API Documentation

API Documentation

January 19, 2013

Contents

1 Package morbid

1.1 Modules

- **admin** *(Section 2, p. 4)*
- **models** *(Section 3, p. 6)*
- **tests**: This file demonstrates writing tests using the unittest module.
  (Section 4, p. 11)
- **urls** *(Section 5, p. 12)*
- **views** *(Section 6, p. 13)*

2 Module morbid.admin

2.1 Class FeatureAnalyticTicketAdmin

django.contrib.admin.ModelAdmin —┐

morbid.admin.FeatureAnalyticTicketAdmin

2.1.1 Class Variables

Name	Description
list_display	**Value:** 'analytic', 'ticket', 'feature', 'value'

2.2 Class TargetPriceAdmin

django.contrib.admin.ModelAdmin —┐

morbid.admin.TargetPriceAdmin

2.2.1 Class Variables

Name	Description
list_display	**Value:** 'analytic', 'date', 'price', 'ticket'

2.3 Class TicketAdmin

django.contrib.admin.ModelAdmin —┐

morbid.admin.TicketAdmin

2.3.1 Class Variables

Name	Description
list_display	**Value:** 'name', 'long_name'

2.4 Class FeatureAdmin

django.contrib.admin.ModelAdmin —┐

morbid.admin.FeatureAdmin

2.4.1 Class Variables

Name	Description
list_display	**Value:** `'name', 'unit', 'display_in_frontpage'`

3 Module morbid.models

3.1 Class Unit

django.db.models.Model ┐

                     morbid.models.Unit
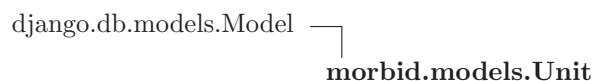
The units of the feature

3.1.1 Methods

__unicode__(*self*)
The object return name

3.1.2 Class Variables

Name	Description
name	Name of the unit of measure **Value:** `models.CharField(max_length= 200)` *(type=str)*
value	The value (sign) of the measure (days, percent) **Value:** `models.CharField(max_length= 200)` *(type=str)*

3.2 Class Feature

django.db.models.Model ┐

                     morbid.models.Feature
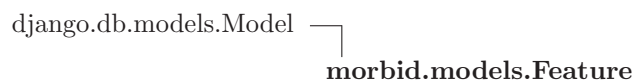
The feature of the measure

3.2.1 Methods

__unicode__(*self*)
The object return name

3.2.2 Class Variables

Name	Description
name	Name of the measure **Value:** `models.CharField(max_length= 200)` *(type=str)*

Name	Description
unit	The units of the measure **Value:** `models.ForeignKey(Unit)` *(type=Unit)*
display_in_frontpage	Boolean to display in frontpage **Value:** `models.BooleanField()` *(type=boolean)*
description	Description of the feature **Value:** `models.TextField()` *(type=text)*

3.3   Class Analytic

django.db.models.Model ⌐

morbid.models.Analytic

Analysing analytic

3.3.1   Methods

natural_key(*self*)

Required for serialization

Return Value
    `string`

__unicode__(*self*)

The object return name

3.3.2   Class Variables

Name	Description
name	Name of analytic **Value:** `models.CharField(max_length= 200)` *(type=str)*
number_of_companies	Number of tp analytic released **Value:** `models.IntegerField()` *(type=integer)*
number_of_tp	Number of target prices released by analytic **Value:** `models.IntegerField()` *(type=integer)*
volatility	Measure of uncertainty **Value:** `models.IntegerField()` *(type=integer)*
last_target_price	Last target price analytic released **Value:** `models.FloatField()` *(type=float)*
slug	Slug to reach the page of analytic **Value:** `models.SlugField()` *(type=str)*

3.4   Class Ticket

django.db.models.Model ⎤
                       morbid.models.Ticket
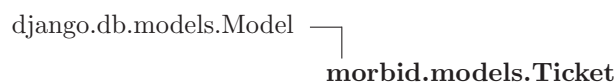
The ticket (or more precise to call it a company) name

3.4.1   Methods

natural_key(*self*)
The natural return key – not index (pk), but the name (short name) of the ticket
Return Value
string

__unicode__(*self*)
The object return name

3.4.2   Class Variables

Name	Description
name	Name of the ticket (AAPL, GOOG) **Value:** `models.CharField(max_length= 200)` *(type=str)*
long_name	Full name of the ticket (Apple Inc, Google Inc) **Value:** `models.CharField(max_length= 200)` *(type=str)*
last_stock_price	Last stock price of the ticket (live update maybe) **Value:** `models.FloatField()` *(type=float)*
number_of_analytics	Number of analytics analyzing the ticket **Value:** `models.IntegerField()` *(type=integer)*
number_of_tp	How many target prices does the ticket have **Value:** `models.IntegerField()` *(type=integer)*
consensus_min	Minimum value for the consensus measure **Value:** `models.FloatField()` *(type=float)*
consensus_avg	Average value for the consensus measure **Value:** `models.FloatField()` *(type=float)*
consensus_max	Maximum value for the consensus measure **Value:** `models.FloatField()` *(type=float)*
slug	The slug to reach the page **Value:** `models.SlugField()` *(type=str)*

3.5 Class FeatureAnalyticTicket

django.db.models.Model ⎯⎯⎤

morbid.models.FeatureAnalyticTicket

The place where analytic, ticket and feature meets

3.5.1 Methods

__unicode__(*self*)
The object return name
Return Value
string

3.5.2 Class Variables

Name	Description
value	The value of the feature **Value:** `models.FloatField()` *(type=float)*
feature	Defines the type of the feature (how near, fixation, ...) **Value:** `models.ForeignKey(Feature)` *(type=Feature)*
analytic	The analytic for which the feature was calculated **Value:** `models.ForeignKey(Analytic)` *(type=Analytic)*
ticket	The ticket on which the feature was calculated **Value:** `models.ForeignKey(Ticket)` *(type=Ticket)*

3.6 Class TargetPrice

django.db.models.Model ⎯⎯⎤

morbid.models.TargetPrice

The list of all the target prices to show to people

3.6.1 Class Variables

Name	Description
date	The date of released target price **Value:** `models.DateField()` *(type=Date)*
price	The price of the target price **Value:** `models.FloatField()` *(type=Float)*
ticket	The ticket on which target price was released **Value:** `models.ForeignKey(Ticket)` *(type=Ticket)*

Name	Description
analytic	The analytic, which published the target price
	Value: `models.ForeignKey(Analytic)`
	(type=`Analytic`)

4 Module morbid.tests

This file demonstrates writing tests using the unittest module. These will pass when you run "manage.py test".

Replace this with more appropriate tests for your application.

4.1 Class SimpleTest

django.test.TestCase ⌐

morbid.tests.SimpleTest

4.1.1 Methods

test_basic_addition(*self*)
Tests that 1 + 1 always equals 2.

5 Module morbid.urls

5.1 Variables

Name	Description
urlpatterns	**Value:** `patterns('morbid.views', url(r'^$', 'index'), url(r'^anal...`

6 Module morbid.views

6.1 Functions

---

index(*request*)

Index page

The main page of the app

Return Value
    Http Response

---

ticket(*request, slug*)

Ticket page

Parameters
    `slug`: the request for the specific ticket

Return Value
    Http Response. If the ticket was not found – rises the 404 error

---

analytic(*request, slug*)

Analytic page

Parameters
    `slug`: the slug to reach the page

Return Value
    Http Response. If the analytic was not found by the slug – 404 error rises

---

feature_by_ticket(*self, slug, feature_id*)

The feature return management on the ticket page

Parameters
    `slug`:   the slug to reach the ticket

    `feature_id`: the feature id to return

Return Value
    Http Response. If the ticket was not found using the slug – 404 error rises

    JSON only

---

feature_by_analytic(*self, slug, feature_id*)

The feature return management on the analytic page

Parameters
    `slug`:   the slug to reach the

    `feature_id`: the feature id to return

Return Value
    Http Response JSON only. If the analytic was not found using the slug – 404
    error rises

---

search(*self, search_me*)

The search of the page

Parameters
    `search_me:`   The string to query search the database

Return Value
    Http Response in JSON.

Index