
Fleeting Forms

Release 0.0.1

Poplar Development

Jul 09, 2021

CONTENTS:

1	Client Onboarding	1
1.1	aka Creating a New Namespace	1
1.2	Creating a New Namespace	1
2	Fleeting Forms API Documentation	5
2.1	Models	5
2.2	Views	24
2.3	Forms	28
2.4	Serializers	29
2.5	Errors	33
2.6	Libraries and Utilities	33
2.7	Authentication	37
2.8	ReST Framework Permissions	37
2.9	Middleware	38
3	Indices and tables	39
	Python Module Index	41
	Index	43

CLIENT ONBOARDING

1.1 aka Creating a New Namespace

The fleetingforms service can be bundled and sold as part of a managed service. In this case, the service provider has a user account but the clients do not.

There are two approaches a service provider can take:

1. Maintain one namespace that all clients use.
2. Maintain a per-client namespace.

The first approach minimizes the administration overhead as only one namespace needs to be maintained. In the event that the namespace token (shared between all clients) is compromised, the overhead of reassigning the keys for all deployments could be very high.

The second approach requires the creation of a new namespace for each customer but enables per client usage tracking as well as a per-client token. Although an extra step is required to onboard a new client, it is very straightforward and easily automated.

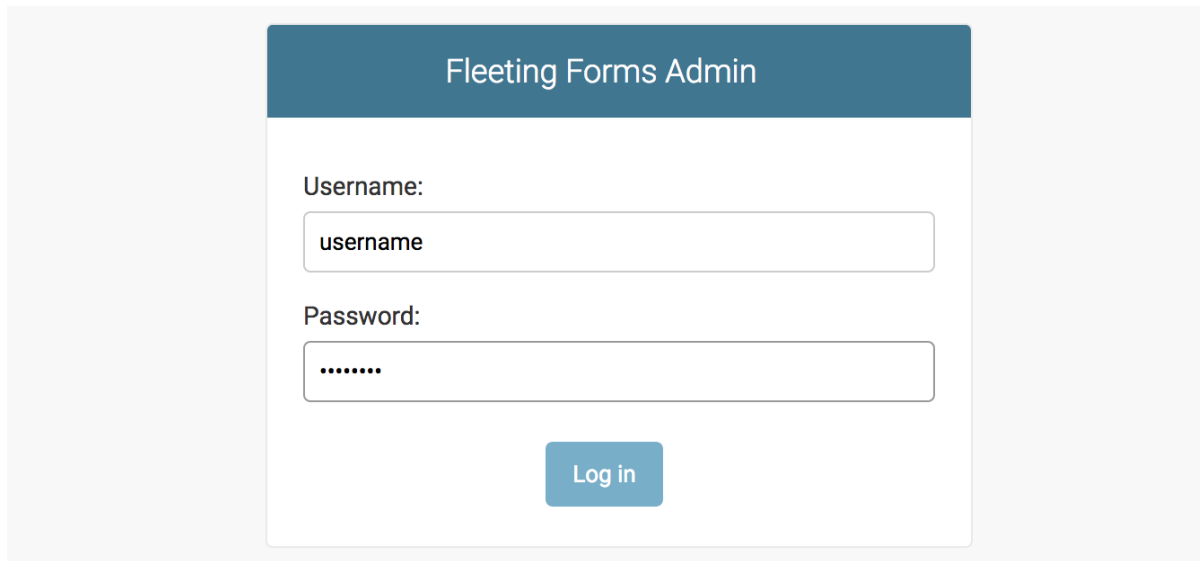
Once a client has been provisioned, they need never be changed, save for a token reset if required.

1.2 Creating a New Namespace

There are two ways to create a namespace: through the administration panel; using the API. The administration panel allows humans to administer namespaces while the API provides a way to automate namespace provisioning and billing.

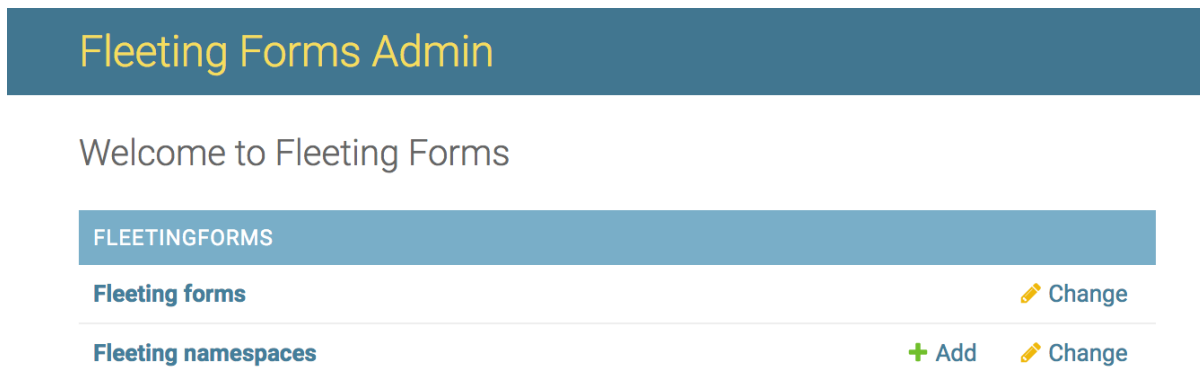
1.2.1 Using the Admin Panel

Visit <https://fleetingforms.io/admin/> and login with your username and password.



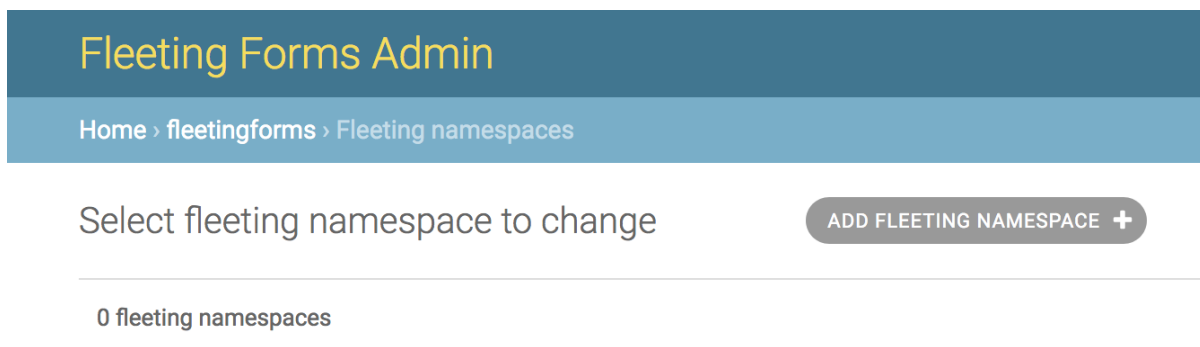
The image shows a login form for the Fleeting Forms Admin interface. It has a dark blue header with the text 'Fleeting Forms Admin'. Below the header, there are two input fields: 'Username:' with the text 'username' and 'Password:' with masked characters '.....'. A blue 'Log in' button is positioned below the password field.

On the Admin Landing Page, select the *Fleeting Namespace* section.



The image shows the Fleeting Forms Admin landing page. It has a dark blue header with the text 'Fleeting Forms Admin'. Below the header, there is a section titled 'Welcome to Fleeting Forms'. Underneath, there is a table with two rows: 'Fleeting forms' and 'Fleeting namespaces'. The 'Fleeting forms' row has a 'Change' button with a pencil icon. The 'Fleeting namespaces' row has a '+ Add' button and a 'Change' button with a pencil icon.

Select the + *Add Fleeting Namespace* button in the top right corner.



The image shows the Fleeting Forms Admin namespace management page. It has a dark blue header with the text 'Fleeting Forms Admin'. Below the header, there is a breadcrumb trail: 'Home > fleetingforms > Fleeting namespaces'. The main content area has the text 'Select fleeting namespace to change' and a button 'ADD FLEETING NAMESPACE +' in the top right corner. Below this, there is a section titled '0 fleeting namespaces'.

Complete the form, setting the client's preferences, and save it.

Fleeting Forms Admin

Home › fleetingforms › Fleeting namespaces › Add fleeting namespace

Add fleeting namespace

Subdomain:	<input type="text" value="mysubdomain"/>
User:	-
Token:	d8667752-05c5-4cfc-8691-be36963c2b45
Retention:	<input type="text" value="14"/>
Support email:	<input type="text" value="support@mycompany.com"/>
Url shortener:	<input type="text" value="fleeti.ng"/>
Style:	<input type="text"/>
Logo:	<input type="text" value="https://company.com/static/images/logo.png"/>

The namespace as now been assigned a token that can be shared it the user.

1.2.2 Using the API

Any user of the service can access the namespaces API to create or manage their namespaces. This allows managed service providers to automate the onboarding of new clients and the monitoring of per-client usage.

To create a new namespace over the API, start by posting a new namespace definition to <https://fleetingforms.io/namespaces/> path:

```
{
  "subdomain": "company-name",
  "retention": 180,
  "logo": "https://company-name.com/images/logo.png",
  "support_email": "accounting-support@company.com",
  "soft_limit": 1000,
  "hard_limit": 1500
}
```

This will return the details of the new namespace, including the client token:

```
{
  "id": 1567,
  "user": "myuser",
  "token": "6544d16d-5ef6-4924-9eee-54e3752b0640",
  "subdomain": "company-name",
  "retention": 180,
  "logo": "https://company-name.com/images/logo.png",
  "style": "",
  "support_email": "accounting-support@company.com",
  "soft_limit": 1000,
  "hard_limit": 1500
  "usage": {
    "this month": 0,
    "last month": 0
  }
}
```

The namespace has now been created along with associated counters and the token can be provided to the user.

FLEETING FORMS API DOCUMENTATION

Welcome to the Fleeting Forms API documentation. You're either here because you're curious, need to maintain something, or want to contribute. Welcome.

The Fleeting Forms project is build on [Django](#) using the [Django ReST Framework](#). Familiarize yourself with them before diving in.

2.1 Models

Models define the core data structures that support the application and are represented in the database.

With Fleeting Forms, the Form is at the heart of the application. They are grouped together in a *FleetingNamespace* and contain the *FleetingTemplate* and *FleetingAuth* that define the form.

class `fleetingform.models.FleetingAction(*args, **kwargs)`
Bases: `django.db.models.base.Model`

A form action.

All templates must have at least one action, which is rendered as the button that submits the form.

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.

label
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>`

template
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.

template_id

class `fleetingform.models.FleetingAuditEntry` (*args, **kwargs)

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

Fleeting Audit Entries track usage without personal details.

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

auth

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

code

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

completed_on

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

created_on

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

delete (*args, **kwargs)

get_auth_display (*, field=<django.db.models.fields.CharField: auth>)

get_next_by_created_on (*, field=<django.db.models.fields.DateTimeField: created_on>, is_next=True, **kwargs)

get_previous_by_created_on (*, field=<django.db.models.fields.DateTimeField: created_on>, is_next=False, **kwargs)

get_status_display (*, field=<django.db.models.fields.CharField: status>)

get_template_display (*, field=<django.db.models.fields.CharField: template>)

id

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

namespace

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.

namespace_id

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

opened_on

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

status

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

template

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

class `fleetingform.models.FleetingAuth(*args, **kwargs)`

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

Auth controls the authentication workflow for the form.

Every form has authentication parameters. Authentication parameters define how the authentication pages are displayed and which authentication workflow the user has to complete to access the form.

The simplest form of authentication, `none`, does not require any user authentication.

The most complex form, `user_otp_phone`, supports multiple users with individual one time pass codes delivered over SMS.

`AUTH_DEFAULT_CONTENT = 'Use your credentials to unlock the form.'`

`AUTH_DEFAULT_TITLE = 'Unlock Fleeting Form'`

`AUTH_TYPES = (('none', 'None'), ('user', 'Username'), ('pass', 'Password'), ('user_pass', 'User Password'))`

`AUTH_TYPES_EMAIL_OTP = ('otp_email', 'user_otp_email')`

`AUTH_TYPES_OTP = ('otp_phone', 'user_otp_phone', 'otp_email', 'user_otp_email')`

`AUTH_TYPES_OTP_ONLY = ('otp_email', 'otp_phone')`

`AUTH_TYPES_PASSWORD = ('pass',)`

`AUTH_TYPES_PHONE_OTP = ('otp_phone', 'user_otp_phone')`

`AUTH_TYPES_USER = ('user', 'user_pass', 'user_otp_email', 'user_otp_phone')`

`AUTH_TYPES_USERNAME = ('user',)`

`AUTH_TYPES_USEROTP = ('user_otp_email', 'user_otp_phone')`

`AUTH_TYPES_USERPASS = ('user_pass',)`

`AUTH_TYPE_NONE = 'none'`

`AUTH_TYPE_PASS = 'pass'`

`AUTH_TYPE_PASS_OTP_EMAIL = 'otp_email'`

`AUTH_TYPE_PASS_OTP_PHONE = 'otp_phone'`

`AUTH_TYPE_USER = 'user'`

`AUTH_TYPE_USER_PASS = 'user_pass'`

`AUTH_TYPE_USER_PASS_OTP_EMAIL = 'user_otp_email'`

`AUTH_TYPE_USER_PASS_OTP_PHONE = 'user_otp_phone'`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

action

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

authenticate (*username*="", *password*="")

Authenticate a user and password.

Given the auth type, authenticate the given username and password.

Parameters

- **username** (*str*) – the username to authenticate.
- **password** (*str*) – the plain text password to authenticate.

Returns True if username and password valid, else False

Return type bool

content

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

property email_otp

Does this form use email for the one time pass?

Returns True if username and email one time pass required, else False

Return type bool

form

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant (Model) :  
    place = OneToOneField(Place, related_name='restaurant')
```

`Place.restaurant` is a `ReverseOneToOneDescriptor` instance.

form_controls

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.

get_type_display (*, *field*=<django.db.models.fields.CharField: type>)

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>

opened_by (*username*)

Set the `opened_by` timestamp for a user on access.

Parameters **username** (*str*) – the user that opened the form.

Returns None

property otp

Does this form require a one time pass?

Returns True if one time pass required, else False

Return type bool

property password_only

Does this form require a password only.

Returns True if password only authentication required, else False

Return type bool

property phone_otp

Does this form use phone for the one time pass?

Returns True if username and phone one time pass required, else False

Return type bool

property required

Does this form require authentication?

Returns True if authentication required, else False

Return type bool

property requires_password

Does this form require a password of some kind? OTP or Static.

Returns True if password required, else False

Return type bool

property requires_username

Does this form require a username?

Returns True if username required, else False

Return type bool

title

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

type

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

property username_and_otp

Does this form require a username and one time passcode?

Returns True if username and static password required, else False

Return type bool

property username_and_password

Does this form require a username and static password.

Returns True if username and static password required, else False

Return type bool

property username_only

Does this form require a username only.

Returns True if username only authentication required, else False

Return type bool

users

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.

verify_password(password, password_hash)

Verify that a plain text password matches a hash.

Automatically identifies and instantiates the correct hasher and performs the hash verification.

Parameters password (str) – password to verify

Returns True if password matches hash, else False

Return type bool

verify_username(username="")

Verify that a username is valid for the form.

Checks whether the username is present in users.

Parameters username (str) – username to verify

Returns True if user is present, else False

Return type bool

class fleetingform.models.FleetingChoice(*args, **kwargs)

Bases: django.db.models.base.Model

A choice field entry.

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

form_control

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.

form_control_id

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>

text

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

value

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

class `fleetingform.models.FleetingForm(*args, **kwargs)`

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

The whole point.

Fleeting forms provide a quick way to set up a one-off interaction with someone.

All forms have a template, which defines how it is rendered and which workflow is executed when the user completes the form.

Easy to use and setup, the default form comes with a title, space for instructions or a message to the user, and whichever form fields and action buttons you need.

Create a new default form:

```
{
    'template': {
        'title': 'Page Title',
        'content': 'Hi Ethel!'
    }
}
```

Some instructions.',

```
        'form_controls':[...], 'actions': ['Submit'] }
    }
}
```

This will return a response that contains a one time code for the the form along with a new URL to access it.

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

FORM_STATUSES = (('created', 'Created'), ('opened', 'Opened'), ('completed', 'Completed'))

FORM_STATUSES_COMPLETED = {'completed', 'error'}

FORM_STATUS_COMPLETED = 'completed'

FORM_STATUS_CREATED = 'created'

FORM_STATUS_ERROR = 'error'

FORM_STATUS_OPENED = 'opened'

exception `MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

app

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

property `audit_entry`

auth

Accessor to the related object on the forward side of a one-to-one relation.

In the example:

```
class Restaurant (Model):  
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` instance.

auth_id**auth_token**

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

code

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

complete (*result={}*)

Put the form into the completed state.

Parameters *result* (*JSON-serializable dict*) – the result for the `FleetingForm`

Raises `FleetingFormCompleteError`

property completed

Is this form complete? Either the user has submitted or error.

Returns True if form complete or errored, else False

Return type bool

completed_on

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

created_on

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

error (*error_code, message, force_status=False*)

Put the form into an error state.

Parameters

- **error_code** (*str*) – the error code that was triggered.
- **message** (*str*) – a descriptive message to the user.
- **force_status** (*bool*) – force the transition even if the form is already complete?

Raises `FleetingFormCompleteError`

property expired

Has the form expired?

Returns True if expired else False

Return type bool

property expires_on

When does the form expire?

Returns form expiry date

Return type `datetime.datetime`

get_next_by_created_on (*, *field*=<`django.db.models.fields.DateTimeField: created_on`>, *is_next*=True, ***kwargs*)

get_previous_by_created_on (*, *field*=<`django.db.models.fields.DateTimeField: created_on`>, *is_next*=False, ***kwargs*)

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

id

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

namespace

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.

namespace_id

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

open (*username*=None)

Open the form for the first time.

Parameters *username* (*str*) – the user that opened the form.

Raises `FleetingFormCompleteError`

opened_on

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

result

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

save (**args*, ***kwargs*)

Inhibit the Form from saving if it is already complete.

Parameters *force_status* (*bool*) – force the save, overriding the status check.

Returns the saved instance.

Return type `FleetingForm`

Raises `FleetingFormCompleteError`

property short_url

Short URL based on the namespace URL shortener and scheme settings.

Returns URL to the form.

Return type `str`

status

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

template

Accessor to the related object on the forward side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` instance.

template_id**property url**

Full length URL based on the default domain and scheme settings.

Returns URL to the form.

Return type str

class `fleetingform.models.FleetingFormControl(*args, **kwargs)`

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

A form control.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

FIELD_TYPES = (('text', 'Text'), ('textarea', 'Text Area'), ('integer', 'Integer'), ('

FIELD_TYPE_BOOLEAN = 'boolean'

FIELD_TYPE_CHOICE = 'choice'

FIELD_TYPE_DATE = 'date'

FIELD_TYPE_DATETIME = 'datetime'

FIELD_TYPE_DECIMAL = 'decimal'

FIELD_TYPE_EMAIL = 'email'

FIELD_TYPE_FLOAT = 'float'

FIELD_TYPE_INTEGER = 'integer'

FIELD_TYPE_TEXT = 'text'

FIELD_TYPE_TEXTAREA = 'textarea'

FIELD_TYPE_TIME = 'time'

FIELD_TYPE_URL = 'url'

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

choices

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.

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.

content_type_id

disabled

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

form

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.

get_type_display (*, *field*=<*django.db.models.fields.CharField: type*>)

help_text

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

hidden

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.

initial

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

label

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.

object_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*>

params

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

required

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

type

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

validations

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 fleetingform.models.FleetingNamespace(*args, **kwargs)

Bases: django.db.models.base.Model

Namespaces are a honking great idea - group forms and settings.

Each namespace groups FleetingForms that require consistent settings and styling, i.e. those supporting a single application or user. All forms in a namespace share the same subdomain (<https://subdomain.fleetingforms.io/>), styling, support email.

Users access the a namespace by providing the correct token in the settings.FLEETING_TOKEN_HEADER HTTP header.

CH_URL_UNSAFE = '< > # % " ; / ? : @ & = + \$, { } | \\'

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

RE_URL_UNSAFE = '[\\[\\]\\<>#%";/?:@&=+\$,{ }|.\\'

URL_FLEETING_LINK = 'fleeting.link'

URL_FLEETING = 'fleeti.ng'

URL_SHORTENERS = (('fleeti.ng', 'fleeti.ng'), ('fleeting.link', 'fleeting.link'))

property active_forms**audit_entries**

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.

delete(*args, **kwargs)

forms

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.

forms_between (*_from*, *_to*)

property `forms_last_month`

forms_since (*date*)

property `forms_this_month`

classmethod `from_request` (*request*)

Get the namespace from a Django Request object.

Parameters `request` (*Django Request*) – the request to extract the namespace from.

Returns `FleetingNamespace` that matches the token or `None`

Return type `FleetingNamespace` or `None`

get_url_shortener_display (*, *field*=<*django.db.models.fields.CharField: url_shortener*>)

hard_limit

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.

logo

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*>

retention

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

save (**args*, ***kwargs*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The ‘`force_insert`’ and ‘`force_update`’ parameters can be used to insist that the “save” must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

soft_limit

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

style

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

subdomain

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

support_email

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

token

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

property total_forms

total_forms_between (*_from, _to*)

total_forms_since (*date*)

url_shortener

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**webhooks**

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 `fleetingform.models.FleetingTemplate` (**args, **kwargs*)

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

Templates control how the form is displayed and which actions are taken.

Every form has a template. The template defines which HTML page will be rendered and which actions will be taken when the user submits the form.

The simplest template, ‘generic’, renders a title, introductory content, and a single form. On submit, the contents of the form are saved in the results.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

```

TEMPLATE_CONTENT_TYPES = (('md', 'Markdown'), ('rst', 'ReStructured Text'), ('txt', 'P
TEMPLATE_CONTENT_TYPE_MARKDOWN = 'md'
TEMPLATE_CONTENT_TYPE_PLAINTEXT = 'txt'
TEMPLATE_CONTENT_TYPE_RESTRUCTURED = 'rst'
TEMPLATE_HELPERS = {'generic': 'fleetingform.template_helpers.generic.GenericTemplate
TEMPLATE_TYPES = (('generic', 'Generic'), ('stripe', 'Stripe'), ('paypal', 'Paypal'))
TEMPLATE_TYPE_GENERIC = 'generic'
TEMPLATE_TYPE_PAYPAL = 'paypal'
TEMPLATE_TYPE_STRIPE = 'stripe'

```

actions

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.

content

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

content_type

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

form

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```

class Restaurant(Model):
    place = OneToOneField(PPlace, related_name='restaurant')

```

`Place.restaurant` is a `ReverseOneToOneDescriptor` instance.

form_controls

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.

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

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

property helper

Get the helper class name for this template.

Returns helper class full dotted path.

Return type str

helper_class ()

Get the helper class.

Returns class object for this template type's helper.

Return type class

Raises ImportError if helper cannot be found.

property html_template

Get the HTML template path for this Template

Returns full static HTML template path.

Return type str

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>

params

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

property supported_templates

Get the list of supported templates.

Returns supported template names.

Return type [str,]

classmethod template_helper_class_for (_type)

Get the helper class for any supported template type.

Returns class object for this template type's helper.

Return type class

Raises KeyError if the type isn't valid, ImportError if helper cannot be found.

title

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

type

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

class fleetingform.models.FleetingUser (*id, auth, username, password, email, phone, opened_on, attempts*)

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

attempts

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

auth

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.

auth_id**authenticate** (*plain_password*)

Authenticate this user with plain_password.

Parameters **plain_password** (*str*) – the plain text password to authenticate

Returns True is the password matches, else False

Return type bool

email

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

encrypt_and_set_password (*plain_password*)

Encrypt and set the user password.

Parameters **plain_password** (*str*) – the password to encrypt and set.

generate_and_send_otp ()

Generate and send a new one time passcode to the user.

Raises FleetingAuthOTPError on OTP send failure.

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>

opened_on

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

property otp_contact_obscured

The contact information used to send OTP, obscured.

Hides the email or phone number with some x's.

Returns email or phone for this user with some x's

Return type str

password

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

phone

The descriptor for the phone number attribute on the model instance. Returns a PhoneNumber when accessed so you can do stuff like:

```
>>> instance.phone_number.as_international
```

Assigns a phone number object on assignment so you can do:

```
>>> instance.phone_number = PhoneNumber(...)
```

or,

```
>>> instance.phone_number = '+414204242'
```

username

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

```
class fleetingform.models.FleetingValidation(*args, **kwargs)
```

Bases: django.db.models.base.Model

A custom field validation.

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

```
VALIDATION_TYPES = (('regex', 'Regular Expression'), ('min-length', 'Minimum Length'),
```

```
VALIDATION_TYPE_MAX = 'max'
```

```
VALIDATION_TYPE_MAX_LENGTH = 'max-length'
```

```
VALIDATION_TYPE_MIN = 'min'
```

```
VALIDATION_TYPE_MIN_LENGTH = 'min-length'
```

```
VALIDATION_TYPE_RANGE = 'range'
```

```
VALIDATION_TYPE_RANGE_LENGTH = 'range-length'
```

```
VALIDATION_TYPE_REGEX = 'regex'
```

form_control

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.

form_control_id

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

id

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

message

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>

params

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

type

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

```
class fleetingform.models.FleetingWebhook(*args, **kwargs)
```

Bases: django.db.models.base.Model

Webhooks for namespaces.

Enables a user-configurable callback when a form changes state.

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

```
WEBHOOK_EVENTS = (('create', 'Created'), ('complete', 'Completed'), ('error', 'Error'))
```

```
WEBHOOK_EVENT_ACCESS = 'access'
```

```
WEBHOOK_EVENT_COMPLETE = 'complete'
```

```
WEBHOOK_EVENT_CREATE = 'create'
```

```
WEBHOOK_EVENT_ERROR = 'error'
```

event

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

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

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.

namespace

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.

namespace_id

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

token

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

url

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

2.2 Views

The views respond to browser requests and control the response that is rendered.

```
class fleetingform.views.FleetingFormListView(**kwargs)
```

```
    Bases: rest_framework.generics.ListCreateAPIView
```

```
    authentication_classes = [<class 'fleetingform.authentication.FleetingAuthentication'>]
```

```
    get_queryset()
```

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

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

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

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

```
    perform_create(serializer)
```

```
    permission_classes = [<class 'fleetingform.permissions.FleetingFormTokenPermission'>]
```

```
    serializer_class
```

```
        alias of fleetingform.serializers.FleetingFormSerializer
```

```
class fleetingform.views.FleetingFormLoginView(**kwargs)
```

```
    Bases: django.views.generic.base.View
```

Handle Fleeting Form Logins

Overall workflow depends on the authentication type configured.

1. Username only: - Display username form.
 - on match proceed
 - on fail return to
2. Password only: - Display password form
 - on match proceed
 - on fail return to
3. Username and Password: 1. Username and password provided:
 - Display Userpass form - on match proceed - on fail return to
1. Username and otp: - Display Username form
 - on match, continue

- Display password form - on match, proceed - on cancel, back to username form - on fail return to

Workflows

USERNAME

- sets auth_token

PASSWORD

- sets auth_token

USERPASS

- sets auth_token

USERPASS_OTP

- sets auth_username
- sets auth_token

get (*request, code*)

Handle a get request for a Fleeting Form Login.

Checks if the form requires authentication. If so, and not authenticated, renders login. If not, or authenticated, redirects to form.

Parameters

- **request** (*django.http.HttpRequest*) – the request
- **code** (*str*) – the one time code for the form extracted from the url

Returns If auth required render, else redirect to form

Return type `django.views.generic.base.RedirectView` or `django.template.response.TemplateResponse`

otp_instructions = 'Unlock this form with your one time code.'

password_instructions = 'Unlock this form with a password.'

post (*request, code*)

Handle a get request for a Fleeting Form Login.

Check if the authentication form is valid. If so, set session variables and either redirect to the form or render the next authentication stage.

Parameters

- **request** (*django.http.HttpRequest*) – the request
- **code** (*str*) – the one time code for the form extracted from the url

Returns If further auth required or invalid render, else redirect to form

Return type `django.views.generic.base.RedirectView` or `django.template.response.TemplateResponse`

username_instructions = 'Unlock this form with a username.'

userpass_instructions = 'Unlock this form with a username and password.'

class `fleetingform.views.FleetingFormRetrieveDestroyView` (***kwargs*)

Bases: `rest_framework.generics.RetrieveDestroyAPIView`

authentication_classes = [`<class 'fleetingform.authentication.FleetingAuthentication'>`]

get_queryset ()

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

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

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

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

```
permission_classes = [<class 'fleetingform.permissions.FleetingFormTokenPermission'>]
```

serializer_class

alias of *fleetingform.serializers.FleetingFormSerializer*

```
class fleetingform.views.FleetingNamespaceListView (**kwargs)
```

Bases: *rest_framework.generics.ListCreateAPIView*

```
authentication_classes = [<class 'rest_framework.authentication.TokenAuthentication'>]
```

get_queryset ()

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

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

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

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

perform_create (serializer)

```
permission_classes = [<class 'fleetingform.permissions.FleetingNamespaceTokenPermission'>]
```

serializer_class

alias of *fleetingform.serializers.FleetingNamespaceSerializer*

```
class fleetingform.views.FleetingNamespaceRetrieveUpdateView (**kwargs)
```

Bases: *rest_framework.mixins.RetrieveModelMixin*, *rest_framework.mixins.UpdateModelMixin*, *rest_framework.generics.GenericAPIView*

```
authentication_classes = [<class 'fleetingform.authentication.FleetingAuthentication'>]
```

get (request, *args, **kwargs)**get_queryset ()**

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

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

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

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

```
permission_classes = [<class 'fleetingform.permissions.FleetingNamespaceTokenPermission'>]
```

put (request, *args, **kwargs)

```

serializer_class
    alias of fleetingform.serializers.FleetingNamespaceSerializer

```

class `fleetingform.views.FleetingNamespaceRetrieveView` (***kwargs*)

Bases: `rest_framework.generics.RetrieveUpdateAPIView`

authentication_classes = [`<class 'fleetingform.authentication.FleetingAuthentication'>`]

get_queryset ()

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

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

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

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

permission_classes = [`<class 'fleetingform.permissions.FleetingNamespaceTokenPermission'>`]

```

serializer_class
    alias of fleetingform.serializers.FleetingNamespaceSerializer

```

class `fleetingform.views.FleetingOTPResetRedirectView` (***kwargs*)

Bases: `django.views.generic.base.RedirectView`

Reset and resend the OTP for a form.

For two stage auth types, silently noops for sessions in which the username hasn't been verified yet.

Otherwise, forces a new passcode to be generated and sent.

get_redirect_url (**args, **kwargs*)

If this is an OTP secured form, and the user is set if required, reset and resend the otp, then redirect to the form login view.

pattern_name = 'form-login'

permanent = **False**

class `fleetingform.views.UserFormView` (***kwargs*)

Bases: `django.views.generic.base.View`

User facing forms.

This view is responsible for rendering the forms and handling user input.

decode_query_params (*params={}*)

Decode query params.

Query params can be sent as base64 encoded values when prefixed with `b64:`. Decode query params into a flat dict of decoded `key: value` pairs.

Parameters *params* (*dict-like*) – GET and/or POST querydict contents

Returns all decoded `key: value` pairs.

Return type dict

get (*request, code*)

Handle a get request for a Fleeting Form.

Checks is the form requires authentication. If so, and not authenticated, redirect to login workflow. If not, or authenticated, render dynamically generated form.

Parameters

- **request** (*django.http.HttpRequest*) – the request
- **code** (*str*) – the one time code for the form extracted from the url

Returns If auth required redirect, else render form

Return type *django.views.generic.base.RedirectView* or *django.template.response.TemplateResponse*

post (*request, code*)

Handle a post request for a Fleeting Form.

Checks is the form requires authentication. If so, and not authenticated, redirect to login workflow. If not, or authenticated, create dynamically generated form and check validity.

If valid, save and complete form. If invalid, render form with errors.

Parameters

- **request** (*django.http.HttpRequest*) – the request
- **code** (*str*) – the one time code for the form extracted from the url

Returns If auth required redirect, else render form.

Return type *django.views.generic.base.RedirectView* or *django.template.response.TemplateResponse*

fleetingform.views.send_otp_and_set_messages (*user, fform, request*)

2.3 Forms

There are two types of forms in this application: relatively static login forms and dynamic FleetingForms. This stuff is all for the former.

class *fleetingform.forms.FleetingAuthenticationForm* (*fform, *args, **kwargs*)

Bases: *django.forms.forms.Form*

Base authentication form that sets the field labels and help text.

This abstract base class looks after the customization of the stock authentication form (auto-generated) with the configuration in a Fleeting Form's authentication configuration.

Parameters **fform** (*FleetingForm*) – *FleetingForm* the authentication page is for.

class *fleetingform.forms.PasswordAuthenticationForm* (*fform, username="", *args, **kwargs*)

Bases: *fleetingform.forms.FleetingAuthenticationForm*

Authenticate based on password only.

This form may be used either for static password or otp auth types.

Parameters **fform** (*FleetingForm*) – *FleetingForm* the authentication page is for.

clean ()

Hook for doing any extra form-wide cleaning after *Field.clean()* has been called on every field. Any *ValidationError* raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

class *fleetingform.forms.UserAuthenticationForm* (*fform, *args, **kwargs*)

Bases: *fleetingform.forms.FleetingAuthenticationForm*

Authenticate based on username only.

Parameters **fform** (*FleetingForm*) – *FleetingForm* the authentication page is for.

clean()

Hook for doing any extra form-wide cleaning after *Field.clean()* has been called on every field. Any *ValidationError* raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

class *fleetingform.forms.UserPasswordAuthenticationForm* (*fform*, **args*, ***kwargs*)

Bases: *fleetingform.forms.FleetingAuthenticationForm*

Authenticate based on username and password.

This form may be used either for static username and password or user and otp auth types.

Parameters **fform** (*FleetingForm*) – *FleetingForm* the authentication page is for.

clean()

Hook for doing any extra form-wide cleaning after *Field.clean()* has been called on every field. Any *ValidationError* raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

2.4 Serializers

Serializers transform models to and from JSON representations for the ReST API.

This is where you'll find inbound data validation.

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

Bases: *rest_framework.serializers.ModelSerializer*

class Meta

Bases: *object*

fields = `['label']`

model

alias of *fleetingform.models.FleetingAction*

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

Bases: *drf_writable_nested.serializers.WritableNestedModelSerializer*

class Meta

Bases: *object*

fields = `['type', 'title', 'content', 'form_controls', 'users']`

model

alias of *fleetingform.models.FleetingAuth*

REQUIRED_FORM_CONTROLS = `{'password', 'username'}`

REQUIRED_FORM_CONTROL_DEFAULTS = `{'password': {'label': 'Password', 'name': 'password'}}`

create (*validated_data*)

Overridden create to add any required form controls.

```
    to_representation (instance)
        Exclude empty or null keys from serialized responses.

    validate (data)
        Validate correct user arguments have been passed given the type

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

Bases: rest_framework.serializers.ModelSerializer

class Meta
    Bases: object

    fields = ['value', 'text']

    model
        alias of fleetingform.models.FleetingChoice

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

Bases: drf_writable_nested.serializers.WritableNestedModelSerializer

class Meta
    Bases: object

    fields = ['type', 'name', 'help_text', 'label', 'validations', 'choices', 'required']

    model
        alias of fleetingform.models.FleetingFormControl

    to_representation (instance)
        Exclude empty or null keys from serialized responses.

    validate (data)
        Validate that the control has a valid combination of options.

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

Bases: drf_writable_nested.serializers.WritableNestedModelSerializer

class Meta
    Bases: object

    fields = ['id', 'code', 'template', 'app', 'auth', 'url', 'short_url', 'status', 'c

    model
        alias of fleetingform.models.FleetingForm

    create (validated_data)
        We have a bit of extra checking around this in order to provide descriptive messages when something goes
        wrong, but this method is essentially just:

        return ExampleModel.objects.create(**validated_data)

        If there are many to many fields present on the instance then they cannot be set until the model is instanti-
        ated, in which case the implementation is like so:
```

```

    example_relationship = validated_data.pop('example_relationship') instance = Example-
    Model.objects.create(**validated_data) instance.example_relationship = example_relationship
    return instance

```

The default implementation also does not handle nested relationships. If you want to support writable nested relationships you'll need to write an explicit *.create()* method.

validate_app (*val*)

Validate that the app parameters are within bounds.

validate_auth (*data*)

Populate default auth type for new creates without type.

validate_namespace (*val*)

Populate the namespace from the request for new creates.

```

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

```

Bases: `rest_framework.serializers.ModelSerializer`

class Meta

Bases: `object`

fields = ['id', 'user', 'subdomain', 'url_shortener', 'retention', 'usage', 'support']

model

alias of `fleetingform.models.FleetingNamespace`

get_usage (*namespace*)

to_representation (*instance*)

Exclude token unless user is authenticated.

```

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

```

Bases: `drf_writable_nested.serializers.WritableNestedModelSerializer`

class Meta

Bases: `object`

fields = ['type', 'title', 'content', 'content_type', 'form_controls', 'params', 'a']

model

alias of `fleetingform.models.FleetingTemplate`

validate (*data*)

Validate that the template has a valid combination of options.

validate_actions (*actions*)

Validate the template actions.

validate_content (*content*)

Bleach any inbound HTML in the form content.

validate_title (*title*)

Bleach any inbound HTML in the form title.

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

Bases: rest_framework.serializers.ModelSerializer

class Meta
    Bases: object

    fields = ['username', 'password', 'email', 'phone', 'opened_on']

    model
        alias of fleetingform.models.FleetingUser

to_representation (instance)
    Exclude empty or null keys and hide password hashes.

validate_password (data)
    Validate an inbound password.

    Passwords may only come in two formats: - a string prefixed with plain: - PHC string format (https://github.com/P-H-C/phc-string-format/blob/master/phc-sf-spec.md)

    Plain text passwords are immediately hashed before being stored. PHC strings are checked for formatting
    and valid hashing type.

validate_phone (data)
    If enabled, use the twilio lookup API to validate a phone number.

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

Bases: rest_framework.serializers.ModelSerializer

class Meta
    Bases: object

    fields = ['type', 'params', 'message']

    model
        alias of fleetingform.models.FleetingValidation

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

Bases: rest_framework.serializers.ModelSerializer

class Meta
    Bases: object

    fields = ['id', 'namespace', 'url', 'token', 'name', 'event']

    model
        alias of fleetingform.models.FleetingWebhook
```

2.5 Errors

Custom error classes used in the application.

exception `fleetingform.errors.FleetingAuthOTPErrror` (*user*, *auth_type*, **args*, ***kwargs*)
 Raised when creating, saving, or sending a one time passcode fails.

Parameters

- **user** (*fleetingform.models.FleetingUser*) – User the OTP was being generated for.
- **auth_type** (*str* from *FleetingAuth.AUTH_TYPES*) – Authentication type of the form.

exception `fleetingform.errors.FleetingDeletionError` (*klass*, *pk*, **args*, ***kwargs*)
 Raised when you try to delete an object that cannot be deleted.

Parameters

- **klass** (*str*) – object class
- **pk** (*object*) – object primary key

exception `fleetingform.errors.FleetingFormCompleteError` (*code*, **args*, ***kwargs*)
 Raised when an attempt is made to save a form that is already complete.

Parameters **code** (*str*) – The form's one time code.

exception `fleetingform.errors.FleetingFormGenerationError`
 Raised when for generation fails.

exception `fleetingform.errors.FleetingOTPRetriesExceeded` (*user*, **args*, ***kwargs*)
 Raised when the max number of OTP resends is exceeded.

Parameters **user** (*fleetingform.models.FleetingUser*) – the user for which the retries were exceeded.

exception `fleetingform.errors.FleetingValidationError` (*field*, *code*, **args*, ***kwargs*)
 Raised while validating requests.

Using a message map in place of handwritten Exception messages provides a more consistent user experience when creating forms.

Parameters

- **field** (*str*) – the field with the value that failed validation.
- **code** (*str*) – the code of the form.

2.6 Libraries and Utilities

A collection of things that are used throughout the application. This includes helpful functions for sending OTPs, the form generator, and other sundry.

`fleetingform.lib.auth_token_field` (*fform*)
 Session field housing the auth_token for a fform.

`fleetingform.lib.auth_username_field` (*fform*)
 Session field housing the verified username for a fform.

`fleetingform.lib.send_otp_email(email, otp, fform)`
Send an email one time passcode.

Parameters

- **email** (*str*) – email to send to
- **otp** (*str*) – the one time passcode to send

Returns True if sent, else False

Return type bool

`fleetingform.lib.send_otp_sms(phone, otp, fform)`
Send an SMS one time passcode.

Parameters

- **phone** (*str* (*E.164 formatting*)) – phone number to send to
- **otp** (*str*) – the one time passcode to send

Returns True if sent, else False

Return type bool

`fleetingform.lib.form_generator.generate_fleeting_form_class_for(form_controls,`
`query_params={})`

Generate a class for the given form and query parameters.

Parameters

- **form_info** (*dict*) – a dictionary that describes a Fleeting Form.
- **query_params** (*django.http.QueryDict*) – URL query parameters.

Fleeting forms are all unique. To take advantage of Django's form handling we need to generate a new class for each one.

This generator accepts a form described in a dictionary, using the Fleeting Form shorthand.

```
"template": {
    # Actions are rendered as buttons at the bottom of the
    # form. The name of the button clicked is stored in
    # the ``action`` key of the result.
    "actions": ["Submit", "Cancel"],

    # Controls are the elements on the page the user interacts
    # with. There can be up to 32 controls on a Fleeting Form.
    # Controls will be rendered in the order they are listed.
    "form_controls": [

        # Controls have a name, type, optional help text, label, and
        # initial value. Any field can be required for the form to
        # be complete.
        {
            "name": "comments",
            "type": "textarea",
            "label": "User Comments",
            "required": true,
            "initial": "What did you think?"
        },
        {
            "name": "toes",
```

(continues on next page)

(continued from previous page)

```

        "type": "integer",
        "label": "How many toes?",
        "required": true,
        "initial": 10
    },
    {
        "name": "Pies",
        "type": "float",
        "label": "How many pies are left?",
        "required": true,
        "initial": 3.25
    },
    # Fleeting forms automatically validate user input and render
    # the correct form elements for you. Use the correct field
    # type to get the most out of your form.
    {
        "name": "pet_name",
        "type": "text",
        "label": "Pet's Name",
        "required": false,
        "initial": "fido",
        "help_text": "The short version."
        "validations": [
            {
                "type": "max-length",
                "params": {"max-length": 32},
                "message": "Names must be 32 characters or fewer."
            }
        ]
    },
    # Special field types are supported to provide extra
    # validation for URLs and Emails.
    {
        "name": "website",
        "type": "url",
        "label": "Your Website",
        "required": true,
        "initial": "https://"
    },
    {
        "name": "email",
        "type": "email",
        "label": "Your Email",
        "required": true,
        "help_text": "We will never send email without asking."
    },
    # For restricting user input, try the choices field. This
    # will render a select and keep track of what the user
    # chooses.
    {
        "name": "breed",
        "type": "choice",
        "label": "Breed",
        "required": true,

```

(continues on next page)

(continued from previous page)

```

        "choices": [
            ["lab", "Labrador"],
            ["shepherd", "Shepherd"],
            ["collie", "Collier"],
            ["burmese", "Burmese Mountain Dog"]
        ]
    },

    # Dates and times use a special form widget so the user can
    # select from a calendar or time picker.
    {
        "name": "incident_datetime",
        "type": "datetime",
        "label": "Incident Date and Time",
        "required": true
    },
    {
        "name": "moms_bday",
        "type": "date",
        "label": "Mom's Birthday",
        "required": true
    },
    {
        "name": "callback_time",
        "type": "time",
        "label": "Preferred Callback Time",
        "required": false
    },

    # Boolean fields can be used to implement "I have read and..."
    # forms, simply set them to required.
    {
        "name": "accept_terms",
        "type": "boolean",
        "label": "I have read and accept and terms.",
        "required": true
    },

    ]
}

```

For any field that does not have an initial value in the Fleeting Form dictionary, if the field name is a key in the GET query params then the query param value will be used as the initial value.

Use this to customize a form default values, by adding a username or other custom touch, to a form that is distributed to many people.

2.7 Authentication

```
class fleetingform.authentication.FleetingAuthentication
```

ReST Framework Authentication handler for Namespace Token.

```
    authenticate (request)
```

Authenticate a request, returns the user and namespace.

2.8 ReST Framwork Permissions

Permissions are used to check whether a particular user can see an object or write to a namespace. Fleeting Form's API permissions are implemented using the ReST Framework's [permissions](#), so have a look there for more background.

```
class fleetingform.permissions.FleetingFormHardLimitPermission
```

Bases: `rest_framework.permissions.BasePermission`

Do not allow creates for users over their hard limit.

```
    has_permission (request, view)
```

Return *True* if permission is granted, *False* otherwise.

```
    message = 'Namespace hard limit exceeded.'
```

```
class fleetingform.permissions.FleetingFormTokenPermission
```

Bases: `rest_framework.permissions.BasePermission`

Require the presence of the correct namespace token.

```
    has_object_permission (request, view, obj)
```

Return *True* if permission is granted, *False* otherwise.

```
    has_permission (request, view)
```

Return *True* if permission is granted, *False* otherwise.

```
    message = 'A valid token and namespace are required.'
```

```
class fleetingform.permissions.FleetingNamespaceTokenPermission
```

Bases: `rest_framework.permissions.BasePermission`

Require the presence of the correct namespace token or user token.

```
    has_object_permission (request, view, obj)
```

Return *True* if permission is granted, *False* otherwise.

```
    has_permission (request, view)
```

Users can create new namespaces, namespace tokens cannot. Users and namespace tokens can both list.

```
    message = 'A valid token is required.'
```

2.9 Middleware

Middleware manipulates the request before it is passed to the view and the response before it is returned to the client.

To learn more about Django's middleware implementation and how the class has been implemented, have a look at the *middleware docs*.

```
class fleetingform.middleware.FleetingNamespaceMiddleware (get_response)
```

```
    Bases: object
```

```
    Manage the FleetingNamespace in the request.
```

```
    Implements the Django Middleware pattern. It should be in the middleware stack somewhere after the AuthenticationMiddleware.
```

```
    Adds the .namespace property to the request object. Set to None if no token in header.
```

INDICES AND TABLES

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

PYTHON MODULE INDEX

f

- `fleetingform.authentication`, [37](#)
- `fleetingform.errors`, [33](#)
- `fleetingform.forms`, [28](#)
- `fleetingform.lib`, [33](#)
- `fleetingform.lib.form_generator`, [34](#)
- `fleetingform.middleware`, [38](#)
- `fleetingform.models`, [5](#)
- `fleetingform.permissions`, [37](#)
- `fleetingform.serializers`, [29](#)
- `fleetingform.views`, [24](#)

A

- `action` (*fleeingform.models.FleetingAuth* attribute), 7
- `actions` (*fleeingform.models.FleetingTemplate* attribute), 19
- `active_forms()` (*fleeingform.models.FleetingNamespace* property), 16
- `app` (*fleeingform.models.FleetingForm* attribute), 11
- `attempts` (*fleeingform.models.FleetingUser* attribute), 20
- `audit_entries` (*fleeingform.models.FleetingNamespace* attribute), 16
- `audit_entry()` (*fleeingform.models.FleetingForm* property), 11
- `auth` (*fleeingform.models.FleetingAuditEntry* attribute), 6
- `auth` (*fleeingform.models.FleetingForm* attribute), 11
- `auth` (*fleeingform.models.FleetingUser* attribute), 21
- `AUTH_DEFAULT_CONTENT` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_DEFAULT_TITLE` (*fleeingform.models.FleetingAuth* attribute), 7
- `auth_id` (*fleeingform.models.FleetingForm* attribute), 12
- `auth_id` (*fleeingform.models.FleetingUser* attribute), 21
- `auth_token` (*fleeingform.models.FleetingForm* attribute), 12
- `auth_token_field()` (in module *fleeingform.lib*), 33
- `AUTH_TYPE_NONE` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPE_PASS` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPE_PASS_OTP_EMAIL` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPE_PASS_OTP_PHONE` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPE_USER` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPE_USER_PASS` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPE_USER_PASS_OTP_EMAIL` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPE_USER_PASS_OTP_PHONE` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPES` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPES_EMAIL_OTP` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPES_OTP` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPES_OTP_ONLY` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPES_PASSWORD` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPES_PHONE_OTP` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPES_USER` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPES_USERNAME` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPES_USEROTP` (*fleeingform.models.FleetingAuth* attribute), 7
- `AUTH_TYPES_USERPASS` (*fleeingform.models.FleetingAuth* attribute), 7
- `auth_username_field()` (in module *fleeingform.lib*), 33
- `authenticate()` (*fleeingform.authentication.FleetingAuthentication* method), 37
- `authenticate()` (*fleeingform.models.FleetingAuth* method), 8
- `authenticate()` (*fleeingform.models.FleetingUser* method), 21
- `authentication_classes` (*fleeingform.views.FleetingFormListCreateView* attribute), 24
- `authentication_classes` (*fleeingform.views.FleetingFormRetrieveDestroyView* attribute), 25
- `authentication_classes` (*fleeingform.views.FleetingNamespaceListCreateView* attribute), 7

- attribute), 26
- authentication_classes (fleetingform.views.FleetingNamespaceRetrieveUpdateView attribute), 26
- authentication_classes (fleetingform.views.FleetingNamespaceRetrieveView attribute), 27
- ## C
- CH_URL_UNSAFE (fleetingform.models.FleetingNamespace attribute), 16
- choices (fleetingform.models.FleetingFormControl attribute), 14
- clean() (fleetingform.forms.PasswordAuthenticationForm method), 28
- clean() (fleetingform.forms.UserAuthenticationForm method), 29
- clean() (fleetingform.forms.UserPasswordAuthenticationForm method), 29
- code (fleetingform.models.FleetingAuditEntry attribute), 6
- code (fleetingform.models.FleetingForm attribute), 12
- complete() (fleetingform.models.FleetingForm method), 12
- completed() (fleetingform.models.FleetingForm property), 12
- completed_on (fleetingform.models.FleetingAuditEntry attribute), 6
- completed_on (fleetingform.models.FleetingForm attribute), 12
- content (fleetingform.models.FleetingAuth attribute), 8
- content (fleetingform.models.FleetingTemplate attribute), 19
- content_type (fleetingform.models.FleetingFormControl attribute), 15
- content_type (fleetingform.models.FleetingTemplate attribute), 19
- content_type_id (fleetingform.models.FleetingFormControl attribute), 15
- create() (fleetingform.serializers.FleetingAuthSerializer method), 29
- create() (fleetingform.serializers.FleetingFormSerializer method), 30
- created_on (fleetingform.models.FleetingAuditEntry attribute), 6
- created_on (fleetingform.models.FleetingForm attribute), 12
- ## D
- decode_query_params() (fleetingform.views.UserFormView method), 27
- delete() (fleetingform.models.FleetingAuditEntry method), 6
- delete() (fleetingform.models.FleetingNamespace method), 16
- disabled (fleetingform.models.FleetingFormControl attribute), 15
- ## E
- email (fleetingform.models.FleetingUser attribute), 21
- email_otp() (fleetingform.models.FleetingAuth property), 8
- encrypt_and_set_password() (fleetingform.models.FleetingUser method), 21
- error() (fleetingform.models.FleetingForm method), 12
- event (fleetingform.models.FleetingWebhook attribute), 23
- expired() (fleetingform.models.FleetingForm property), 12
- expires_on() (fleetingform.models.FleetingForm property), 12
- ## F
- FIELD_TYPE_BOOLEAN (fleetingform.models.FleetingFormControl attribute), 14
- FIELD_TYPE_CHOICE (fleetingform.models.FleetingFormControl attribute), 14
- FIELD_TYPE_DATE (fleetingform.models.FleetingFormControl attribute), 14
- FIELD_TYPE_DATETIME (fleetingform.models.FleetingFormControl attribute), 14
- FIELD_TYPE_DECIMAL (fleetingform.models.FleetingFormControl attribute), 14
- FIELD_TYPE_EMAIL (fleetingform.models.FleetingFormControl attribute), 14
- FIELD_TYPE_FLOAT (fleetingform.models.FleetingFormControl attribute), 14
- FIELD_TYPE_INTEGER (fleetingform.models.FleetingFormControl attribute), 14
- FIELD_TYPE_TEXT (fleetingform.models.FleetingFormControl attribute), 14

[FIELD_TYPE_TEXTAREA](#) (*fleetingform.models.FleetingFormControl* attribute), [14](#)
[FIELD_TYPE_TIME](#) (*fleetingform.models.FleetingFormControl* attribute), [14](#)
[FIELD_TYPE_URL](#) (*fleetingform.models.FleetingFormControl* attribute), [14](#)
[FIELD_TYPES](#) (*fleetingform.models.FleetingFormControl* attribute), [14](#)
[fields](#) (*fleetingform.serializers.FleetingActionSerializer.Meta* attribute), [29](#)
[fields](#) (*fleetingform.serializers.FleetingAuthSerializer.Meta* attribute), [29](#)
[fields](#) (*fleetingform.serializers.FleetingChoiceSerializer.Meta* attribute), [30](#)
[fields](#) (*fleetingform.serializers.FleetingFormControlSerializer.Meta* attribute), [30](#)
[fields](#) (*fleetingform.serializers.FleetingFormSerializer.Meta* attribute), [30](#)
[fields](#) (*fleetingform.serializers.FleetingNamespaceSerializer.Meta* attribute), [31](#)
[fields](#) (*fleetingform.serializers.FleetingTemplateSerializer.Meta* attribute), [31](#)
[fields](#) (*fleetingform.serializers.FleetingUserSerializer.Meta* attribute), [32](#)
[fields](#) (*fleetingform.serializers.FleetingValidationSerializer.Meta* attribute), [32](#)
[fields](#) (*fleetingform.serializers.FleetingWebhookSerializer.Meta* attribute), [32](#)
[FleetingAction](#) (class in *fleetingform.models*), [5](#)
[FleetingAction.DoesNotExist](#), [5](#)
[FleetingAction.MultipleObjectsReturned](#), [5](#)
[FleetingActionSerializer](#) (class in *fleetingform.serializers*), [29](#)
[FleetingActionSerializer.Meta](#) (class in *fleetingform.serializers*), [29](#)
[FleetingAuditEntry](#) (class in *fleetingform.models*), [6](#)
[FleetingAuditEntry.DoesNotExist](#), [6](#)
[FleetingAuditEntry.MultipleObjectsReturned](#), [6](#)
[FleetingAuth](#) (class in *fleetingform.models*), [7](#)
[FleetingAuth.DoesNotExist](#), [7](#)
[FleetingAuth.MultipleObjectsReturned](#), [7](#)
[FleetingAuthentication](#) (class in *fleetingform.authentication*), [37](#)
[FleetingAuthenticationForm](#) (class in *fleetingform.forms*), [28](#)
[FleetingAuthOTPErrors](#), [33](#)
[FleetingAuthSerializer](#) (class in *fleetingform.serializers*), [29](#)
[FleetingAuthSerializer.Meta](#) (class in *fleetingform.serializers*), [29](#)
[FleetingChoice](#) (class in *fleetingform.models*), [10](#)
[FleetingChoice.DoesNotExist](#), [10](#)
[FleetingChoice.MultipleObjectsReturned](#), [10](#)
[FleetingChoiceSerializer](#) (class in *fleetingform.serializers*), [30](#)
[FleetingChoiceSerializer.Meta](#) (class in *fleetingform.serializers*), [30](#)
[FleetingDeletionError](#), [33](#)
[FleetingForm](#) (class in *fleetingform.models*), [11](#)
[fleetingform.authentication](#) module, [37](#)
[FleetingForm.DoesNotExist](#), [11](#)
[fleetingform.errors](#) module, [33](#)
[fleetingform.forms](#) module, [28](#)
[fleetingform.lib](#) module, [33](#)
[fleetingform.lib.form_generator](#) module, [34](#)
[fleetingform.middleware](#) module, [38](#)
[fleetingform.models](#) module, [5](#)
[FleetingForm.MultipleObjectsReturned](#), [11](#)
[fleetingform.permissions](#) module, [37](#)
[fleetingform.serializers](#) module, [29](#)
[fleetingform.views](#) module, [24](#)
[FleetingFormCompleteError](#), [33](#)
[FleetingFormControl](#) (class in *fleetingform.models*), [14](#)
[FleetingFormControl.DoesNotExist](#), [14](#)
[FleetingFormControl.MultipleObjectsReturned](#), [14](#)
[FleetingFormControlSerializer](#) (class in *fleetingform.serializers*), [30](#)
[FleetingFormControlSerializer.Meta](#) (class in *fleetingform.serializers*), [30](#)
[FleetingFormGenerationError](#), [33](#)
[FleetingFormHardLimitPermission](#) (class in *fleetingform.permissions*), [37](#)
[FleetingFormListCreateView](#) (class in *fleetingform.views*), [24](#)
[FleetingFormLoginView](#) (class in *fleetingform.views*), [24](#)

<code>FleetingFormRetrieveDestroyView</code> (class in <code>fleetingform.views</code>), 25	<code>FleetingValidationSerializer.Meta</code> (class in <code>fleetingform.serializers</code>), 32
<code>FleetingFormSerializer</code> (class in <code>fleetingform.serializers</code>), 30	<code>FleetingWebhook</code> (class in <code>fleetingform.models</code>), 23
<code>FleetingFormSerializer.Meta</code> (class in <code>fleetingform.serializers</code>), 30	<code>FleetingWebhook.DoesNotExist</code> , 23
<code>FleetingFormTokenPermission</code> (class in <code>fleetingform.permissions</code>), 37	<code>FleetingWebhook.MultipleObjectsReturned</code> , 23
<code>FleetingNamespace</code> (class in <code>fleetingform.models</code>), 16	<code>FleetingWebhookSerializer</code> (class in <code>fleetingform.serializers</code>), 32
<code>FleetingNamespace.DoesNotExist</code> , 16	<code>FleetingWebhookSerializer.Meta</code> (class in <code>fleetingform.serializers</code>), 32
<code>FleetingNamespace.MultipleObjectsReturned</code> , 16	<code>form</code> (<code>fleetingform.models.FleetingAuth</code> attribute), 8
<code>FleetingNamespaceListCreateView</code> (class in <code>fleetingform.views</code>), 26	<code>form</code> (<code>fleetingform.models.FleetingFormControl</code> attribute), 15
<code>FleetingNamespaceMiddleware</code> (class in <code>fleetingform.middleware</code>), 38	<code>form</code> (<code>fleetingform.models.FleetingTemplate</code> attribute), 19
<code>FleetingNamespaceRetrieveUpdateView</code> (class in <code>fleetingform.views</code>), 26	<code>form_control</code> (<code>fleetingform.models.FleetingChoice</code> attribute), 10
<code>FleetingNamespaceRetrieveView</code> (class in <code>fleetingform.views</code>), 27	<code>form_control</code> (<code>fleetingform.models.FleetingValidation</code> attribute), 22
<code>FleetingNamespaceSerializer</code> (class in <code>fleetingform.serializers</code>), 31	<code>form_control_id</code> (<code>fleetingform.models.FleetingChoice</code> attribute), 10
<code>FleetingNamespaceSerializer.Meta</code> (class in <code>fleetingform.serializers</code>), 31	<code>form_control_id</code> (<code>fleetingform.models.FleetingValidation</code> attribute), 22
<code>FleetingNamespaceTokenPermission</code> (class in <code>fleetingform.permissions</code>), 37	<code>form_controls</code> (<code>fleetingform.models.FleetingAuth</code> attribute), 8
<code>FleetingOTPRestRedirectView</code> (class in <code>fleetingform.views</code>), 27	<code>form_controls</code> (<code>fleetingform.models.FleetingTemplate</code> attribute), 19
<code>FleetingOTPRetriesExceeded</code> , 33	<code>FORM_STATUS_COMPLETED</code> (<code>fleetingform.models.FleetingForm</code> attribute), 11
<code>FleetingTemplate</code> (class in <code>fleetingform.models</code>), 18	<code>FORM_STATUS_CREATED</code> (<code>fleetingform.models.FleetingForm</code> attribute), 11
<code>FleetingTemplate.DoesNotExist</code> , 18	<code>FORM_STATUS_ERROR</code> (<code>fleetingform.models.FleetingForm</code> attribute), 11
<code>FleetingTemplate.MultipleObjectsReturned</code> , 18	<code>FORM_STATUS_OPENED</code> (<code>fleetingform.models.FleetingForm</code> attribute), 11
<code>FleetingTemplateSerializer</code> (class in <code>fleetingform.serializers</code>), 31	<code>FORM_STATUSES</code> (<code>fleetingform.models.FleetingForm</code> attribute), 11
<code>FleetingTemplateSerializer.Meta</code> (class in <code>fleetingform.serializers</code>), 31	<code>FORM_STATUSES_COMPLETED</code> (<code>fleetingform.models.FleetingForm</code> attribute), 11
<code>FleetingUser</code> (class in <code>fleetingform.models</code>), 20	<code>forms</code> (<code>fleetingform.models.FleetingNamespace</code> attribute), 17
<code>FleetingUser.DoesNotExist</code> , 20	<code>forms_between()</code> (<code>fleetingform.models.FleetingNamespace</code> method), 17
<code>FleetingUser.MultipleObjectsReturned</code> , 20	<code>forms_last_month()</code> (<code>fleetingform.models.FleetingNamespace</code> property), 17
<code>FleetingUserSerializer</code> (class in <code>fleetingform.serializers</code>), 31	<code>forms_since()</code> (<code>fleetingform.models.FleetingNamespace</code> method), 17
<code>FleetingUserSerializer.Meta</code> (class in <code>fleetingform.serializers</code>), 32	<code>forms_this_month()</code> (<code>fleetingform.models.FleetingNamespace</code> method), 17
<code>FleetingValidation</code> (class in <code>fleetingform.models</code>), 22	
<code>FleetingValidation.DoesNotExist</code> , 22	
<code>FleetingValidation.MultipleObjectsReturned</code> , 22	
<code>FleetingValidationError</code> , 33	
<code>FleetingValidationSerializer</code> (class in <code>fleetingform.serializers</code>), 32	

<i>form.models.FleetingNamespace</i> 17	property),	<i>get_status_display()</i> <i>form.models.FleetingAuditEntry</i> 6	(fleeting- method),
<i>from_request()</i> <i>form.models.FleetingNamespace</i> method), 17	(fleeting- class	<i>get_status_display()</i> <i>form.models.FleetingForm</i> method), 13	(fleeting- method),
G		<i>get_template_display()</i> <i>form.models.FleetingAuditEntry</i> 6	(fleeting- method),
<i>generate_and_send_otp()</i> <i>form.models.FleetingUser</i> method), 21	(fleeting- method), 21	<i>get_type_display()</i> <i>form.models.FleetingAuth</i> method), 8	(fleeting- method),
<i>generate_fleeting_form_class_for()</i> (in module <i>fleetingform.lib.form_generator</i>), 34	(in module <i>fleetingform.lib.form_generator</i>), 34	<i>get_type_display()</i> <i>form.models.FleetingFormControl</i> 15	(fleeting- method),
<i>get()</i> (<i>fleetingform.views.FleetingFormLoginView</i> method), 25	(<i>fleetingform.views.FleetingFormLoginView</i> method), 25	<i>get_type_display()</i> <i>form.models.FleetingTemplate</i> 19	(fleeting- method),
<i>get()</i> (<i>fleetingform.views.FleetingNamespaceRetrieveUpdateView</i> method), 26	(<i>fleetingform.views.FleetingNamespaceRetrieveUpdateView</i> method), 26	<i>get_type_display()</i> <i>form.models.FleetingValidation</i> 22	(fleeting- method),
<i>get()</i> (<i>fleetingform.views.UserFormView</i> method), 27	(<i>fleetingform.views.UserFormView</i> method), 27	<i>get_url_shortener_display()</i> <i>form.models.FleetingNamespace</i> 17	(fleeting- method),
<i>get_auth_display()</i> <i>form.models.FleetingAuditEntry</i> 6	(fleeting- method), 6	<i>get_usage()</i> <i>form.serializers.FleetingNamespaceSerializer</i> method), 31	(fleeting- method), 31
<i>get_content_type_display()</i> <i>form.models.FleetingTemplate</i> 19	(fleeting- method), 19	H	
<i>get_event_display()</i> <i>form.models.FleetingWebhook</i> 23	(fleeting- method), 23	<i>hard_limit</i> (<i>fleetingform.models.FleetingNamespace</i> attribute), 17	(fleeting- attribute), 17
<i>get_next_by_created_on()</i> <i>form.models.FleetingAuditEntry</i> 6	(fleeting- method), 6	<i>has_object_permission()</i> <i>form.permissions.FleetingFormTokenPermission</i> method), 37	(fleeting- method), 37
<i>get_next_by_created_on()</i> <i>form.models.FleetingForm</i> method), 13	(fleeting- method), 13	<i>has_object_permission()</i> <i>form.permissions.FleetingNamespaceTokenPermission</i> method), 37	(fleeting- method), 37
<i>get_previous_by_created_on()</i> <i>form.models.FleetingAuditEntry</i> 6	(fleeting- method), 6	<i>has_permission()</i> <i>form.permissions.FleetingFormHardLimitPermission</i> method), 37	(fleeting- method), 37
<i>get_previous_by_created_on()</i> <i>form.models.FleetingForm</i> method), 13	(fleeting- method), 13	<i>has_permission()</i> <i>form.permissions.FleetingFormTokenPermission</i> method), 37	(fleeting- method), 37
<i>get_queryset()</i> <i>form.views.FleetingFormListCreateView</i> method), 24	(fleeting- method), 24	<i>has_permission()</i> <i>form.permissions.FleetingNamespaceTokenPermission</i> method), 37	(fleeting- method), 37
<i>get_queryset()</i> <i>form.views.FleetingFormRetrieveDestroyView</i> method), 25	(fleeting- method), 25	<i>help_text</i> (<i>fleetingform.models.FleetingFormControl</i> attribute), 15	(fleeting- attribute), 15
<i>get_queryset()</i> <i>form.views.FleetingNamespaceListCreateView</i> method), 26	(fleeting- method), 26	<i>helper()</i> (<i>fleetingform.models.FleetingTemplate</i> prop- erty), 19	(fleeting- property), 19
<i>get_queryset()</i> <i>form.views.FleetingNamespaceRetrieveUpdateView</i> method), 26	(fleeting- method), 26	<i>helper_class()</i> <i>form.models.FleetingTemplate</i> 20	(fleeting- method), 20
<i>get_queryset()</i> <i>form.views.FleetingNamespaceRetrieveView</i> method), 27	(fleeting- method), 27	<i>hidden</i> (<i>fleetingform.models.FleetingFormControl</i> at- tribute), 15	(fleeting- attribute), 15
<i>get_redirect_url()</i> <i>form.views.FleetingOTPRestResetRedirectView</i> method), 27	(fleeting- method), 27		

html_template() (fleetingform.models.FleetingTemplate attribute), 20

I

- id (fleetingform.models.FleetingAction attribute), 5
- id (fleetingform.models.FleetingAuditEntry attribute), 6
- id (fleetingform.models.FleetingAuth attribute), 8
- id (fleetingform.models.FleetingChoice attribute), 10
- id (fleetingform.models.FleetingForm attribute), 13
- id (fleetingform.models.FleetingFormControl attribute), 15
- id (fleetingform.models.FleetingNamespace attribute), 17
- id (fleetingform.models.FleetingTemplate attribute), 20
- id (fleetingform.models.FleetingUser attribute), 21
- id (fleetingform.models.FleetingValidation attribute), 22
- id (fleetingform.models.FleetingWebhook attribute), 23
- initial (fleetingform.models.FleetingFormControl attribute), 15

L

- label (fleetingform.models.FleetingAction attribute), 5
- label (fleetingform.models.FleetingFormControl attribute), 15
- logo (fleetingform.models.FleetingNamespace attribute), 17

M

- message (fleetingform.models.FleetingValidation attribute), 22
- message (fleetingform.permissions.FleetingFormHardLimitPermission attribute), 37
- message (fleetingform.permissions.FleetingFormTokenPermission attribute), 37
- message (fleetingform.permissions.FleetingNamespaceTokenPermission attribute), 37
- model (fleetingform.serializers.FleetingActionSerializer.Meta attribute), 29
- model (fleetingform.serializers.FleetingAuthSerializer.Meta attribute), 29
- model (fleetingform.serializers.FleetingChoiceSerializer.Meta attribute), 30
- model (fleetingform.serializers.FleetingFormControlSerializer.Meta attribute), 30
- model (fleetingform.serializers.FleetingFormSerializer.Meta attribute), 30
- model (fleetingform.serializers.FleetingNamespaceSerializer.Meta attribute), 31
- model (fleetingform.serializers.FleetingTemplateSerializer.Meta attribute), 31
- model (fleetingform.serializers.FleetingUserSerializer.Meta attribute), 32
- model (fleetingform.serializers.FleetingValidationSerializer.Meta attribute), 32
- model (fleetingform.serializers.FleetingWebhookSerializer.Meta attribute), 32
- module
 - fleetingform.authentication, 37
 - fleetingform.errors, 33
 - fleetingform.forms, 28
 - fleetingform.lib, 33
 - fleetingform.lib.form_generator, 34
 - fleetingform.middleware, 38
 - fleetingform.models, 5
 - fleetingform.permissions, 37
 - fleetingform.serializers, 29
 - fleetingform.views, 24

N

- name (fleetingform.models.FleetingFormControl attribute), 15
- name (fleetingform.models.FleetingWebhook attribute), 23
- namespace (fleetingform.models.FleetingAuditEntry attribute), 6
- namespace (fleetingform.models.FleetingForm attribute), 13
- namespace (fleetingform.models.FleetingWebhook attribute), 23
- namespace_id (fleetingform.models.FleetingAuditEntry attribute), 6
- namespace_id (fleetingform.models.FleetingForm attribute), 13
- namespace_id (fleetingform.models.FleetingWebhook attribute), 23

O

- object_id (fleetingform.models.FleetingFormControl attribute), 15
- objects (fleetingform.models.FleetingAction attribute), 5
- objects (fleetingform.models.FleetingAuditEntry attribute), 6
- objects (fleetingform.models.FleetingAuth attribute), 8
- objects (fleetingform.models.FleetingChoice attribute), 10
- objects (fleetingform.models.FleetingForm attribute), 13
- objects (fleetingform.models.FleetingFormControl attribute), 15
- objects (fleetingform.models.FleetingNamespace attribute), 17
- objects (fleetingform.models.FleetingTemplate attribute), 20

- objects (*fleetingform.models.FleetingUser* attribute), 21
- objects (*fleetingform.models.FleetingValidation* attribute), 23
- objects (*fleetingform.models.FleetingWebhook* attribute), 23
- open() (*fleetingform.models.FleetingForm* method), 13
- opened_by() (*fleetingform.models.FleetingAuth* method), 8
- opened_on (*fleetingform.models.FleetingAuditEntry* attribute), 6
- opened_on (*fleetingform.models.FleetingForm* attribute), 13
- opened_on (*fleetingform.models.FleetingUser* attribute), 21
- otp() (*fleetingform.models.FleetingAuth* property), 8
- otp_contact_obscured() (*fleetingform.models.FleetingUser* property), 21
- otp_instructions (*fleetingform.views.FleetingFormLoginView* attribute), 25
- ## P
- params (*fleetingform.models.FleetingFormControl* attribute), 15
- params (*fleetingform.models.FleetingTemplate* attribute), 20
- params (*fleetingform.models.FleetingValidation* attribute), 23
- password (*fleetingform.models.FleetingUser* attribute), 21
- password_instructions (*fleetingform.views.FleetingFormLoginView* attribute), 25
- password_only() (*fleetingform.models.FleetingAuth* property), 9
- PasswordAuthenticationForm (class in *fleetingform.forms*), 28
- pattern_name (*fleetingform.views.FleetingOTPRestRedirectView* attribute), 27
- perform_create() (*fleetingform.views.FleetingFormListView* method), 24
- perform_create() (*fleetingform.views.FleetingNamespaceListView* method), 26
- permanent (*fleetingform.views.FleetingOTPRestRedirectView* attribute), 27
- permission_classes (*fleetingform.views.FleetingFormListView* attribute), 24
- permission_classes (*fleetingform.views.FleetingFormRetrieveDestroyView* attribute), 26
- permission_classes (*fleetingform.views.FleetingNamespaceListView* attribute), 26
- permission_classes (*fleetingform.views.FleetingNamespaceRetrieveUpdateView* attribute), 26
- permission_classes (*fleetingform.views.FleetingNamespaceRetrieveView* attribute), 27
- phone (*fleetingform.models.FleetingUser* attribute), 21
- phone_otp() (*fleetingform.models.FleetingAuth* property), 9
- post() (*fleetingform.views.FleetingFormLoginView* method), 25
- post() (*fleetingform.views.UserFormView* method), 28
- put() (*fleetingform.views.FleetingNamespaceRetrieveUpdateView* method), 26
- ## R
- RE_URL_UNSAFE (*fleetingform.models.FleetingNamespace* attribute), 16
- required (*fleetingform.models.FleetingFormControl* attribute), 15
- required() (*fleetingform.models.FleetingAuth* property), 9
- REQUIRED_FORM_CONTROL_DEFAULTS (*fleetingform.serializers.FleetingAuthSerializer* attribute), 29
- REQUIRED_FORM_CONTROLS (*fleetingform.serializers.FleetingAuthSerializer* attribute), 29
- requires_password() (*fleetingform.models.FleetingAuth* property), 9
- requires_username() (*fleetingform.models.FleetingAuth* property), 9
- result (*fleetingform.models.FleetingForm* attribute), 13
- retention (*fleetingform.models.FleetingNamespace* attribute), 17
- ## S
- save() (*fleetingform.models.FleetingForm* method), 13
- save() (*fleetingform.models.FleetingNamespace* method), 17
- send_otp_and_set_messages() (in module *fleetingform.views*), 28
- send_otp_email() (in module *fleetingform.lib*), 33
- send_otp_sms() (in module *fleetingform.lib*), 34
- serializer_class (*fleetingform.views.FleetingFormListView* attribute), 24

serializer_class (fleetingform.views.FleetingFormRetrieveDestroyView attribute), 26	template_id (fleetingform.models.FleetingAction attribute), 5
serializer_class (fleetingform.views.FleetingNamespaceListCreateView attribute), 26	template_id (fleetingform.models.FleetingForm attribute), 14
serializer_class (fleetingform.views.FleetingNamespaceRetrieveUpdateView attribute), 26	TEMPLATE_TYPE_GENERIC (fleetingform.models.FleetingTemplate attribute), 19
serializer_class (fleetingform.views.FleetingNamespaceRetrieveView attribute), 27	TEMPLATE_TYPE_PAYPAL (fleetingform.models.FleetingTemplate attribute), 19
short_url() (fleetingform.models.FleetingForm property), 13	TEMPLATE_TYPE_STRIPE (fleetingform.models.FleetingTemplate attribute), 19
soft_limit (fleetingform.models.FleetingNamespace attribute), 17	TEMPLATE_TYPES (fleetingform.models.FleetingTemplate attribute), 19
status (fleetingform.models.FleetingAuditEntry attribute), 6	text (fleetingform.models.FleetingChoice attribute), 11
status (fleetingform.models.FleetingForm attribute), 13	title (fleetingform.models.FleetingAuth attribute), 9
style (fleetingform.models.FleetingNamespace attribute), 17	title (fleetingform.models.FleetingTemplate attribute), 20
subdomain (fleetingform.models.FleetingNamespace attribute), 17	to_representation() (fleetingform.serializers.FleetingAuthSerializer method), 29
support_email (fleetingform.models.FleetingNamespace attribute), 18	to_representation() (fleetingform.serializers.FleetingFormControlSerializer method), 30
supported_templates() (fleetingform.models.FleetingTemplate property), 20	to_representation() (fleetingform.serializers.FleetingNamespaceSerializer method), 31
	to_representation() (fleetingform.serializers.FleetingUserSerializer method), 32
T	token (fleetingform.models.FleetingNamespace attribute), 18
template (fleetingform.models.FleetingAction attribute), 5	token (fleetingform.models.FleetingWebhook attribute), 23
template (fleetingform.models.FleetingAuditEntry attribute), 7	total_forms() (fleetingform.models.FleetingNamespace property), 18
template (fleetingform.models.FleetingForm attribute), 13	total_forms_between() (fleetingform.models.FleetingNamespace method), 18
TEMPLATE_CONTENT_TYPE_MARKDOWN (fleetingform.models.FleetingTemplate attribute), 19	total_forms_since() (fleetingform.models.FleetingNamespace method), 18
TEMPLATE_CONTENT_TYPE_PLAINTEXT (fleetingform.models.FleetingTemplate attribute), 19	type (fleetingform.models.FleetingAuth attribute), 9
TEMPLATE_CONTENT_TYPE_RESTRUCTURED (fleetingform.models.FleetingTemplate attribute), 19	type (fleetingform.models.FleetingFormControl attribute), 16
TEMPLATE_CONTENT_TYPES (fleetingform.models.FleetingTemplate attribute), 19	type (fleetingform.models.FleetingTemplate attribute), 20
template_helper_class_for() (fleetingform.models.FleetingTemplate class method), 20	type (fleetingform.models.FleetingValidation attribute), 23
TEMPLATE_HELPERS (fleetingform.models.FleetingTemplate attribute), 19	

U

[url](#) (*fleetingform.models.FleetingWebhook* attribute), 24
[url\(\)](#) (*fleetingform.models.FleetingForm* property), 14
[URL_FLEETING](#) (*fleetingform.models.FleetingNamespace* attribute), 16
[URL_FLEETING_LINK](#) (*fleetingform.models.FleetingNamespace* attribute), 16
[url_shortener](#) (*fleetingform.models.FleetingNamespace* attribute), 18
[URL_SHORTENERS](#) (*fleetingform.models.FleetingNamespace* attribute), 16
[user](#) (*fleetingform.models.FleetingNamespace* attribute), 18
[user_id](#) (*fleetingform.models.FleetingNamespace* attribute), 18
[UserAuthenticationForm](#) (class in *fleetingform.forms*), 28
[UserFormView](#) (class in *fleetingform.views*), 27
[username](#) (*fleetingform.models.FleetingUser* attribute), 22
[username_and_otp\(\)](#) (*fleetingform.models.FleetingAuth* property), 9
[username_and_password\(\)](#) (*fleetingform.models.FleetingAuth* property), 9
[username_instructions](#) (*fleetingform.views.FleetingFormLoginView* attribute), 25
[username_only\(\)](#) (*fleetingform.models.FleetingAuth* property), 9
[userpass_instructions](#) (*fleetingform.views.FleetingFormLoginView* attribute), 25
[UserPasswordAuthenticationForm](#) (class in *fleetingform.forms*), 29
[users](#) (*fleetingform.models.FleetingAuth* attribute), 10

V

[validate\(\)](#) (*fleetingform.serializers.FleetingAuthSerializer* method), 30
[validate\(\)](#) (*fleetingform.serializers.FleetingFormControlSerializer* method), 30
[validate\(\)](#) (*fleetingform.serializers.FleetingTemplateSerializer* method), 31
[validate_actions\(\)](#) (*fleetingform.serializers.FleetingTemplateSerializer* method), 31

[validate_app\(\)](#) (*fleetingform.serializers.FleetingFormSerializer* method), 31
[validate_auth\(\)](#) (*fleetingform.serializers.FleetingFormSerializer* method), 31
[validate_content\(\)](#) (*fleetingform.serializers.FleetingTemplateSerializer* method), 31
[validate_namespace\(\)](#) (*fleetingform.serializers.FleetingFormSerializer* method), 31
[validate_password\(\)](#) (*fleetingform.serializers.FleetingUserSerializer* method), 32
[validate_phone\(\)](#) (*fleetingform.serializers.FleetingUserSerializer* method), 32
[validate_title\(\)](#) (*fleetingform.serializers.FleetingTemplateSerializer* method), 31
[VALIDATION_TYPE_MAX](#) (*fleetingform.models.FleetingValidation* attribute), 22
[VALIDATION_TYPE_MAX_LENGTH](#) (*fleetingform.models.FleetingValidation* attribute), 22
[VALIDATION_TYPE_MIN](#) (*fleetingform.models.FleetingValidation* attribute), 22
[VALIDATION_TYPE_MIN_LENGTH](#) (*fleetingform.models.FleetingValidation* attribute), 22
[VALIDATION_TYPE_RANGE](#) (*fleetingform.models.FleetingValidation* attribute), 22
[VALIDATION_TYPE_RANGE_LENGTH](#) (*fleetingform.models.FleetingValidation* attribute), 22
[VALIDATION_TYPE_REGEX](#) (*fleetingform.models.FleetingValidation* attribute), 22
[VALIDATION_TYPES](#) (*fleetingform.models.FleetingValidation* attribute), 22
[validations](#) (*fleetingform.models.FleetingFormControl* attribute), 16
[value](#) (*fleetingform.models.FleetingChoice* attribute), 11
[verify_password\(\)](#) (*fleetingform.models.FleetingAuth* method), 10
[verify_username\(\)](#) (*fleetingform.models.FleetingAuth* method), 10

W

WEBHOOK_EVENT_ACCESS	(<i>fleeting-</i> <i>form.models.FleetingWebhook</i> 23	(<i>fleeting-</i> <i>attribute</i>),
WEBHOOK_EVENT_COMPLETE	(<i>fleeting-</i> <i>form.models.FleetingWebhook</i> 23	(<i>fleeting-</i> <i>attribute</i>),
WEBHOOK_EVENT_CREATE	(<i>fleeting-</i> <i>form.models.FleetingWebhook</i> 23	(<i>fleeting-</i> <i>attribute</i>),
WEBHOOK_EVENT_ERROR	(<i>fleeting-</i> <i>form.models.FleetingWebhook</i> 23	(<i>fleeting-</i> <i>attribute</i>),
WEBHOOK_EVENTS	(<i>fleeting-</i> <i>form.models.FleetingWebhook</i> 23	(<i>fleeting-</i> <i>attribute</i>),
webhooks	(<i>fleetingform.models.FleetingNamespace</i> <i>at-</i> <i>tribute</i>), 18	