Django Models Internals Docs Documentation

Release 0.1

Brian Luft

April 05, 2016

Contents

1 abstract 3

2 abstract_managers 5

3 add_field 7

4 add_virtual_field 9

5 admin 11

6 app_label 13

7 auto_created 15

8 auto_field 17

9 concrete_managers 19

10 contribute_to_class 21

11 db_table 23

12 db_tablespace 25

13 duplicate_targets 27

14 fields 29

15 get_add_permission 31

16 get_all_field_names 33

17 get_all_related_m2m_objects_with_model 35

18 get_all_related_many_to_many_objects 37

19 get_all_related_objects 39

20 get_all_related_objects_with_model 41

21 get_ancestor_link 43

22 get_base_chain 45

23 get_change_permission 47

24 get_delete_permission 49

25 get_field 51

26 get_field_by_name 53

27 get_fields_with_model 55

28 get_latest_by 57

29 get_m2m_with_model 59

30 get_ordered_objects 61

31 get_parent_list 63

32 has_auto_field 65

33 init_name_map 67

34 installed 69

35 local_fields 71

36 local_many_to_many 73

37 managed 75

38 many_to_many 77

39 module_name 79

40 object_name 81

41 order_with_respect_to 83

42 ordering 85

43 parents 87

44 permissions 89

45 pk 91

46 pk_index 93

47 proxy 95

48 proxy_for_model 97

49 setup_pk 99

50 setup_proxy 101

51 unique_together 103

52 verbose_name 105

53 verbose_name_plural 107

54 verbose_name_raw 109

55 virtual_fields 111

56 Indices and tables 113

The goal of this project is to create and collect a good set of reference information for the Django Model internals as of version 1.2.1. Rather than repeat information that the Django documentation already covers adequately, the permalink back to the Django docs will be favored.

Additionally, collecting high quality tutorials and/or well documented snippets into a single location will help users get a better sense of the patterns involved in tweaking model internals.

Model Options:

CHAPTER 1

abstract

In the Django documentation:

- Abstract base classes
- Model options reference

CHAPTER 2

abstract_managers

TODO: need description

CHAPTER 3

add_field

TODO: need description

CHAPTER 4

add_virtual_field

TODO: need description

CHAPTER 5

admin

TODO: need description

CHAPTER 6

app_label

TODO: need description

CHAPTER 7

auto_created

TODO: need description

CHAPTER 8

auto_field

TODO: need description

CHAPTER 9

concrete_managers

TODO: need description

CHAPTER 10

contribute_to_class

TODO: need description

CHAPTER 11

db_table

TODO: need description

CHAPTER 12

db_tablespace

TODO: need description

CHAPTER 13

duplicate_targets

TODO: need description

CHAPTER 14

fields

TODO: need description

CHAPTER 15

get_add_permission

TODO: need description

CHAPTER 16

get_all_field_names

Returns a list of all field names that are possible for this model (including reverse relation names). This is used for pretty printing debugging output (a list of choices), so any internal-only field names are not included.

CHAPTER 17

get_all_related_m2m_objects_with_model

Returns a list of (related-m2m-object, model) pairs. Similar to get_fields_with_model().

CHAPTER 18

get_all_related_many_to_many_objects

TODO: need description

CHAPTER 19

get_all_related_objects

TODO: need description

CHAPTER 20

get_all_related_objects_with_model

Returns a list of (related-object, model) pairs. Similar to get_fields_with_model().

CHAPTER 21

get_ancestor_link

Returns the field on the current model which points to the given "ancestor". This is possible an indirect link (a pointer to a parent model, which points, eventually, to the ancestor). Used when constructing table joins for model inheritance. Returns None if the model isn't an ancestor of this one.

CHAPTER 22

get_base_chain

Returns a list of parent classes leading to 'model' (order from closet to most distant ancestor). This has to handle the case were 'model' is a granparent or even more distant relation.

CHAPTER 23

get_change_permission

TODO: need description

CHAPTER 24

get_delete_permission

TODO: need description

CHAPTER 25

get_field

Returns the requested field by name. Raises FieldDoesNotExist on error.

CHAPTER 26

get_field_by_name

Returns the (field_object, model, direct, m2m), where field_object is the Field instance for the given name, model is the model containing this field (None for local fields), direct is True if the field exists on this model, and m2m is True for many-to-many relations. When 'direct' is False, 'field_object' is the corresponding RelatedObject for this field (since the field doesn't have an instance associated with it). Uses a cache internally, so after the first access, this is very fast.

CHAPTER 27

get_fields_with_model

Returns a sequence of (field, model) pairs for all fields. The "model" element is None for fields on the current model. Mostly of use when constructing queries so that we know which model a field belongs to.

CHAPTER 28

get_latest_by

TODO: need description

CHAPTER 29

get_m2m_with_model

The many-to-many version of get_fields_with_model().

CHAPTER 30

get_ordered_objects

Returns a list of Options objects that are ordered with respect to this object.

CHAPTER 31

get_parent_list

Returns a list of all the ancestor of this model as a list. Useful for determining if something is an ancestor, regardless of lineage.

CHAPTER 32

has_auto_field

TODO: need description

CHAPTER 33

init_name_map

Initialises the field name -> field object mapping.

CHAPTER 34

installed

TODO: need description

CHAPTER 35

local_fields

TODO: need description

CHAPTER 36

local_many_to_many

TODO: need description

CHAPTER 37

managed

TODO: need description

CHAPTER 38

many_to_many

TODO: need description

CHAPTER 39

module_name

TODO: need description

CHAPTER 40

object_name

TODO: need description

CHAPTER 41

order_with_respect_to

TODO: need description

CHAPTER 42

ordering

TODO: need description

CHAPTER 43

parents

TODO: need description

CHAPTER 44

permissions

TODO: need description

CHAPTER 45

pk

TODO: need description

CHAPTER 46

pk_index

Returns the index of the primary key field in the self.fields list.

CHAPTER 47

proxy

TODO: need description

CHAPTER 48

proxy_for_model

TODO: need description

CHAPTER 49

setup_pk

TODO: need description

CHAPTER 50

setup_proxy

Does the internal setup so that the current model is a proxy for “target”.

CHAPTER 51

unique_together

TODO: need description

CHAPTER 52

verbose_name

TODO: need description

CHAPTER 53

verbose_name_plural

TODO: need description

CHAPTER 54

verbose_name_raw

TODO: need description

CHAPTER 55

virtual_fields

TODO: need description

CHAPTER 56

Indices and tables

- genindex
- modindex
- search