From basic distance search to a complex multi criteria search

Antonin Lacombe
EuroPython 2015
@antoninlacombe
About Me

- Antonin Lacombe
- Django and Python dev since 2009
- Technical Lead @ JeLoueMonCampingCar.com
  - AlquilarMiAutoCaravana.com (ES)
  - IRentMyMotorhome.com (EN)
  - AlugoMiNhaautoCaravana.com (PT)
  - LlogarLaMevaAutoCaravana.com (CAT)
  - NoleggioIlMioCamper.com (IT)

@antoninlaccombe
JeLoueMonCampingCar.com

posts an ad

makes money

looks for a vehicle

enjoy holidays

Owner

JLM

Traveler
What we are going to talk?

<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>• distance search</td>
<td>• installing haystack</td>
</tr>
<tr>
<td>• haystack backends</td>
<td>• installing ElasticSearch</td>
</tr>
<tr>
<td>• ES function_score</td>
<td>• configure indexes</td>
</tr>
<tr>
<td>• campers</td>
<td>• go deep in ES</td>
</tr>
</tbody>
</table>
Previously at JLM

Basic distance search like this:

```python
searched_point = Point(longitude, latitude)
Ad.objects.distance(searched_point).order_by('distance')
```
And What?
Searching is hard
maybe harder than you think
Closer is not the better
What do we want?

Reward owners that deliver a great experience to guests

- rent vehicle close to a city
- to a top notch owner (rent often, good reviews, answer fast...)
- an available vehicle
- with many pictures
Let's do things better
Select a stack
Rewrite with Haystack

Basic distance search search using Haystack and ES:

```python
squared = SearchQuerySet().models(Ad).distance('location', searched_point).order_by('distance')
```
I said better

Not the same function refactored
Problem....

We CAN'T add more ordering rules with Haystack

Warning

You can not specify both a distance & lexicographic ordering. If you specify more than just `distance` or `-distance`, Haystack assumes distance is a field in the index & tries to sort on it.
This is a limitation in the engine’s implementation.
If you actually have a field called `distance` (& aren’t using calculated distance information), Haystack will do the right thing in these circumstances.
function_score and decay

- function_score allow you to combine multiple functions like decay

- function_score are not implemented in haystack (decay neither)
GET /_search
{
  "query": {
    "function_score": {
      "query": {...},
      "score_mode": "sum",
      "functions": [
        {
          "gauss": {
            "location": {
              "origin": { "lat": 51.5, "lon": 0.12 },
              "offset": "2km",
              "scale": "3km"
            }
          }
        },
        {
          "gauss": {
            "price": {
              "origin": "50",
              "offset": "50",
              "scale": "20"
            }
          }
        },
        "weight": 2
      ]
    }
  }
}
So
Let's write a custom ES backend
More than a backend

- We need write:
  - ElasticsearchSearchEngine
    - ElasticsearchSearchBackend
    - ElasticsearchSearchQuery
  - SearchQuerySet
class FunctionScoreSearchQuery(ElasticsearchSearchQuery):

    def __init__(self, **kwargs):
        super(FunctionScoreSearchQuery, self).__init__(**kwargs)
        self.decay_functions = []

    def build_params(self, *args, **kwargs):
        search_kwargs = super(
            FunctionScoreSearchQuery, self).build_params(*args, **kwargs)
        if self.decay_functions:
            search_kwargs['decay_functions'] = self.decay_functions
        return search_kwargs

    def add_decay_function(self, function_dict):
        self.decay_functions.append(function_dict)

    def _clone(self, **kwargs):
        clone = super(FunctionScoreSearchQuery, self)._clone(**kwargs)
        clone.decay_functions = self.decay_functions[:]
        return clone
class FunctionScoreSearchBackend(ElasticsearchSearchBackend):

    """
    Search backend that add the function score to the elastic search query
    """

def build_search_kwargs(self, query_string, decay_functions=None, **kwargs):
    
    kwargs = super(FunctionScoreSearchBackend, self).build_search_kwargs(query_string, **kwargs)
    if not decay_functions:
        return kwargs

    original_query = kwargs['query']
    function_score_query = {
        'function_score': {
            'functions': decay_functions,
            'query': original_query,
            'score_mode': 'sum'
        }
    }
    kwargs['query'] = function_score_query
    return kwargs
class FunctionScoreSearchQuerySet(SearchQuerySet):

    '''
    usage example:
    SearchQuerySet().filter(text='foo').decay({'gauss': {'end_time':
        {'origin': '2014-05-07', 'scale': '10d' }}})
    '''

    def decay(self, function_dict):
        clone = self._clone()
        clone.query.add_decay_function(function_dict)
        return clone

class CustomEsSearchEngine(ElasticsearchSearchEngine):
    backend = FunctionScoreSearchSearchBackend
    query = FunctionScoreSearchSearchQuery
Available on GitHub

https://github.com/JeLoueMonCampingCar/haystack-function-score-backend

@antoninlacombe
Let’s use it now!
from search.custom_elasticsearch import FunctionScoreSearchQuerySet

sqs = FunctionScoreSearchQuerySet().models(Ad)
# custom filters
sqs = sqs.filter(park_assistance=True)
sqs = sqs.filter(gps=True)
# compute the distance
search_location = Point(longitude, latitude)
sqs = sqs.distance('vehicle_location', search_location)
# create the distance decay
decay_distance = {
    'exp': {
        'vehicle_location': {
            'origin': {'lat': str(latitude), 'lon': str(longitude)},
            'scale': '50km',
            'offset': '15km',
        },
    },
    "weight": 2,
}
sqs = sqs.decay(decay_distance)
# decay gauss depending on the picture count

decay_picture_count = {
    'gauss': {
        'pictures_count': {
            'origin': 50,
            'offset': 40,
            'scale': 9
        }
    },
    "weight": 0.5,
}
sqs = sqs.decay(decay_picture_count)

# decay linear depending on the host activity rate

decay_owner_quality_rate = {
    'linear': {
        'owner_quality_rate': {
            'origin': 1.0,
            'scale': 0.5,
        }
    }
}
sqs = sqs.decay(decay_owner_quality_rate)
Motorhome hire

1. Ford Transit 125T350
   - Nanterre (92) à 12km de Paris
   - 1 reviews received after rental
   - Bonus "GROUPE ELECTOGÈNE AU GAZ" (pour une autonomie total) OFFRE EXCEPTIONNELLE! ABS avec roues jumelées, idéal pour une famille, tout confort et aussi bien pour les longs ou...
   - From 119 € per rental day
   - 6 seats 6 Berth
   - Score: 0.21775 Owner quality rate: 0.218 Pictures count: 8

2. Pilote Atlantis A2
   - Rosny Sous Bois (93) à 10km de Paris
   - 1 reviews received after rental
   - Camping Car PILOTE ATLANTIS A2 très compact (6 mètres) Petit camping car compacte idéal pour les ferrys... Sur fiat ducato 2.8, 127cv, année 2003. 6 places carte grise et 6...
   - From 91 € per rental day
   - 6 seats 6 Berth
   - Score: 0.16011 Owner quality rate: 0.160 Pictures count: 3

3. Fiat Bavaria T71
   - 6 seats 6 Berth
   - Score: 0.045 Owner quality rate: 0.045 Pictures count: 2

1,439 motorhomes to rent
HMM... THE PRESENTATION IS FINISHED

ANY.... QUESTION?