests

1. Push your commit to get it back up to your fork: `git push origin HEAD`
2. Visit Github, click handy “Pull request” button that it will make upon noticing your new branch.
3. In the description field, write down issue number (if submitting code fixing an existing issue) or describe the issue + your fix (if submitting a wholly new bugfix).
4. Hit ‘submit’! And please be patient - the maintainers will get to you when they can.

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

CHAPTER 3

Credits

Logo made with Logomakr.com

e

`extender.forms`, [3](#)
`extender.models`, [7](#)
`extender.utils`, [24](#)

A

addon (*extender.models.RoleObject* attribute), 7
 ao_pre_save_receiver() (in module *extender.models*), 24
 api (*extender.models.ApiSectionObject* attribute), 15
 api_id (*extender.models.ApiSectionObject* attribute), 15
 ApiCreateForm (class in *extender.forms*), 4
 ApiCreateForm.Meta (class in *extender.forms*), 4
 ApiObject (class in *extender.models*), 13
 ApiObject.DoesNotExist, 14
 ApiObject.MultipleObjectsReturned, 14
 ApiObjectManager (class in *extender.models*), 13
 ApiObjectQuerySet (class in *extender.models*), 13
 ApiSectionCreateForm (class in *extender.forms*), 5
 ApiSectionCreateForm.Meta (class in *extender.forms*), 5
 ApiSectionObject (class in *extender.models*), 14
 ApiSectionObject.DoesNotExist, 15
 ApiSectionObject.MultipleObjectsReturned, 15
 apisessionobject_set (in *extender.models.ApiObject* attribute), 14
 ApiSectionObjectManager (class in *extender.models*), 14
 ApiSectionObjectQuerySet (class in *extender.models*), 14
 ApiSubCreateForm (class in *extender.forms*), 5
 ApiSubCreateForm.Meta (class in *extender.forms*), 5
 ApiSubObject (class in *extender.models*), 19
 ApiSubObject.DoesNotExist, 20
 ApiSubObject.MultipleObjectsReturned, 20
 apisubobject_set (in *extender.models.ApiSectionObject* attribute), 15
 ApiSubObjectManager (class in *extender.models*), 19
 ApiSubObjectQuerySet (class in *extender.models*), 19
 ApiValueCreateForm (class in *extender.forms*), 6
 ApiValueCreateForm.Meta (class in *extender.forms*), 6
 ApiValueObject (class in *extender.models*), 22
 ApiValueObject.DoesNotExist, 23
 ApiValueObject.MultipleObjectsReturned, 23
 apivalueobject_set (in *extender.models.ApiSubObject* attribute), 20
 ApiValueObjectManager (class in *extender.models*), 22
 ApiValueObjectQuerySet (class in *extender.models*), 22
 aps_pre_save_receiver() (in module *extender.models*), 24
 as_pre_save_receiver() (in module *extender.models*), 24
 av_pre_save_receiver() (in module *extender.models*), 24

B

base_fields (*extender.forms.ApiCreateForm* attribute), 5
 base_fields (*extender.forms.ApiSectionCreateForm* attribute), 5
 base_fields (*extender.forms.ApiSubCreateForm* attribute), 5
 base_fields (*extender.forms.ApiValueCreateForm* attribute), 7
 base_fields (in *extender.forms.ConfigSectionCreateForm* attribute), 4
 base_fields (*extender.forms.ConfigSubCreateForm* attribute), 6
 base_fields (*extender.forms.ConfigValueCreateForm* attribute), 6
 base_fields (in *extender.forms.LogicalGroupCreateForm* attribute), 4

`base_fields` (*extender.forms.RoleCreateForm* attribute), 3

`base_fields` (*extender.forms.RoleTaskCreateForm* attribute), 4

`base_fields` (*extender.forms.RoleTemplateCreateForm* attribute), 6

C

`co_pre_save_receiver()` (in module *extender.models*), 24

`ConfigSectionCreateForm` (class in *extender.forms*), 4

`ConfigSectionCreateForm.Meta` (class in *extender.forms*), 4

`ConfigSectionObject` (class in *extender.models*), 10

`ConfigSectionObject.DoesNotExist`, 11

`ConfigSectionObject.MultipleObjectsReturned`, 11

`ConfigSectionObjectManager` (class in *extender.models*), 10

`ConfigSectionObjectQuerySet` (class in *extender.models*), 10

`ConfigSubCreateForm` (class in *extender.forms*), 6

`ConfigSubCreateForm.Meta` (class in *extender.forms*), 6

`ConfigSubObject` (class in *extender.models*), 18

`ConfigSubObject.DoesNotExist`, 18

`ConfigSubObject.MultipleObjectsReturned`, 19

`configsubobject_set` (*extender.models.ConfigSectionObject* attribute), 11

`ConfigSubObjectManager` (class in *extender.models*), 17

`ConfigSubObjectQuerySet` (class in *extender.models*), 17

`ConfigValueCreateForm` (class in *extender.forms*), 6

`ConfigValueCreateForm.Meta` (class in *extender.forms*), 6

`ConfigValueObject` (class in *extender.models*), 21

`ConfigValueObject.DoesNotExist`, 22

`ConfigValueObject.MultipleObjectsReturned`, 22

`configvalueobject_set` (*extender.models.ConfigSubObject* attribute), 19

`ConfigValueObjectManager` (class in *extender.models*), 21

`ConfigValueObjectQuerySet` (class in *extender.models*), 21

`cs_pre_save_receiver()` (in module *extender.models*), 24

`cv_pre_save_receiver()` (in module *extender.models*), 24

D

`declared_fields` (*extender.forms.ApiCreateForm* attribute), 5

`declared_fields` (*extender.forms.ApiSectionCreateForm* attribute), 5

`declared_fields` (*extender.forms.ApiSubCreateForm* attribute), 5

`declared_fields` (*extender.forms.ApiValueCreateForm* attribute), 7

`declared_fields` (*extender.forms.ConfigSectionCreateForm* attribute), 4

`declared_fields` (*extender.forms.ConfigSubCreateForm* attribute), 6

`declared_fields` (*extender.forms.ConfigValueCreateForm* attribute), 6

`declared_fields` (*extender.forms.LogicalGroupCreateForm* attribute), 4

`declared_fields` (*extender.forms.RoleCreateForm* attribute), 3

`declared_fields` (*extender.forms.RoleTaskCreateForm* attribute), 4

`declared_fields` (*extender.forms.RoleTemplateCreateForm* attribute), 6

`desc` (*extender.models.ApiObject* attribute), 13

`desc` (*extender.models.ApiSectionObject* attribute), 15

`desc` (*extender.models.ApiSubObject* attribute), 20

`desc` (*extender.models.ApiValueObject* attribute), 23

`desc` (*extender.models.ConfigSectionObject* attribute), 10

`desc` (*extender.models.ConfigSubObject* attribute), 18

`desc` (*extender.models.ConfigValueObject* attribute), 22

`desc` (*extender.models.RoleObject* attribute), 7

`desc` (*extender.models.RoleTaskObject* attribute), 12

`desc` (*extender.models.RoleTemplateObject* attribute), 16

`docurl` (*extender.models.ApiSectionObject* attribute), 15

E

`error_messages` (*extender.forms.ApiCreateForm.Meta* attribute), 5

[error_messages \(extender:forms.ApiSectionCreateForm.Meta attribute\), 5](#)
[error_messages \(extender:forms.ApiSubCreateForm.Meta attribute\), 5](#)
[error_messages \(extender:forms.ApiValueCreateForm.Meta attribute\), 7](#)
[error_messages \(extender:forms.ConfigSectionCreateForm.Meta attribute\), 4](#)
[error_messages \(extender:forms.ConfigSubCreateForm.Meta attribute\), 6](#)
[error_messages \(extender:forms.ConfigValueCreateForm.Meta attribute\), 6](#)
[error_messages \(extender:forms.LogicalGroupCreateForm.Meta attribute\), 4](#)
[error_messages \(extender:forms.RoleCreateForm.Meta attribute\), 3](#)
[error_messages \(extender:forms.RoleTaskCreateForm.Meta attribute\), 4](#)
[error_messages \(extender:forms.RoleTemplateCreateForm.Meta attribute\), 6](#)
[exid \(extender.models.ApiSectionObject attribute\), 15](#)
[exid \(extender.models.ApiSubObject attribute\), 20](#)
[exid \(extender.models.ApiValueObject attribute\), 23](#)
[exid \(extender.models.ConfigSectionObject attribute\), 10](#)
[exid \(extender.models.ConfigSubObject attribute\), 18](#)
[exid \(extender.models.ConfigValueObject attribute\), 21](#)
[exid \(extender.models.LogicalGroupObject attribute\), 9](#)
[exid \(extender.models.RoleObject attribute\), 7](#)
[extender.forms \(module\), 3](#)
[extender.models \(module\), 7](#)
[extender.utils \(module\), 24](#)

F

[fields \(extender:forms.ApiCreateForm.Meta attribute\), 4](#)
[fields \(extender:forms.ApiSectionCreateForm.Meta attribute\), 5](#)
[fields \(extender:forms.ApiSubCreateForm.Meta attribute\), 5](#)
[fields \(extender:forms.ApiValueCreateForm.Meta attribute\), 6](#)
[fields \(extender:forms.ConfigSectionCreateForm.Meta attribute\), 4](#)
[fields \(extender:forms.ConfigSubCreateForm.Meta attribute\), 6](#)
[fields \(extender:forms.ConfigValueCreateForm.Meta attribute\), 6](#)
[fields \(extender:forms.LogicalGroupCreateForm.Meta attribute\), 3](#)
[fields \(extender:forms.RoleCreateForm.Meta attribute\), 3](#)
[fields \(extender:forms.RoleTaskCreateForm.Meta attribute\), 4](#)
[fields \(extender:forms.RoleTemplateCreateForm.Meta attribute\), 5](#)

G

[get_absolute_url \(\) \(extender.models.ApiObject method\), 13](#)
[get_absolute_url \(\) \(extender.models.ApiSectionObject method\), 15](#)
[get_absolute_url \(\) \(extender.models.ApiSubObject method\), 20](#)
[get_absolute_url \(\) \(extender.models.ApiValueObject method\), 23](#)
[get_absolute_url \(\) \(extender.models.ConfigSectionObject method\), 10](#)
[get_absolute_url \(\) \(extender.models.ConfigSubObject method\), 18](#)
[get_absolute_url \(\) \(extender.models.ConfigValueObject method\), 22](#)
[get_absolute_url \(\) \(extender.models.LogicalGroupObject method\), 9](#)
[get_absolute_url \(\) \(extender.models.RoleObject method\), 8](#)
[get_absolute_url \(\) \(extender.models.RoleTaskObject method\), 12](#)
[get_absolute_url \(\) \(extender.models.RoleTemplateObject method\), 17](#)
[get_absolute_url_edit \(\) \(extender.models.ApiObject method\), 14](#)
[get_absolute_url_edit \(\) \(extender.models.ApiSectionObject method\), 15](#)
[get_absolute_url_edit \(\) \(extender.models.ApiSubObject method\), 20](#)
[get_absolute_url_edit \(\) \(extender.models.ApiValueObject method\), 23](#)
[get_absolute_url_edit \(\) \(extender.models.ConfigSectionObject method\), 11](#)
[get_absolute_url_edit \(\) \(extender.models.ConfigSubObject method\), 18](#)

<code>get_absolute_url_edit()</code> <i>(extender.models.ConfigValueObject</i> 22	<i>method),</i>	<code>help_texts (extender.forms.ApiValueCreateForm.Meta</code> <i>attribute), 7</i>
<code>get_absolute_url_edit()</code> <i>(extender.models.LogicalGroupObject</i> 9	<i>method),</i>	<code>help_texts (extender.forms.ConfigSectionCreateForm.Meta</code> <i>attribute), 4</i>
<code>get_absolute_url_edit()</code> <i>(extender.models.RoleObject method), 8</i>		<code>help_texts (extender.forms.ConfigSubCreateForm.Meta</code> <i>attribute), 6</i>
<code>get_absolute_url_edit()</code> <i>(extender.models.RoleTaskObject method), 12</i>		<code>help_texts (extender.forms.ConfigValueCreateForm.Meta</code> <i>attribute), 6</i>
<code>get_absolute_url_edit()</code> <i>(extender.models.RoleTemplateObject</i> 17	<i>method),</i>	<code>help_texts (extender.forms.LogicalGroupCreateForm.Meta</code> <i>attribute), 4</i>
<code>get_queryset()</code> <i>(extender.models.RoleObjectManager</i> 13	<i>method),</i>	<code>help_texts (extender.forms.RoleCreateForm.Meta</code> <i>attribute), 3</i>
<code>get_queryset()</code> <i>(extender.models.RoleTaskCreateForm.Meta</i> 14		<code>help_texts (extender.forms.RoleTaskCreateForm.Meta</code> <i>attribute), 4</i>
<code>get_queryset()</code> <i>(extender.models.ApiObjectManager</i> 19	<i>method),</i>	<code>help_texts (extender.forms.RoleTemplateCreateForm.Meta</code> <i>attribute), 6</i>
<code>get_queryset()</code> <i>(extender.models.ApiSectionObjectManager method),</i> 19		
<code>get_queryset()</code> <i>(extender.models.ApiSubObjectManager</i> 22	<i>method),</i>	
<code>get_queryset()</code> <i>(extender.models.ApiValueObjectManager</i> 22	<i>method),</i>	
<code>get_queryset()</code> <i>(extender.models.ConfigSectionObjectManager</i> <i>method), 10</i>		
<code>get_queryset()</code> <i>(extender.models.ConfigSubObjectManager</i> 17	<i>method),</i>	
<code>get_queryset()</code> <i>(extender.models.ConfigValueObjectManager</i> <i>method), 21</i>		
<code>get_queryset()</code> <i>(extender.models.LogicalGroupObjectManager</i> <i>method), 9</i>		
<code>get_queryset()</code> <i>(extender.models.RoleObjectManager</i> 7	<i>method),</i>	
<code>get_queryset()</code> <i>(extender.models.RoleTaskObjectManager</i> 11	<i>method),</i>	
<code>get_queryset()</code> <i>(extender.models.RoleTemplateObjectManager</i> <i>method), 16</i>		
H		
<code>help_texts (extender.forms.ApiCreateForm.Meta</code> <i>attribute), 5</i>		<code>id (extender.models.ApiObject attribute), 14</code>
<code>help_texts (extender.forms.ApiSectionCreateForm.Meta</code> <i>attribute), 5</i>		<code>id (extender.models.ApiSectionObject attribute), 16</code>
<code>help_texts (extender.forms.ApiSubCreateForm.Meta</code> <i>attribute), 5</i>		<code>id (extender.models.ApiSubObject attribute), 21</code>
		<code>id (extender.models.ApiValueObject attribute), 23</code>
		<code>id (extender.models.ConfigSectionObject attribute), 11</code>
		<code>id (extender.models.ConfigSubObject attribute), 19</code>
		<code>id (extender.models.ConfigValueObject attribute), 22</code>
		<code>id (extender.models.LogicalGroupObject attribute), 9</code>
		<code>id (extender.models.RoleObject attribute), 8</code>
		<code>id (extender.models.RoleTaskObject attribute), 12</code>
		<code>id (extender.models.RoleTemplateObject attribute), 17</code>
		<code>isdev (extender.models.ApiObject attribute), 13</code>
		<code>isdev (extender.models.RoleTaskObject attribute), 12</code>
		<code>isdev (extender.models.RoleTemplateObject attribute),</code> 17
		<code>ispublic (extender.models.ApiObject attribute), 13</code>
		<code>ispublic (extender.models.ApiSectionObject</code> <i>attribute), 15</i>
		<code>ispublic (extender.models.ApiSubObject attribute), 20</code>
		<code>ispublic (extender.models.ApiValueObject attribute),</code> 23
		<code>ispublic (extender.models.ConfigSectionObject</code> <i>attribute), 10</i>
		<code>ispublic (extender.models.ConfigSubObject</code> <i>attribute), 18</i>
		<code>ispublic (extender.models.ConfigValueObject</code> <i>attribute), 22</i>
		<code>ispublic (extender.models.LogicalGroupObject</code> <i>attribute), 9</i>
		<code>ispublic (extender.models.RoleObject attribute), 8</code>
		<code>ispublic (extender.models.RoleTaskObject attribute),</code> 12
		<code>ispublic (extender.models.RoleTemplateObject</code> <i>attribute), 17</i>
		<code>ispublished (extender.models.ApiObject attribute),</code> 13

ispublished (extender.models.RoleTaskObject attribute), 12	media (extender.forms.ConfigSectionCreateForm attribute), 4
ispublished (extender.models.RoleTemplateObject attribute), 17	media (extender.forms.ConfigSubCreateForm attribute), 6
istemplate (extender.models.RoleTemplateObject attribute), 16	media (extender.forms.ConfigValueCreateForm attribute), 6
L	media (extender.forms.LogicalGroupCreateForm attribute), 4
labels (extender.forms.ApiCreateForm.Meta attribute), 5	media (extender.forms.RoleCreateForm attribute), 3
labels (extender.forms.ApiSectionCreateForm.Meta attribute), 5	media (extender.forms.RoleTaskCreateForm attribute), 4
labels (extender.forms.ApiSubCreateForm.Meta attribute), 5	media (extender.forms.RoleTemplateCreateForm attribute), 6
labels (extender.forms.ApiValueCreateForm.Meta attribute), 7	model (extender.forms.ApiCreateForm.Meta attribute), 4
labels (extender.forms.ConfigSectionCreateForm.Meta attribute), 4	model (extender.forms.ApiSectionCreateForm.Meta attribute), 5
labels (extender.forms.ConfigSubCreateForm.Meta attribute), 6	model (extender.forms.ApiSubCreateForm.Meta attribute), 5
labels (extender.forms.ConfigValueCreateForm.Meta attribute), 6	model (extender.forms.ApiValueCreateForm.Meta attribute), 6
labels (extender.forms.LogicalGroupCreateForm.Meta attribute), 4	model (extender.forms.ConfigSectionCreateForm.Meta attribute), 4
labels (extender.forms.RoleCreateForm.Meta attribute), 3	model (extender.forms.ConfigSubCreateForm.Meta attribute), 6
labels (extender.forms.RoleTaskCreateForm.Meta attribute), 4	model (extender.forms.ConfigValueCreateForm.Meta attribute), 6
labels (extender.forms.RoleTemplateCreateForm.Meta attribute), 6	model (extender.forms.LogicalGroupCreateForm.Meta attribute), 3
lgo_pre_save_receiver() (in module extender.models), 24	model (extender.forms.RoleCreateForm.Meta attribute), 3
listobject (extender.models.ConfigSectionObject attribute), 10	model (extender.forms.RoleTaskCreateForm.Meta attribute), 4
LogicalGroupCreateForm (class in extender.forms), 3	model (extender.forms.RoleTemplateCreateForm.Meta attribute), 5
LogicalGroupCreateForm.Meta (class in extender.forms), 3	O
LogicalGroupObject (class in extender.models), 9	objects (extender.models.ApiObject attribute), 13
LogicalGroupObject.DoesNotExist, 9	objects (extender.models.ApiSectionObject attribute), 15
LogicalGroupObject.MultipleObjectsReturned, 9	objects (extender.models.ApiSubObject attribute), 20
LogicalGroupObjectManager (class in extender.models), 9	objects (extender.models.ApiValueObject attribute), 23
LogicalGroupObjectQuerySet (class in extender.models), 8	objects (extender.models.ConfigSectionObject attribute), 10
M	objects (extender.models.ConfigSubObject attribute), 18
media (extender.forms.ApiCreateForm attribute), 5	objects (extender.models.ConfigValueObject attribute), 22
media (extender.forms.ApiSectionCreateForm attribute), 5	objects (extender.models.LogicalGroupObject attribute), 9
media (extender.forms.ApiSubCreateForm attribute), 5	objects (extender.models.RoleObject attribute), 8
media (extender.forms.ApiValueCreateForm attribute), 7	objects (extender.models.RoleTaskObject attribute), 12

objects (*extender.models.RoleTemplateObject attribute*), 17

owner (*extender.models.ApiObject attribute*), 13

owner (*extender.models.ApiSectionObject attribute*), 14

owner (*extender.models.ApiSubObject attribute*), 19

owner (*extender.models.ApiValueObject attribute*), 22

owner (*extender.models.ConfigSectionObject attribute*), 10

owner (*extender.models.ConfigSubObject attribute*), 18

owner (*extender.models.ConfigValueObject attribute*), 21

owner (*extender.models.LogicalGroupObject attribute*), 9

owner (*extender.models.RoleObject attribute*), 7

owner (*extender.models.RoleTaskObject attribute*), 11

owner (*extender.models.RoleTemplateObject attribute*), 16

owner_id (*extender.models.ApiObject attribute*), 14

owner_id (*extender.models.ApiSectionObject attribute*), 16

owner_id (*extender.models.ApiSubObject attribute*), 21

owner_id (*extender.models.ApiValueObject attribute*), 23

owner_id (*extender.models.ConfigSectionObject attribute*), 11

owner_id (*extender.models.ConfigSubObject attribute*), 19

owner_id (*extender.models.ConfigValueObject attribute*), 22

owner_id (*extender.models.LogicalGroupObject attribute*), 9

owner_id (*extender.models.RoleObject attribute*), 8

owner_id (*extender.models.RoleTaskObject attribute*), 12

owner_id (*extender.models.RoleTemplateObject attribute*), 17

P

popularity (*extender.models.ApiObject attribute*), 13

popularity (*extender.models.ApiSectionObject attribute*), 15

popularity (*extender.models.ApiSubObject attribute*), 20

popularity (*extender.models.ApiValueObject attribute*), 23

popularity (*extender.models.ConfigSectionObject attribute*), 10

popularity (*extender.models.ConfigSubObject attribute*), 18

popularity (*extender.models.ConfigValueObject attribute*), 22

popularity (*extender.models.LogicalGroupObject attribute*), 9

popularity (*extender.models.RoleObject attribute*), 8

popularity (*extender.models.RoleTaskObject attribute*), 12

popularity (*extender.models.RoleTemplateObject attribute*), 17

R

random_string_generator() (in module *extender.utils*), 24

ro_pre_save_receiver() (in module *extender.models*), 24

role (*extender.models.RoleTaskObject attribute*), 12

role (*extender.models.RoleTemplateObject attribute*), 16

role_id (*extender.models.RoleTaskObject attribute*), 12

role_id (*extender.models.RoleTemplateObject attribute*), 17

RoleCreateForm (class in *extender.forms*), 3

RoleCreateForm.Meta (class in *extender.forms*), 3

RoleObject (class in *extender.models*), 7

RoleObject.DoesNotExist, 8

RoleObject.MultipleObjectsReturned, 8

RoleObjectManager (class in *extender.models*), 7

RoleObjectQuerySet (class in *extender.models*), 7

RoleTaskCreateForm (class in *extender.forms*), 4

RoleTaskCreateForm.Meta (class in *extender.forms*), 4

RoleTaskObject (class in *extender.models*), 11

RoleTaskObject.DoesNotExist, 12

RoleTaskObject.MultipleObjectsReturned, 12

roletaskobject_set (*extender.models.RoleObject attribute*), 8

RoleTaskObjectManager (class in *extender.models*), 11

RoleTaskObjectQuerySet (class in *extender.models*), 11

RoleTemplateCreateForm (class in *extender.forms*), 5

RoleTemplateCreateForm.Meta (class in *extender.forms*), 5

RoleTemplateObject (class in *extender.models*), 16

RoleTemplateObject.DoesNotExist, 17

RoleTemplateObject.MultipleObjectsReturned, 17

roletemplateobject_set (*extender.models.RoleObject attribute*), 8

RoleTemplateObjectManager (class in *extender.models*), 16

RoleTemplateObjectQuerySet (class in *extender.models*), 16

rt_pre_save_receiver() (in module *extender.models*), 24

`rtm_pre_save_receiver()` (in module `extender.models`), 24

S

`search()` (`extender.models.ApiObjectManager` method), 13

`search()` (`extender.models.ApiObjectQuerySet` method), 13

`search()` (`extender.models.ApiSectionObjectManager` method), 14

`search()` (`extender.models.ApiSectionObjectQuerySet` method), 14

`search()` (`extender.models.ApiSubObjectManager` method), 19

`search()` (`extender.models.ApiSubObjectQuerySet` method), 19

`search()` (`extender.models.ApiValueObjectManager` method), 22

`search()` (`extender.models.ApiValueObjectQuerySet` method), 22

`search()` (`extender.models.ConfigSectionObjectManager` method), 10

`search()` (`extender.models.ConfigSectionObjectQuerySet` method), 10

`search()` (`extender.models.ConfigSubObjectManager` method), 18

`search()` (`extender.models.ConfigSubObjectQuerySet` method), 17

`search()` (`extender.models.ConfigValueObjectManager` method), 21

`search()` (`extender.models.ConfigValueObjectQuerySet` method), 21

`search()` (`extender.models.LogicalGroupObjectManager` method), 9

`search()` (`extender.models.LogicalGroupObjectQuerySet` method), 8

`search()` (`extender.models.RoleObjectManager` method), 7

`search()` (`extender.models.RoleObjectQuerySet` method), 7

`search()` (`extender.models.RoleTaskObjectManager` method), 11

`search()` (`extender.models.RoleTaskObjectQuerySet` method), 11

`search()` (`extender.models.RoleTemplateObjectManager` method), 16

`search()` (`extender.models.RoleTemplateObjectQuerySet` method), 16

`section` (`extender.models.ApiSubObject` attribute), 20

`section` (`extender.models.ConfigSubObject` attribute), 18

`section_id` (`extender.models.ApiSubObject` attribute), 21

`section_id` (`extender.models.ConfigSubObject` attribute), 19

`shortname` (`extender.models.ApiObject` attribute), 13

`shortname` (`extender.models.ApiSectionObject` attribute), 14

`shortname` (`extender.models.ApiSubObject` attribute), 19

`shortname` (`extender.models.ConfigSectionObject` attribute), 10

`shortname` (`extender.models.ConfigSubObject` attribute), 18

`shortname` (`extender.models.LogicalGroupObject` attribute), 9

`shortname` (`extender.models.RoleObject` attribute), 7

`shortname` (`extender.models.RoleTaskObject` attribute), 11

`shortname` (`extender.models.RoleTemplateObject` attribute), 16

`slug` (`extender.models.ApiObject` attribute), 13

`slug` (`extender.models.ApiSectionObject` attribute), 15

`slug` (`extender.models.ApiSubObject` attribute), 20

`slug` (`extender.models.ApiValueObject` attribute), 23

`slug` (`extender.models.ConfigSectionObject` attribute), 10

`slug` (`extender.models.ConfigSubObject` attribute), 18

`slug` (`extender.models.ConfigValueObject` attribute), 22

`slug` (`extender.models.LogicalGroupObject` attribute), 9

`slug` (`extender.models.RoleObject` attribute), 7

`slug` (`extender.models.RoleTaskObject` attribute), 12

`slug` (`extender.models.RoleTemplateObject` attribute), 16

`subobject` (`extender.models.ApiValueObject` attribute), 23

`subobject` (`extender.models.ConfigValueObject` attribute), 21

`subobject_id` (`extender.models.ApiValueObject` attribute), 23

`subobject_id` (`extender.models.ConfigValueObject` attribute), 22

T

`title` (`extender.models.ApiObject` attribute), 14

`title` (`extender.models.ApiSectionObject` attribute), 15

`title` (`extender.models.ApiSubObject` attribute), 20

`title` (`extender.models.ApiValueObject` attribute), 23

`title` (`extender.models.ConfigSectionObject` attribute), 11

`title` (`extender.models.ConfigSubObject` attribute), 18

`title` (`extender.models.ConfigValueObject` attribute), 22

`title` (`extender.models.LogicalGroupObject` attribute), 9

`title` (`extender.models.RoleObject` attribute), 8

`title` (`extender.models.RoleTaskObject` attribute), 12

`title` (*extender.models.RoleTemplateObject* attribute),
[17](#)
`trailingspace` (*extender.models.ApiSubObject* at-
tribute), [20](#)

U

`unique_slug_generator()` (in module *exten-
der.utils*), [24](#)

V

`value` (*extender.models.ApiValueObject* attribute), [23](#)
`value` (*extender.models.ConfigValueObject* attribute),
[21](#)