
Django-inspect-model Documentation

Release 0.1

Mathieu Agopian

May 23, 2016

1	Installation	3
2	Usage	5
3	Changes	7
4	Indices and tables	9

Django-inspect-model is a model inspection utility for Django. It allows you to easily list all available “items” on a model, and get their value.

An item is either:

- a django field (standard field or relation field)
- a standard attribute
- a method that only takes one attribute: ‘self’

The code is generic enough to be applied on just any python object, so Django isn’t a requirement. However, it was tailored towards Django models.

The source code is [available on Github](#) under the 3-clause BSD licence.

Installation

```
pip install django-inspect-model
```

Usage

Instantiate `inspect_model.InspectModel` with your model class or instance, and profit.

```
>>> from django.contrib.comments.models import Comment
>>> from inspect_model import InspectModel
>>> im = InspectModel(Comment)
>>> im.fields
['comment', 'id', 'ip_address', 'is_public', 'is_removed', 'object_pk',
'submit_date', 'user_email', 'user_name', 'user_url']
>>> im.relation_fields
['content_type', 'site', 'user']
>>> im.many_fields
['flags']
>>> im.attributes
[]
>>> im.methods
['get_as_text', 'get_content_object_url']
>>> im.items
['comment', 'content_type', 'flags', 'get_as_text',
'get_content_object_url', 'id', 'ip_address', 'is_public', 'is_removed',
'object_pk', 'site', 'submit_date', 'user', 'user_email', 'user_name',
'user_url']
```

Changes

- 0.4: renamed `get_FOO` methods to `update_FOO`, `self.items` uses a sorted set internally
- 0.3: added `self.relation_fields` and `self.many_fields`
- 0.2: added `self.items`
- 0.1: initial version

Indices and tables

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