

Лабораторная работа №6
«Работа с СУБД»

Выполнила:
Вранцева Наталья
Студентка гр.ИУ5-51

Описание задания лабораторной работы

В этой лабораторной работе вы познакомитесь с популярной СУБД MySQL, создать свою базу данных. Также нужно будет дополнить свои классы предметной области, связав их с созданной базой. После этого вы создадите свои модели с помощью Django ORM, отобразите объекты из БД с помощью этих моделей и ClassBasedViews.

Реализация

Настройка БД

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'django',  
        'USER': 'dbuser',  
        'PASSWORD': '123',  
        'HOST': 'localhost',  
        'PORT': '3306',  
        'OPTIONS': {'charset': 'utf8'},  
        'TEST_CHARSET': 'utf8'  
    }  
}
```

Класс подключения к БД без django.

```
import MySQLdb  
  
class Connection:  
    def __init__(self, user, passwd, db, host='localhost'):  
        self.user = user  
        self.passwd = passwd  
        self.db = db  
        self.host = host  
        self._connection = None  
  
    @property  
    def connection(self):  
        return self._connection  
  
    def __enter__(self):  
        self.connect()  
  
    def __exit__(self, exc_type, exc_val, exc_tb):  
        self.disconnect()  
  
    def connect(self):  
        if not self._connection:  
            self._connection = MySQLdb.connect(  
                host = self.host,  
                user = self.user,  
                passwd = self.passwd,  
                db = self.db  
            )  
  
    def disconnect(self):  
        if self._connection:  
            self._connection.close()
```

Класс Товары.

```
class Товар:
    def __init__(self, db_connection, name, desc, cout):
        self.db_connection = db_connection.connection
        self.name = name
        self.desc = desc
        self.cout = cout

    def save(self):
        c = self.db_connection.cursor()
        c.execute("INSERT INTO tovar (`name`, `desc`, cout) values (%s, %s, %s)",
                    (self.name, self.desc, self.cout))
        self.db_connection.commit()
        c.close()

con = Connection('dbuser', '123', 'first')

with con:
    tov = Товар(con, 'New order', 'Description', 255.25)
    tov.save()
```

Класс товары в django models

```
from django.db import models

class ТоварModel(models.Model):
    name = models.CharField(max_length=70)
    desc = models.CharField(max_length=300)
    cout = models.IntegerField()
```

Класс отображения товаров TovarView

```
class TovarView(View):
    def get(self, request):
        a = ТоварModel.objects.all()
        return render(request, 'order.html', {'a':a})
```

Urls.py

```
from django.conf.urls import url
from django.contrib import admin

from views import TovarView

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^test/$', TovarView.as_view())
]
```

Шаблон

```
{% extends 'base.html' %}

{% block title %}Товары{% endblock %}
```

```
{% block content %}

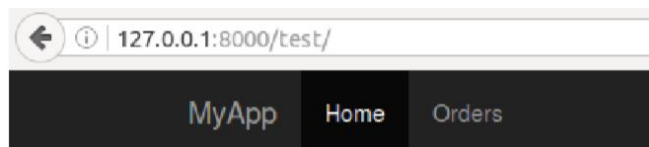
    Список товаров
    {% for i in a %}
        <br>{{ i.name }} : {{ i.desc }}
    {% empty %}
        <h1>Пустой список</h1>
    {% endfor %}
{% endblock %}
```

Скриншоты

Тестирование класса подключения без django

```
/usr/bin/python3.5 /home/toxa/PycharmProjects/Lab5/testApp/db.py
(1, 'Телевизор Samsung', '4К', 25999.5)
(3, 'Samsung note 7', 'Взрывная мощ', 45000.0)
(4, 'New order', 'Description', 255.25)
(5, 'New order', 'Description', 255.25)
(6, 'Samsung note 7', 'Взрывная мощ', 45000.0)
(7, 'Samsung note 7', 'Взрывная мощ', 45000.0)
```

Тестирование в браузере класса с Django



```
Список товаров
Товар1 : Описание1
Товар2 : Описание2
Товар3 : Описание3
```