
Tarrab.me Server Documentation

Release 0.1.0a

Alfred Santacatalina

February 19, 2015

1	Commons module	3
1.1	Commons Models	3
1.2	Commons AdminForms	4
1.3	Commons Admin	4
2	Events module	5
2.1	Events Models	5
2.2	Events AdminForms	8
2.3	Events Admin	8
3	Tickets module	9
3.1	Tickets Models	9
3.2	Tickets AdminForms	17
3.3	Tickets Admin	18
3.4	Tickets Exporters	19
3.5	Tickets Importers	20
3.6	Tickets Helpers	20
4	Users module	21
4.1	Users Models	21
5	Indices and tables	23
	Python Module Index	25

Contents:

Commons module

1.1 Commons Models

```
class commons.models.CommonModel(*args, **kwargs)
```

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

creation_date

Creation datetime

Type: `django.db.models.fields.DateTimeField`

Sets now when create.

Date (with time)

modified_date

Modified datetime

Type: `django.db.models.fields.DateTimeField`

Sets now on each updates.

Date (with time)

created_by

Created by

Type: `django.db.models.fields.related.ForeignKey`

Related to: `django.contrib.auth.models.User`

Foreign key (type determined by related field)

modified_by

Modified by

Type: `django.db.models.fields.related.ForeignKey`

Related to: `django.contrib.auth.models.User`

Foreign key (type determined by related field)

class Meta

Bases: `builtins.object`

abstract = `False`

`CommonModel.created_by`

`CommonModel.get_next_by_creation_date(*moreargs, **morekwargs)`

`CommonModel.get_next_by_modified_date(*moreargs, **morekwargs)`

`CommonModel.get_previous_by_creation_date(*moreargs, **morekwargs)`

`CommonModel.get_previous_by_modified_date(*moreargs, **morekwargs)`

`CommonModel.modified_by`

1.2 Commons AdminForms

```
class commons.adminForms.CommonAdminForm(data=None, files=None, auto_id='id_%s', prefix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=None, empty_permitted=False, instance=None)
```

Bases: `django.forms.models.ModelForm`

class Meta

Bases: `builtins.object`

exclude = `('created_by', 'modified_by')`

`CommonAdminForm.base_fields` = `OrderedDict()`

`CommonAdminForm.declared_fields` = `OrderedDict()`

`CommonAdminForm.media`

1.3 Commons Admin

```
class commons.admin.CommonAdmin(model, admin_site)
```

Bases: `django.contrib.admin.options.ModelAdmin`

class Meta

Bases: `builtins.object`

exclude = `('created_by', 'modified_by')`

`CommonAdmin.media`

`CommonAdmin.save_model(request, obj, form, change)`

Events module

2.1 Events Models

```
class events.models.Event(*args, **kwargs)
```

```
    Bases: commons.models.CommonModel
```

```
    Event(id, creation_date, modified_date, created_by_id, modified_by_id, name, commercial_name, prefix,
          start_date, end_date, active, description)
```

```
    id
```

```
        Id
```

```
    Type: django.db.models.fields.AutoField
```

```
    Integer
```

```
    creation_date
```

```
        Creation datetime
```

```
    Type: django.db.models.fields.DateTimeField
```

```
    Sets now when create.
```

```
    Date (with time)
```

```
    modified_date
```

```
        Modified datetime
```

```
    Type: django.db.models.fields.DateTimeField
```

```
    Sets now on each updates.
```

```
    Date (with time)
```

```
    created_by
```

```
        Created by
```

```
    Type: django.db.models.fields.related.ForeignKey
```

Related to: `django.contrib.auth.models.User`

Foreign key (type determined by related field)

modified_by

Modified by

Type: `django.db.models.fields.related.ForeignKey`

Related to: `django.contrib.auth.models.User`

Foreign key (type determined by related field)

name

Name

Type: `django.db.models.fields.CharField`

Maximum length: 100

String (up to 100)

commercial_name

Commercial name

Type: `django.db.models.fields.CharField`

Maximum length: 150

String (up to 150)

prefix

Prefix

Type: `django.db.models.fields.CharField`

Maximum length: 4

String (up to 4)

start_date

Start datetime

Type: `django.db.models.fields.DateTimeField`

Date (with time)

end_date

End datetime

Type: `django.db.models.fields.DateTimeField`

Date (with time)

active

Active

Type: `django.db.models.fields.BooleanField`

Default value: False

Boolean (either true or false)

description

Description

Type: `django.db.models.fields.TextField`

Default value:

Text

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception Event.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

Event.created_by

`Event.get_next_by_creation_date(*moreargs, **morekwargs)`

`Event.get_next_by_end_date(*moreargs, **morekwargs)`

`Event.get_next_by_modified_date(*moreargs, **morekwargs)`

`Event.get_next_by_start_date(*moreargs, **morekwargs)`

`Event.get_previous_by_creation_date(*moreargs, **morekwargs)`

`Event.get_previous_by_end_date(*moreargs, **morekwargs)`

`Event.get_previous_by_modified_date(*moreargs, **morekwargs)`

`Event.get_previous_by_start_date(*moreargs, **morekwargs)`

Event.modified_by

`Event.objects = <django.db.models.manager.Manager object at 0x7f6c04a6b588>`

class `events.models.EventSerializer` (*instance=None, data=None, **kwargs*)

Bases: `rest_framework.serializers.ModelSerializer`

class Meta

Bases: `builtins.object`

model

alias of `Event`

2.2 Events AdminForms

```
class events.adminForms.EventAdminForm(data=None, files=None, auto_id='id_%s', pre-
                                     fix=None, initial=None, error_class=<class
                                     'django.forms.utils.ErrorList'>, label_suffix=None,
                                     empty_permitted=False, instance=None)
    Bases: commons.adminForms.CommonAdminForm

    class Meta
        Bases: commons.adminForms.Meta

        model
            alias of Event

    EventAdminForm.base_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f6c04991860>)]
    EventAdminForm.declared_fields = OrderedDict()
    EventAdminForm.media
```

2.3 Events Admin

```
class events.admin.EventAdmin(model, admin_site)
    Bases: commons.admin.CommonAdmin

    form
        alias of EventAdminForm

    list_display = ('name', 'start_date', 'active', 'created_by')
    list_filter = ['active', 'start_date', 'created_by']
    media

    search_fields = ['name', 'commercial_name']
```

Tickets module

3.1 Tickets Models

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

    date

        Attempt datetime

        Type: django.db.models.fields.DateTimeField

        Sets now when create.

        Date (with time)

    user

        User

        Type: django.db.models.fields.related.ForeignKey

        Related to: django.contrib.auth.models.User

        Foreign key (type determined by related field)

class Meta
    Bases: builtins.object

    abstract = False

    AbstractAttempt.get_next_by_date (*moreargs, **morekwargs)
    AbstractAttempt.get_previous_by_date (*moreargs, **morekwargs)
    AbstractAttempt.save (force_insert=False, force_update=False, using=None, request=None)
    AbstractAttempt.user

class tickets.models.Attempt (*args, **kwargs)
    Bases: tickets.models.AbstractAttempt
    Attempt(id, date, user_id, code, ticket_code_id, success)

id
```

Id

Type: `django.db.models.fields.AutoField`

Integer

date

Attempt datetime

Type: `django.db.models.fields.DateTimeField`

Sets now when create.

Date (with time)

user

User

Type: `django.db.models.fields.related.ForeignKey`

Related to: `django.contrib.auth.models.User`

Foreign key (type determined by related field)

code

Code

Type: `django.db.models.fields.CharField`

Maximum length: 60

String (up to 60)

ticket_code

Ticket code

Type: `django.db.models.fields.related.ForeignKey`

Related to: `tickets.models.TicketCode`

Foreign key (type determined by related field)

success

Using success

Type: `django.db.models.fields.BooleanField`

Default value: False

Boolean (either true or false)

```
exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception Attempt.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

Attempt.get_next_by_date(*moreargs, **morekwargs)

Attempt.get_previous_by_date(*moreargs, **morekwargs)

Attempt.objects = <django.db.models.manager.Manager object at 0x7f6c0495f390>

Attempt.ticket_code

Attempt.user

class tickets.models.AttemptListItemSerializer(instance=None, data=None, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

class Meta
    Bases: builtins.object

    fields = ('id', 'user', 'success', 'date')

    model
        alias of Attempt

class tickets.models.AttemptSerializer(instance=None, data=None, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

class Meta
    Bases: builtins.object

    model
        alias of Attempt

class tickets.models.FakeAttempt(*args, **kwargs)
    Bases: tickets.models.AbstractAttempt

    FakeAttempt(id, date, user_id, code)

id

    Id

    Type: django.db.models.fields.AutoField

    Integer

date

    Attempt datetime

    Type: django.db.models.fields.DateTimeField

    Sets now when create.

    Date (with time)

user
```

User

Type: `django.db.models.fields.related.ForeignKey`

Related to: `django.contrib.auth.models.User`

Foreign key (type determined by related field)

code

Code

Type: `django.db.models.fields.CharField`

Maximum length: 90

String (up to 90)

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception FakeAttempt.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`FakeAttempt.get_next_by_date(*moreargs, **morekwargs)`

`FakeAttempt.get_previous_by_date(*moreargs, **morekwargs)`

`FakeAttempt.objects = <django.db.models.manager.Manager object at 0x7f6c0495f710>`

`FakeAttempt.user`

class `tickets.models.TicketCode(*args, **kwargs)`

Bases: `commons.models.CommonModel`

`TicketCode(id, creation_date, modified_date, created_by_id, modified_by_id, code, ticket_pack_id, status, ticket_number, external_id, external_customer_name, external_fiscal_number, external_locator)`

id

Id

Type: `django.db.models.fields.AutoField`

Integer

creation_date

Creation datetime

Type: `django.db.models.fields.DateTimeField`

Sets now when create.

Date (with time)

modified_date

Modified datetime

Type: `django.db.models.fields.DateTimeField`

Sets now on each updates.

Date (with time)

created_by

Created by

Type: `django.db.models.fields.related.ForeignKey`

Related to: `django.contrib.auth.models.User`

Foreign key (type determined by related field)

modified_by

Modified by

Type: `django.db.models.fields.related.ForeignKey`

Related to: `django.contrib.auth.models.User`

Foreign key (type determined by related field)

code

Code

Type: `django.db.models.fields.CharField`

Maximum length: 60

String (up to 60)

ticket_pack

Ticket pack

Type: `django.db.models.fields.related.ForeignKey`

Related to: `tickets.models.TicketPack`

Foreign key (type determined by related field)

status

Status

Type: `django.db.models.fields.CharField`

Maximum length: 2

Choices:

- NW -> New
- US -> Used
- DS -> Disabled

Default value: NW

String (up to 2)

ticket_number

Ticket number

Type: `django.db.models.fields.PositiveIntegerField`

Positive integer

external_id

External id

Type: `django.db.models.fields.CharField`

Maximum length: 50

String (up to 50)

external_customer_name

External customer name

Type: `django.db.models.fields.CharField`

Maximum length: 200

String (up to 200)

external_fiscal_number

External fiscal number

Type: `django.db.models.fields.CharField`

Maximum length: 20

String (up to 20)

external_locator

External locator

Type: `django.db.models.fields.CharField`

Maximum length: 40

String (up to 40)

```
exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception TicketCode.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

TicketCode.TICKET_CODE_STATUS = (('NW', 'New'), ('US', 'Used'), ('DS', 'Disabled'))

TicketCode.TICKET_CODE_STATUS_DISABLED = 'DS'

TicketCode.TICKET_CODE_STATUS_NEW = 'NW'

TicketCode.TICKET_CODE_STATUS_USED = 'US'

TicketCode.attempt_count()

TicketCode.attempt_list

TicketCode.created_by

TicketCode.event()

TicketCode.get_next_by_creation_date(*moreargs, **morekwargs)

TicketCode.get_next_by_modified_date(*moreargs, **morekwargs)

TicketCode.get_previous_by_creation_date(*moreargs, **morekwargs)

TicketCode.get_previous_by_modified_date(*moreargs, **morekwargs)

TicketCode.get_status_display(*moreargs, **morekwargs)

TicketCode.make_attempt(user)

TicketCode.modified_by

TicketCode.objects = <tickets.models.TicketCodeManager object at 0x7f6c049416d8>

TicketCode.ticket_pack

class tickets.models.TicketCodeManager
    Bases: django.db.models.manager.Manager

    get_for_attempt(code)

    make_attempt(code, user)

class tickets.models.TicketCodeSerializer(instance=None, data=None, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    class Meta
        Bases: builtins.object

        model
            alias of TicketCode

class tickets.models.TicketPack(*args, **kwargs)
    Bases: commons.models.CommonModel

    TicketPack(id, creation_date, modified_date, created_by_id, modified_by_id, name, prefix, event_id)

    id

        Id

        Type: django.db.models.fields.AutoField

        Integer
```

creation_date

Creation datetime

Type: `django.db.models.fields.DateTimeField`

Sets now when create.

Date (with time)

modified_date

Modified datetime

Type: `django.db.models.fields.DateTimeField`

Sets now on each updates.

Date (with time)

created_by

Created by

Type: `django.db.models.fields.related.ForeignKey`

Related to: `django.contrib.auth.models.User`

Foreign key (type determined by related field)

modified_by

Modified by

Type: `django.db.models.fields.related.ForeignKey`

Related to: `django.contrib.auth.models.User`

Foreign key (type determined by related field)

name

Name

Type: `django.db.models.fields.CharField`

Maximum length: 50

String (up to 50)

prefix

Prefix

Type: `django.db.models.fields.CharField`

Maximum length: 2

String (up to 2)

event

Event

Type: `django.db.models.fields.related.ForeignKey`

Related to: `events.models.Event`

Foreign key (type determined by related field)

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception TicketPack.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`TicketPack.created_by`

`TicketPack.event`

`TicketPack.get_next_by_creation_date(*moreargs, **morekwargs)`

`TicketPack.get_next_by_modified_date(*moreargs, **morekwargs)`

`TicketPack.get_prefix()`

`TicketPack.get_previous_by_creation_date(*moreargs, **morekwargs)`

`TicketPack.get_previous_by_modified_date(*moreargs, **morekwargs)`

`TicketPack.get_ticket_count()`

`TicketPack.modified_by`

`TicketPack.objects = <django.db.models.manager.Manager object at 0x7f6c04941080>`

`TicketPack.ticket_codes`

class `tickets.models.TicketPackSerializer` (*instance=None, data=None, **kwargs*)

Bases: `rest_framework.serializers.ModelSerializer`

class Meta

Bases: `builtins.object`

model

alias of `TicketPack`

3.2 Tickets AdminForms

```
class tickets.adminForms.TicketPackAdminForm (data=None, files=None,
auto_id='id_%s', prefix=None, initial=None, error_class=<class
'django.forms.utils.ErrorList'>, label_suffix=None, empty_permitted=False,
instance=None)
```

Bases: `commons.adminForms.CommonAdminForm`

class Meta

Bases: `commons.adminForms.Meta`

```
    model
        alias of TicketPack

TicketPackAdminForm.base_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f6c049e...
TicketPackAdminForm.declared_fields = OrderedDict([('upload_file_field', <django.forms.fields.FileField obje...
TicketPackAdminForm.media
```

3.3 Tickets Admin

```
class tickets.admin.TicketCodeAdmin(model, admin_site)
    Bases: commons.admin.CommonAdmin

    actions = ['disable', 'remove_attempts', 'mark_as_used', 'mark_as_new', 'make_attempt', <tickets.exporters.DefaultT...
    disable(request, queryset)

    form
        alias of TicketPackAdminForm

    list_display = ['code', 'ticket_number', 'status', 'ticket_pack', 'attempt_count', 'event', 'external_customer_name',
    list_filter = ['ticket_pack__event', 'status', 'ticket_pack']

    list_per_page = 500

    make_attempt(request, queryset)

    mark_as_new(request, queryset)

    mark_as_used(request, queryset)

    media

    remove_attempts(request, queryset)

    search_fields = ['code', 'ticket_number', 'external_locator', 'external_id']

class tickets.admin.TicketPackAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    actions = ['put_ticket_number', <tickets.exporters.DefaultTicketPackWithAttemptsExporter object at 0x7f6c0493e240...
    fieldsets = ((None, {'fields': ('name', 'event', 'prefix')}), ('Generate ticket codes', {'fields': ('generation_count', 'code...
    form
        alias of TicketPackAdminForm

    generate_codes(ticket_pack, request, count, length=14)

    list_display = ['__str__', 'event', 'get_ticket_count']

    list_filter = ['event']

    media

    process_upload_field(ticket_pack, request, up_file, importer)

    put_ticket_number(request, queryset)

    save_model(request, obj, form, change)
```

3.4 Tickets Exporters

```
class tickets.exporters.BaseExporter
    Bases: builtins.object

    create_response()

    get_tickets(queryset)

    mime_type = 'text/csv'

    short_description = 'Base class for exporters'

    write_data(response, tickets)

class tickets.exporters.BaseTicketExporter
    Bases: tickets.exporters.BaseExporter

    get_tickets(queryset)

    short_description = 'Base class for ticket exporters'

class tickets.exporters.BaseTicketPackExporter
    Bases: tickets.exporters.BaseExporter

    get_tickets(queryset)

    short_description = 'Base class for ticket pack exporters'

class tickets.exporters.DefaultTicketExporter
    Bases: tickets.exporters.BaseTicketExporter

    create_writer(response)

    short_description = 'Export selected tickets as CSV file'

    write_data(response, tickets)

    write_item(writer, item)

class tickets.exporters.DefaultTicketPackExporter
    Bases: tickets.exporters.BaseTicketPackExporter, tickets.exporters.DefaultTicketExporter

    short_description = 'Export selected ticket packs as CSV file'

class tickets.exporters.DefaultTicketPackWithAttemptsExporter
    Bases: tickets.exporters.BaseTicketPackExporter, tickets.exporters.DefaultTicketWithAttemptsExporter

    short_description = 'Export selected ticket packs with attempts as CSV file'

class tickets.exporters.DefaultTicketWithAttemptsExporter
    Bases: tickets.exporters.DefaultTicketExporter

    short_description = 'Export selected tickets with attempts as CSV file'

    write_item(writer, item)

class tickets.exporters.ExporterManager
    Bases: builtins.object

    register(exporter_cls)

    unregister(exporter_cls)
```

3.5 Tickets Importers

```
class tickets.importers.AtrapaloImporter (csv_file=None)
    Bases: tickets.importers.BaseImporter

    label = 'Atrapalo'

    map_to_ticket (ticket, row)

class tickets.importers.BaseImporter (csv_file=None)
    Bases: builtins.object

    create_reader (csv_file)

    iter_tickets ()

    label = 'Base importer'

    map_to_ticket (ticket, row)

class tickets.importers.DefaultImporter (csv_file=None)
    Bases: tickets.importers.BaseImporter

    label = 'Tarrab.me'

class tickets.importers.ImporterManager
    Bases: builtins.object

    create_importer (name, csv_file)

    get_select_options ()

    register (name, importer_cls)

    unregister (name)

class tickets.importers.ProductesDeLaTerraImporter (csv_file=None)
    Bases: tickets.importers.BaseImporter

    label = 'Productes de la terra'

    map_to_ticket (ticket, row)

class tickets.importers.TicketeaImporter (csv_file=None)
    Bases: tickets.importers.BaseImporter

    label = 'Ticketea'

    map_to_ticket (ticket, row)
```

3.6 Tickets Helpers

```
tickets.helpers.create_code (length=18, prefix='')

tickets.helpers.luhn_digit (code)
```

Users module

4.1 Users Models

```
class users.models.UserSerializer (instance=None, data=None, **kwargs)
    Bases: rest_framework.serializers.HyperlinkedModelSerializer

    class Meta
        Bases: builtins.object
        fields = ('id', 'username', 'first_name')
        model
            alias of User
```

Indices and tables

- *genindex*
- *modindex*
- *search*

c

`commons.admin`, 4
`commons.adminForms`, 4
`commons.models`, 3

e

`events.admin`, 8
`events.adminForms`, 8
`events.models`, 5

t

`tickets.admin`, 18
`tickets.adminForms`, 17
`tickets.exporters`, 19
`tickets.helpers`, 20
`tickets.importers`, 20
`tickets.models`, 9

u

`users.models`, 21

A

abstract (commons.models.CommonModel.Meta attribute), 4
 abstract (tickets.models.AbstractAttempt.Meta attribute), 9
 AbstractAttempt (class in tickets.models), 9
 AbstractAttempt.Meta (class in tickets.models), 9
 actions (tickets.admin.TicketCodeAdmin attribute), 18
 actions (tickets.admin.TicketPackAdmin attribute), 18
 active (events.models.Event attribute), 7
 AtrapaloImporter (class in tickets.importers), 20
 Attempt (class in tickets.models), 9
 Attempt.DoesNotExist, 10
 Attempt.MultipleObjectsReturned, 11
 attempt_count() (tickets.models.TicketCode method), 15
 attempt_list (tickets.models.TicketCode attribute), 15
 AttemptListItemSerializer (class in tickets.models), 11
 AttemptListItemSerializer.Meta (class in tickets.models), 11
 AttemptSerializer (class in tickets.models), 11
 AttemptSerializer.Meta (class in tickets.models), 11

B

base_fields (commons.adminForms.CommonAdminForm attribute), 4
 base_fields (events.adminForms.EventAdminForm attribute), 8
 base_fields (tickets.adminForms.TicketPackAdminForm attribute), 18
 BaseExporter (class in tickets.exporters), 19
 BaseImporter (class in tickets.importers), 20
 BaseTicketExporter (class in tickets.exporters), 19
 BaseTicketPackExporter (class in tickets.exporters), 19

C

code (tickets.models.Attempt attribute), 10
 code (tickets.models.FakeAttempt attribute), 12
 code (tickets.models.TicketCode attribute), 13
 commercial_name (events.models.Event attribute), 6
 CommonAdmin (class in commons.admin), 4

CommonAdmin.Meta (class in commons.admin), 4
 CommonAdminForm (class in commons.adminForms), 4
 CommonAdminForm.Meta (class in commons.adminForms), 4
 CommonModel (class in commons.models), 3
 CommonModel.Meta (class in commons.models), 4
 commons.admin (module), 4
 commons.adminForms (module), 4
 commons.models (module), 3
 create_code() (in module tickets.helpers), 20
 create_importer() (tickets.importers.ImporterManager method), 20
 create_reader() (tickets.importers.BaseImporter method), 20
 create_response() (tickets.exporters.BaseExporter method), 19
 create_writer() (tickets.exporters.DefaultTicketExporter method), 19
 created_by (commons.models.CommonModel attribute), 3, 4
 created_by (events.models.Event attribute), 5, 7
 created_by (tickets.models.TicketCode attribute), 13, 15
 created_by (tickets.models.TicketPack attribute), 16, 17
 creation_date (commons.models.CommonModel attribute), 3
 creation_date (events.models.Event attribute), 5
 creation_date (tickets.models.TicketCode attribute), 12
 creation_date (tickets.models.TicketPack attribute), 16

D

date (tickets.models.AbstractAttempt attribute), 9
 date (tickets.models.Attempt attribute), 10
 date (tickets.models.FakeAttempt attribute), 11
 declared_fields (commons.adminForms.CommonAdminForm attribute), 4
 declared_fields (events.adminForms.EventAdminForm attribute), 8
 declared_fields (tickets.adminForms.TicketPackAdminForm attribute), 18
 DefaultImporter (class in tickets.importers), 20
 DefaultTicketExporter (class in tickets.exporters), 19

DefaultTicketPackExporter (class in tickets.exporters), 19
DefaultTicketPackWithAttemptsExporter (class in tickets.exporters), 19
DefaultTicketWithAttemptsExporter (class in tickets.exporters), 19
description (events.models.Event attribute), 7
disable() (tickets.admin.TicketCodeAdmin method), 18

E

end_date (events.models.Event attribute), 6
Event (class in events.models), 5
event (tickets.models.TicketPack attribute), 17
event() (tickets.models.TicketCode method), 15
Event.DoesNotExist, 7
Event.MultipleObjectsReturned, 7
EventAdmin (class in events.admin), 8
EventAdminForm (class in events.adminForms), 8
EventAdminForm.Meta (class in events.adminForms), 8
events.admin (module), 8
events.adminForms (module), 8
events.models (module), 5
EventSerializer (class in events.models), 7
EventSerializer.Meta (class in events.models), 7
exclude (commons.admin.CommonAdmin.Meta attribute), 4
exclude (commons.adminForms.CommonAdminForm.Meta attribute), 4
ExporterManager (class in tickets.exporters), 19
external_customer_name (tickets.models.TicketCode attribute), 14
external_fiscal_number (tickets.models.TicketCode attribute), 14
external_id (tickets.models.TicketCode attribute), 14
external_locator (tickets.models.TicketCode attribute), 14

F

FakeAttempt (class in tickets.models), 11
FakeAttempt.DoesNotExist, 12
FakeAttempt.MultipleObjectsReturned, 12
fields (tickets.models.AttemptListItemSerializer.Meta attribute), 11
fields (users.models.UserSerializer.Meta attribute), 21
fieldsets (tickets.admin.TicketPackAdmin attribute), 18
form (events.admin.EventAdmin attribute), 8
form (tickets.admin.TicketCodeAdmin attribute), 18
form (tickets.admin.TicketPackAdmin attribute), 18

G

generate_codes() (tickets.admin.TicketPackAdmin method), 18
get_for_attempt() (tickets.models.TicketCodeManager method), 15
get_next_by_creation_date() (commons.models.CommonModel method), 4

get_next_by_creation_date() (events.models.Event method), 7
get_next_by_creation_date() (tickets.models.TicketCode method), 15
get_next_by_creation_date() (tickets.models.TicketPack method), 17
get_next_by_date() (tickets.models.AbstractAttempt method), 9
get_next_by_date() (tickets.models.Attempt method), 11
get_next_by_date() (tickets.models.FakeAttempt method), 12
get_next_by_end_date() (events.models.Event method), 7
get_next_by_modified_date() (commons.models.CommonModel method), 4
get_next_by_modified_date() (events.models.Event method), 7
get_next_by_modified_date() (tickets.models.TicketCode method), 15
get_next_by_modified_date() (tickets.models.TicketPack method), 17
get_next_by_start_date() (events.models.Event method), 7
get_prefix() (tickets.models.TicketPack method), 17
get_previous_by_creation_date() (commons.models.CommonModel method), 4
get_previous_by_creation_date() (events.models.Event method), 7
get_previous_by_creation_date() (tickets.models.TicketCode method), 15
get_previous_by_creation_date() (tickets.models.TicketPack method), 17
get_previous_by_date() (tickets.models.AbstractAttempt method), 9
get_previous_by_date() (tickets.models.Attempt method), 11
get_previous_by_date() (tickets.models.FakeAttempt method), 12
get_previous_by_end_date() (events.models.Event method), 7
get_previous_by_modified_date() (commons.models.CommonModel method), 4
get_previous_by_modified_date() (events.models.Event method), 7
get_previous_by_modified_date() (tickets.models.TicketCode method), 15
get_previous_by_modified_date() (tickets.models.TicketPack method), 17
get_previous_by_start_date() (events.models.Event method), 7
get_select_options() (tickets.importers.ImporterManager method), 20
get_status_display() (tickets.models.TicketCode method), 15
get_ticket_count() (tickets.models.TicketPack method),

17
 get_tickets() (tickets.exporters.BaseExporter method), 19
 get_tickets() (tickets.exporters.BaseTicketExporter method), 19
 get_tickets() (tickets.exporters.BaseTicketPackExporter method), 19

I

id (events.models.Event attribute), 5
 id (tickets.models.Attempt attribute), 9
 id (tickets.models.FakeAttempt attribute), 11
 id (tickets.models.TicketCode attribute), 12
 id (tickets.models.TicketPack attribute), 15
 ImporterManager (class in tickets.importers), 20
 iter_tickets() (tickets.importers.BaseImporter method), 20

L

label (tickets.importers.AtrapaloImporter attribute), 20
 label (tickets.importers.BaseImporter attribute), 20
 label (tickets.importers.DefaultImporter attribute), 20
 label (tickets.importers.ProductesDeLaTerraImporter attribute), 20
 label (tickets.importers.TicketeaImporter attribute), 20
 list_display (events.admin.EventAdmin attribute), 8
 list_display (tickets.admin.TicketCodeAdmin attribute), 18
 list_display (tickets.admin.TicketPackAdmin attribute), 18
 list_filter (events.admin.EventAdmin attribute), 8
 list_filter (tickets.admin.TicketCodeAdmin attribute), 18
 list_filter (tickets.admin.TicketPackAdmin attribute), 18
 list_per_page (tickets.admin.TicketCodeAdmin attribute), 18
 luhn_digit() (in module tickets.helpers), 20

M

make_attempt() (tickets.admin.TicketCodeAdmin method), 18
 make_attempt() (tickets.models.TicketCode method), 15
 make_attempt() (tickets.models.TicketCodeManager method), 15
 map_to_ticket() (tickets.importers.AtrapaloImporter method), 20
 map_to_ticket() (tickets.importers.BaseImporter method), 20
 map_to_ticket() (tickets.importers.ProductesDeLaTerraImporter method), 20
 map_to_ticket() (tickets.importers.TicketeaImporter method), 20
 mark_as_new() (tickets.admin.TicketCodeAdmin method), 18
 mark_as_used() (tickets.admin.TicketCodeAdmin method), 18
 media (commons.admin.CommonAdmin attribute), 4

media (commons.adminForms.CommonAdminForm attribute), 4
 media (events.admin.EventAdmin attribute), 8
 media (events.adminForms.EventAdminForm attribute), 8
 media (tickets.admin.TicketCodeAdmin attribute), 18
 media (tickets.admin.TicketPackAdmin attribute), 18
 media (tickets.adminForms.TicketPackAdminForm attribute), 18
 mime_type (tickets.exporters.BaseExporter attribute), 19
 model (events.adminForms.EventAdminForm.Meta attribute), 8
 model (events.models.EventSerializer.Meta attribute), 7
 model (tickets.adminForms.TicketPackAdminForm.Meta attribute), 17
 model (tickets.models.AttemptListItemSerializer.Meta attribute), 11
 model (tickets.models.AttemptSerializer.Meta attribute), 11
 model (tickets.models.TicketCodeSerializer.Meta attribute), 15
 model (tickets.models.TicketPackSerializer.Meta attribute), 17
 model (users.models.UserSerializer.Meta attribute), 21
 modified_by (commons.models.CommonModel attribute), 3, 4
 modified_by (events.models.Event attribute), 6, 7
 modified_by (tickets.models.TicketCode attribute), 13, 15
 modified_by (tickets.models.TicketPack attribute), 16, 17
 modified_date (commons.models.CommonModel attribute), 3
 modified_date (events.models.Event attribute), 5
 modified_date (tickets.models.TicketCode attribute), 12
 modified_date (tickets.models.TicketPack attribute), 16

N

name (events.models.Event attribute), 6
 name (tickets.models.TicketPack attribute), 16

O

objects (events.models.Event attribute), 7
 objects (tickets.models.Attempt attribute), 11
 objects (tickets.models.FakeAttempt attribute), 12
 objects (tickets.models.TicketCode attribute), 15
 objects (tickets.models.TicketPack attribute), 17

P

prefix (events.models.Event attribute), 6
 prefix (tickets.models.TicketPack attribute), 16
 process_upload_field() (tickets.admin.TicketPackAdmin method), 18
 ProductesDeLaTerraImporter (class in tickets.importers), 20

put_ticket_number() (tickets.admin.TicketPackAdmin method), 18

R

register() (tickets.exporters.ExporterManager method), 19

register() (tickets.importers.ImporterManager method), 20

remove_attempts() (tickets.admin.TicketCodeAdmin method), 18

S

save() (tickets.models.AbstractAttempt method), 9

save_model() (commons.admin.CommonAdmin method), 4

save_model() (tickets.admin.TicketPackAdmin method), 18

search_fields (events.admin.EventAdmin attribute), 8

search_fields (tickets.admin.TicketCodeAdmin attribute), 18

short_description (tickets.exporters.BaseExporter attribute), 19

short_description (tickets.exporters.BaseTicketExporter attribute), 19

short_description (tickets.exporters.BaseTicketPackExporter attribute), 19

short_description (tickets.exporters.DefaultTicketExporter attribute), 19

short_description (tickets.exporters.DefaultTicketPackExporter attribute), 19

short_description (tickets.exporters.DefaultTicketPackWithAttemptsExporter attribute), 19

short_description (tickets.exporters.DefaultTicketWithAttemptsExporter attribute), 19

start_date (events.models.Event attribute), 6

status (tickets.models.TicketCode attribute), 13

success (tickets.models.Attempt attribute), 10

T

ticket_code (tickets.models.Attempt attribute), 10, 11

TICKET_CODE_STATUS (tickets.models.TicketCode attribute), 15

TICKET_CODE_STATUS_DISABLED (tickets.models.TicketCode attribute), 15

TICKET_CODE_STATUS_NEW (tickets.models.TicketCode attribute), 15

TICKET_CODE_STATUS_USED (tickets.models.TicketCode attribute), 15

ticket_codes (tickets.models.TicketPack attribute), 17

ticket_number (tickets.models.TicketCode attribute), 14

ticket_pack (tickets.models.TicketCode attribute), 13, 15

TicketCode (class in tickets.models), 12

TicketCode.DoesNotExist, 14

TicketCode.MultipleObjectsReturned, 15

TicketCodeAdmin (class in tickets.admin), 18

TicketCodeManager (class in tickets.models), 15

TicketCodeSerializer (class in tickets.models), 15

TicketCodeSerializer.Meta (class in tickets.models), 15

TicketeaImporter (class in tickets.importers), 20

TicketPack (class in tickets.models), 15

TicketPack.DoesNotExist, 17

TicketPack.MultipleObjectsReturned, 17

TicketPackAdmin (class in tickets.admin), 18

TicketPackAdminForm (class in tickets.adminForms), 17

TicketPackAdminForm.Meta (class in tickets.adminForms), 17

TicketPackSerializer (class in tickets.models), 17

TicketPackSerializer.Meta (class in tickets.models), 17

tickets.admin (module), 18

tickets.adminForms (module), 17

tickets.exporters (module), 19

tickets.helpers (module), 20

tickets.importers (module), 20

tickets.models (module), 9

U

unregister() (tickets.exporters.ExporterManager method), 19

unregister() (tickets.importers.ImporterManager method), 20

user (tickets.models.AbstractAttempt attribute), 9

user (tickets.models.Attempt attribute), 10, 11

user (tickets.models.FakeAttempt attribute), 11, 12

users.models (module), 21

UserSerializer (class in users.models), 21

UserSerializer.Meta (class in users.models), 21

W

write_data() (tickets.exporters.BaseExporter method), 19

write_data() (tickets.exporters.DefaultTicketExporter method), 19

write_item() (tickets.exporters.DefaultTicketExporter method), 19

write_item() (tickets.exporters.DefaultTicketWithAttemptsExporter method), 19