Find The Best Quality Results

Django retrieve data

Prerequisites

- Django Models [/course/django/models/]
- Django Insert Data [/course/django/insert-data/]

Django shell

- Run the command `python manage.py shell` open the django shell.
- Write django ORM queries to insert data into the database table.

Django Model - Contact

Do you like cookies? 🍪 We use cookies to ensure you get the best experience on our website. Learn more   I agree

my_app/models.py

```python
from django.db import models

class Contact(models.Model):
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
    email = models.EmailField(max_length=255)
    phone = models.CharField(max_length=10, null=True)

    class Meta:
        db_table = "contact"
```

insert the data

- Let's insert some data which will help in understanding the retrieve queries.

```python
from my_app.models import Contact

contacts = [
    {"first_name": "Johan", "last_name": "K", "email":
"johan@example.com"},
    {"first_name": "Johanan", "last_name": "M", "email":
"johanan@example.com"},
    {"first_name": "Joharr", "last_name": "K", "email":
"joharr@example.com"},
    {"first_name": "Mani", "last_name": "M", "email":
"mani@example.com"},
    {"first_name": "Rich", "last_name": "B", "email":
"rich@example.com"},
    {"first_name": "Ammulu", "last_name": "D", "email":
"ammulu@example.com"},
]
objects = [Contact(**d) for d in contacts]
results = Contact.objects.bulk_create(objects)
```

Retrieve all records - all()

Do you like cookies? 🍪 We use cookies to ensure you get the best
experience on our website. Learn more

- Look at the below query

```python
from my_app.models import Contact

contacts = Contact.objects.all()
print(contacts)
# output: <QuerySet [<Contact: Contact object (1)>, <Contact:
Contact object (2)>, <Contact: Contact object (3)>, <Contact:
Contact object (4)>, <Contact: Contact object (5)>, <Contact:
Contact object (6)>, <Contact: Contact object (7)>, <Contact:
Contact object (8)>, <Contact: Contact object (9)>, <Contact:
Contact object (10)>, <Contact: Contact object (11)>, <Contact:
Contact object (12)>]>
```

- To get `SQL` from the django orm query just use `query` attribute like below.

```python
from my_app.models import Contact

contacts = Contact.objects.all()
print(contacts.query)
```

- Above code outputs below SQL

```sql
SELECT "contact"."id", "contact"."first_name",
"contact"."last_name", "contact"."email", "contact"."phone" FROM
"contact"
```

Retrieve only one record - get()

- use `get()` method on django ORM to get only one record.
- Let's retrieve the record whose email is `mani@example.com`

```python
from my_app.models import Contact

mani = Contact.objects.get(email='mani@example.com')
print(mani)
```

Do you like cookies? 🍪 We use cookies to ensure you get the best experience on our website. Learn more

- Above, ORM query is equivalent to the below SQL

```sql
SELECT "contact"."id", "contact"."first_name",
"contact"."last_name", "contact"."email", "contact"."phone"
FROM "contact" WHERE email='mani@example.com'
```

Retrieve matching records - filter()

- Filter records whose name starts with "Johan"

```python
from my_app.models import Contact

contacts = Contact.objects.filter(first_name__startswith='Johan')
print(contacts)
# output: <QuerySet [<Contact: Contact object (7)>, <Contact:
Contact object (8)>]>
```

Count matching records - count()

- use `count()` method to count the matching records.

```python
from my_app.models import Contact

count =
Contact.objects.filter(first_name__startswith='Johan').count()
print(count)
# output: 2
```

Iterate through ORM records

- we can loop through the retrieved query results.

- These also called as queryset results.

Do you like cookies? 🍪 We use cookies to ensure you get the best experience on our website. Learn more

```python
from my_app.models import Contact

contacts = Contact.objects.filter(first_name__startswith='Johan')
for contact in contacts:
    print(contact.email)
# output:
# johan@example.com
# johanan@example.com
```

> Note: Django queries are very powerfull. We can do more by writing less code.

References

- https://docs.djangoproject.com/en/dev/topics/db/queries/
  [https://docs.djangoproject.com/en/dev/topics/db/queries/]

Do you like cookies? 🍪 We use cookies to ensure you get the best experience on our website. Learn more