nicedjango-py Documentation

Release latest

November 25, 2014

Contents

Nice django tools

CHAPTER 1

ModelGraph

Selective dumping and loading of only the needed model data for all objects and their related objects of one or more querysets.

This is done by

- getting a graph of all relations between models,
- getting all pks first in chunks,
- dump them in an order that enables correct loading.

1.1 Examples

```
# show model graph parts that would be dumped and those which not:
# example for query model a1.A with relation to child a1.B(A)
./manage.py dump_graph -p -q a -r a.b
    a1-a:
               a1-a.b          to child            a1-b.pk
       excludes:
               a1-a.f          to foreign          a1-f.a
     a1-b:
               a1-b.pk         to parent           a1-a.pk
       excludes:
               a1-b.c          to child            a1-c.pk
               a1-b.e          to child            a1-e.pk

# dump all objects from a1.models.A.objects.filter() with relation a.b as compact yaml:
./manage.py dump_graph -f dump.yaml -s compact_yaml -q a.filter(pk__in=(1,2)) -r a.b
    - a1-a: [pk]
     - [1]
     - [2]
     - a1-b: [pk]
     - [2]

# load back the dumped dump.yaml
./manage.py load_graph -f dump.yaml -s compact_yaml

# by default serializing into compact csv files is enabled:
mkdir dump_folder
./manage.py dump_graph -f dump_folder -q a.filter(pk__in=(1,2)) -r a.b
#results in two files under dump_folder:
# a1-a.csv:
```

```
 a1-a:pk
 1
 2
# and a1-b.csv:
 a1-b:pk
 2
```