

Strategies for Using MongoDB with Django

Presented by:

Alan Viars

<http://about.me/aviars>

+1 202 503 4228

@aviars

Videntity - <http://videntity.com>

Why MongoDB

- “Best of breed” NoSQL technology (opinion).
- Very easy to setup and use.
- Extremely fast and scalable. Written in C++.
- Good Python support with the `pymongo` library. `pymongo` has optional C bindings for extra speed.
- Data is “schema-less”.

MongoDB in a Nutshell

- MongoDB is organized by *databases* and *collections*.
- Each database can have multiple collections.
- A collection can be thought of like an SQL table, except each “row” is a *document* (i.e. a dict-like object), thus the “columns” can vary between rows.

A MongoDB Document

A document is almost like a dict/JSON

```
person = {  
    "description": "Alan's person record.",  
    "sinceid": 1,  
    "datetime_added": 1382638235,  
    "name": "alan",  
    "happy": true,  
    "colors": ("black", "white", "clear"),  
    "resume": {"skills": ["django", "mongodb",  
                          "javascript"],  
              "hobbies": ["longboarding", "raspberrypi",  
                          "pickling"]}  
}
```

Simple Insert & Query Examples

```
// These examples work within the mongo shell
// Insert the document
db.people.save(person);
// Find me
db.people.find({'name': 'alan'});
// Find all that like any of these colors
db.people.find({'colors': $in: [ 'green',
'red','black' ]})
//Drill down into documents by using dot notation.
db.people.find({'resume.skills': 'django'});
```

Django and MongoDB can Play Nice

- We can use Django's form framework or Django's Model framework to define a “schema” for our data. The key is in overloading save() to write to MongoDB.
- Often you want some consistency even in your “schemaless” design. We can use the frameworks to ensure some fields are there and contain data formatted in a way we want.
- If you use the model framework the admin will work for Create/Update.

Playing Nice: URLs

- Use Django's url patterns to pass information about MongoDB. (e.g. the database and collection name).

...

```
url(r'^(?P<database_name>\w+)/(?P<collection_name>\w+)/ensureindex$', simple_ensure_index,  
name="simple_index_create"),
```

Playing Nice: User Management

- Although MongoDB does have its own user management system, it comes out-of-the-box and is often run in an “admin party” mode.
- Use Django's rich auth module to control access to MongoDB via a web app/API. Django allows you to create custom authentication back-ends So we could use LDAP for authentication to a resource in MongoDB.

Gotchas and Work Aroun

- Datetime can be tricky, especially when converting to JSON. I use epochs (integers) and within MongoDB and convert as needed.
- Django template engine doesn't like “_id”. If I need this in my template, I convert it to “id”. Another workaround is to create a template filter.
- Getting data back in the order it was put in is unreliable unless you use a “capped” collection. Use epochs or a “sinceid” in your documents as a work around.

Shameless Plug

- Use Restcat!
- An open source MongoDB CRUD operations + import/export capabilities in CSV, XLS, JSON, and XML.
- Out-of-the-box UI and RESTful API for MongoDB. Build your own “saved search” in the web user interface.
- Users are managed by Django, so you get all that built-in authorization infrastructure, including pluggable authentication.
- Social graph features (optional).

Contact Me

- Thanks for listening.
- We are looking for contributors.

Alan Viars

aviars@videntity.com or +1 202 503 4228