



Django Busan User Group

Django로 만들어보는 '영화 추천 시스템' (1/5) v1.1

latest updated: 2018.12.11

Created by Sangkon Han([@sigmadream](#))

Part1. Winerama - a recommender tutorial

- @Jose A Dienes가 작성한 Winerama 튜토리얼 진행
 - 우리의 목표인 영화 추천 시스템을 만들기전에 '가볍게' 시작해 볼 수 있는 작은 규모의 추천 시스템
 - 해당 튜토리얼은 Python2 를 기반으로 작성되어 있어서 Python3 과 몇부분에선 호환이 되지 않고, 해당 튜토리얼에서 사용하는 Django 버전이 1.x을 사용함
 - 따라서 해당 튜토리얼을 진행하면서 Python Django 2.x 버전에서 실행할 수 있도록 약간의 코드를 수정하고, Python3 코드에 맞게 개선하면서 진행

Python 설치

OS X 사용자

- `brew` 를 사용해서 Python(>=3.5)를 설치

```
$ brew install python
```

Windows 사용자

- Python 3.6.7 64bit 설치
 - 설치시 열린마음으로 모든 '옵션'에 체크 할 것

Linux 사용자

- `pyenv` 나 `sudo apt install python3-pip python3-venv` 로 필요한 패키지 선택

VSCode 설치

OS X 사용자

```
$ brew cask  
$ brew cask install visual-studio-code
```

Windows / *NIX 사용자

- VSCode 를 다운로드 받아서 설치
 - 열린 마음으로 거의 대부분의 '옵션'에 체크 할 것

virtualenv

- 윈도우 사용자의 경우 PowerShell 을 CMD 로 변경하거나 Set-ExecutionPolicy Unrestricted 를 사용해서 제한사항을 해제해서 사용(개인적으로 CMD 를 사용할 것을 권장, 서명되지 않은 스크립트 실행을 허가하는 것은 일반적으로 위험함)
 - 작업 표시줄 설정 > 시작 단추를 마우스 오른쪽 [...] Windows PowerShell로 바꾸기 > 끄

```
C:\Users\USERNAME> mkdir works\winerama
C:\Users\USERNAME> cd works\winerama
C:\Users\USERNAME\works\winerama> python -m venv venv
C:\Users\USERNAME\works\winerama> venv\Scripts\activate
(venv) C:\Users\USERNAME\works\winerama>
```

- OS X, *NIX 사용자는 terminal 을 사용 할 것

```
$ mkdir -p Works/winerama
$ cd works/winerama
$ python -m venv venv
$ source venv/bin/activate
(venv) $
```

Django 및 필요한 라이브러리 설치

- 커맨드 라인의 (venv) 를 항상 확인할 것

```
(venv) pip list
Package      Version
-----
pip          10.0.1
setuptools   39.0.1
```

```
(venv) pip install django numpy
...
Successfully installed django-2.1.3 numpy-1.15.4 pytz-2018.7
```

```
(venv) pip list
Package      Version
-----
Django       2.1.3
```

```
(venv) pip freeze > requirement.txt
```

Django 프로젝트 생성

```
(venv) django-admin startproject winerama
(venv) cd winerama
(venv) python manage.py runserver
...
Run 'python manage.py migrate' to apply them.
November 30, 2018 - 02:27:23
Django version 2.1.3, using settings 'winerama.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

```
(venv) python manage.py migrate
...
Applying sessions.0001_initial... OK
```

Django 앱 추가

```
(venv) python manage.py startapp reviews
```

- `winerama/settings.py`

```
INSTALLED_APPS = (  
    ...  
    'django.contrib.staticfiles',  
    'reviews',  
)
```


Model 추가

- reviews\model.py

```
from django.db import models
import numpy as np

class Wine(models.Model):
    name = models.CharField(max_length=200)

    def average_rating(self):
        all_ratings = list(map(lambda x: x.rating,
                                self.review_set.all()))
        return np.mean(all_ratings)

    def __str__(self):
        return self.name
```

```
class Review(models.Model):
    RATING_CHOICES = (
        (1, '1'),
        (2, '2'),
        (3, '3'),
        (4, '4'),
        (5, '5'),
    )
    wine = models.ForeignKey(Wine, on_delete=models.PROTECT)
    pub_date = models.DateTimeField('date published')
    user_name = models.CharField(max_length=100)
    comment = models.CharField(max_length=200)
    rating = models.IntegerField(choices=RATING_CHOICES)
```

- DB에 반영

```
(venv) python manage.py makemigrations
(venv) python manage.py migrate
```

Admin 페이지

```
(venv) python manage.py createsuperuser
```

- `reviews/admin.py`

```
from django.contrib import admin

from .models import Wine, Review

class ReviewAdmin(admin.ModelAdmin):
    model = Review
    list_display = ('wine', 'rating', 'user_name',
                    'comment', 'pub_date')
    list_filter = ['pub_date', 'user_name']
    search_fields = ['comment']

admin.site.register(Wine)
admin.site.register(Review, ReviewAdmin)
```

URL 추가

- reviews\urls.py

```
from django.urls import path
from . import views

app_name = 'reviews'

urlpatterns = []
```

- winerama\urls.py

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('reviews/', include('reviews.urls',
                             namespace='reviews')),
    path('admin/', admin.site.urls),
]
```

View 추가

- `reviews\views.py`

```
from django.shortcuts import get_object_or_404, render

from .models import Review

def review_list(request):
    latest_review_list = Review.objects
        .order_by('-pub_date')[:9]
    context = {'latest_review_list': latest_review_list}
    return render(request, 'reviews/review_list.html',
                    context)

def review_detail(request, review_id):
    review = get_object_or_404(Review, pk=review_id)
    context = {'review': review}
    return render(request, 'reviews/review_detail.html',
                    context)
```

- reviews\urls.py

```
urlpatterns = [  
    path('', views.review_list, name='review_list'),  
    path('review/<int:review_id>/', views.review_detail,  
        name='review_detail'),  
]
```

Templates 추가

- winerama\reviews\templates\reviews\review_list.html

```
<h2>Latest reviews</h2>

{% if latest_review_list %}
<div>
    {% for review in latest_review_list %}
    <div>
        <h4><a href="{% url 'reviews:review_detail'
            review.id %}">
            {{ review.wine.name }}
        </a></h4>
        <h6>rated {{ review.rating }} of 5 by
            {{ review.user_name }}</h6>
        <p>{{ review.comment }}</p>
    </div>
    {% endfor %}
</div>
{% else %}
<p>No reviews are available.</p>
{% endif %}
```

- winerama\reviews\templates\reviews\review_detail.html

```
<h4>Rated {{ review.rating }} of 5 by  
        {{ review.user_name }}</h4>  
<p>{{ review.pub_date }}</p>  
<p>{{ review.comment }}</p>
```


View 추가(wine)

- reviews\views.py

```
from .models import Review, Wine

def wine_list(request):
    wine_list = Wine.objects.order_by('-name')
    context = {'wine_list': wine_list}
    return render(request, 'reviews/wine_list.html',
                  context)

def wine_detail(request, wine_id):
    wine = get_object_or_404(Wine, pk=wine_id)
    context = {'wine': wine}
    return render(request, 'reviews/wine_detail.html',
                  context)
```

Templates 추가(wine)

- winerama\reviews\templates\reviews\wine_list.html

```
<h2>Wine list</h2>

{% if wine_list %}
<div>
    {% for wine in wine_list %}
    <div>
        <h4>
            <a href="{% url 'reviews:wine_detail' wine.id %}">
                {{ wine.name }}
            </a>
        </h4>
        <h5>{{ wine.review_set.count }} reviews</h5>
        <h5>{{ wine.average_rating | floatformat }}
            average rating</h5>
    </div>
    {% endfor %}
</div>
{% else %}
    <p>No wines are available.</p>
{% endif %}
```

- winerama\reviews\templates\reviews\wine_detail.html

```
<h2>{{ wine.name }}</h2>
<h5>{{ wine.review_set.count }} reviews
({{ wine.average_rating | floatformat }} average rating)</h5>
<h3>Recent reviews</h3>
{% if wine.review_set.all %}
<div>
    {% for review in wine.review_set.all %}
    <div>
        <em>{{ review.comment }}</em>
        <h6>Rated {{ review.rating }} of 5 by
        {{ review.user_name }}</h6>
        <h5>
        <a href="{% url 'reviews:review_detail' review.id %}">
            Read more
        </a>
        </h5>
    </div>
    {% endfor %}
</div>
{% else %}
<p>No reviews for this wine yet</p>
{% endif %}
```

- reviews\urls.py

```
...
path('wine/', views.wine_list, name='wine_list'),
path('wine/<int:wine_id>/', views.wine_detail,
name='wine_detail'),
```

- winerama\reviews\templates\reviews\review_detail.html

```
<h2>
<a href="{% url 'reviews:wine_detail' review.wine.id %}">
{{ review.wine.name }}</a>
</h2>

<h4>
Rated {{ review.rating }} of 5 by {{ review.user_name }}
</h4>
<p>{{ review.pub_date }}</p>
<p>{{ review.comment }}</p>` ``
```

form

- winerama\reviews\forms.py

```
from django.forms import ModelForm, Textarea
from reviews.models import Review

class ReviewForm(ModelForm):
    class Meta:
        model = Review
        fields = ['user_name', 'rating', 'comment']
        widgets = {
            'comment': Textarea(attrs={'cols': 40, 'rows': 15}),
        }
```

- reviews\views.py

```
from django.urls import reverse_lazy
from django.http import HttpResponseRedirect
from .models import Review, Wine
from .forms import ReviewForm
import datetime
```

- reviews\views.py

```
def wine_detail(request, wine_id):
    form = ReviewForm()
    context = {'wine': wine, 'form': form}
    return render(request, 'reviews/wine_detail.html',
                  context)

def add_review(request, wine_id):
    wine = get_object_or_404(Wine, pk=wine_id)
    if request.POST:
        form = ReviewForm(request.POST)
    else:
        form = ReviewForm()
    if form.is_valid():
        user_name = request.user.username
        review = form.save(commit=False)
        review.wine = wine
        review.user_name = user_name
        review.pub_date = datetime.datetime.now()
        review.save()
        return HttpResponseRedirect(
            reverse_lazy('reviews:wine_detail', args=(wine.id,)))
    return render(request, 'reviews/wine_detail.html',
                  {'wine': wine, 'form': form})
```

- `reviews\urls.py`

```
path('wine/<int:wine_id>/add_review/', views.add_review,  
name='add_review'),
```


- winerama\reviews\templates\reviews\wine_detail.html

```
...
<h3>Add your review</h3>
{% if error_message %}
<p><strong>{{ error_message }}</strong></p>
{% endif %}

<form action="{% url 'reviews:add_review' wine.id %}"
method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit" value="Add" />
</form>
```

Review & Break Time!

- Python 설치
- 가상환경 설치
- pip를 사용해서 라이브러리 설치
- Django 프로젝트 생성
- Django 앱 추가
- Model -> View -> Templates

Bootstrap3

```
(venv) pip install django-bootstrap3
```

```
INSTALLED_APPS = [  
    ...  
    'bootstrap3'  
]
```

- winerama\winerama\reviews\templates\base.html

```
{% load bootstrap3 %}
{% bootstrap_css %}{% bootstrap_javascript %}
{% block bootstrap3_content %}
<div class="container">
  <nav class="navbar navbar-default">
    <div class="navbar-header">
      <a class="navbar-brand"
        href="{% url 'reviews:review_list' %}">
        Winerama</a>
    </div>
    <div id="navbar" class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li><a href="{% url 'reviews:wine_list' %}">
        Wine list</a></li>
        <li><a href="{% url 'reviews:review_list' %}">
        Home</a></li></ul>
    </div>
  </nav>
  <h1>{% block title %}(no title){% endblock %}</h1>
  {% bootstrap_messages %}
  {% block content %}(no content){% endblock %}
</div>
{% endblock %}
```

- winerama\winerama\reviews\templates\review_detail.html

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

{% block title %}
<h2><a href="{% url 'reviews:wine_detail' review.wine.id %}">{{ review.wine.name }}
{% endblock %}

{% block content %}
<h4>Rated {{ review.rating }} of 5 by {{ review.user_name }}</h4>
<p>{{ review.pub_date }}</p>
<p>{{ review.comment }}</p>
{% endblock %}
```

- winerama\winerama\reviews\templates\review_list.html

```
{% extends 'base.html' %}
{% block title %}
<h2>Latest reviews</h2>
{% endblock %}
{% block content %}
{% if latest_review_list %}
<div class="row">
  {% for review in latest_review_list %}
    <div class="col-xs-6 col-lg-4">
      <h4>
        <a href="{% url 'reviews:review_detail' review.id %}">
          {{ review.wine.name }}
        </a></h4>
        <h6>rated {{ review.rating }} of 5 by
          {{ review.user_name }}</h6>
        <p>{{ review.comment }}</p>
      </div>
    {% endfor %}
  </div>
{% else %}
<p>No reviews are available.</p>
{% endif %}
{% endblock %}
```

- winerama\winerama\reviews\templates\wine_detail.html

```
{% extends 'base.html' %}
{% load bootstrap3 %}
{% block title %}
<h2>{{ wine.name }}</h2>
<h5>{{ wine.review_set.count }} reviews
({{ wine.average_rating | floatformat }} average rating)</h5>
{% endblock %}
{% block content %}
<h3>Recent reviews</h3>
{% if wine.review_set.all %}
<div class="row">
  {% for review in wine.review_set.all %}
    <div class="col-xs-6 col-lg-4">
      <em>{{ review.comment }}</em>
      <h6>Rated {{ review.rating }} of 5 by
      {{ review.user_name }}</h6>
      <h5>
        <a href="{% url 'reviews:review_detail' review.id %}">
          Read more</a></h5>
      </div>{% endfor %}
    </div>
  {% else %}
    <p>No reviews for this wine yet</p>{% endif %}
```

- winerama\winerama\reviews\templates\wine_detail.html

```
<h3>Add your review</h3>
{% if error_message %}<p><strong>
{{ error_message }}</strong></p>{% endif %}

<form action="{% url 'reviews:add_review' wine.id %}"
method="post" class="form">
    {% csrf_token %}
    {% for field in form %}
        {% bootstrap_field field %}
    {% endfor %}
    {% buttons %}
        <button type="submit" class="btn btn-primary">
            {% bootstrap_icon "star" %} Add
        </button>
    {% endbuttons %}
</form>
{% endblock %}
```


- winerama\winerama\reviews\templates\wine_list.html

```
{% extends 'base.html' %}
{% block title %}
<h2>Wine list</h2>
{% endblock %}
{% block content %}
{% if wine_list %}
<div class="row">
    {% for wine in wine_list %}
    <div class="col-xs-6 col-lg-4">
        <h4>
            <a href="{% url 'reviews:wine_detail' wine.id %}">
                {{ wine.name }}
            </a></h4>
            <h5>{{ wine.review_set.count }} reviews</h5>
            <h5>{{ wine.average_rating | floatformat }}
                average rating</h5>
        </div>
    {% endfor %}
</div>
{% else %}<p>No wines are available.</p>{% endif %}
{% endblock %}
```

Login

- `winerama\url.py`

```
path('accounts/', include(('django.contrib.auth.urls', 'auth'), namespace='auth'))
```

- reviews\views.py

```
from django.contrib.auth.decorators import login_required

@login_required
def add_review(request, wine_id):
    """
    rating = form.cleaned_data['rating']
    comment = form.cleaned_data['comment']
    user_name = request.user.username
    review = Review()
    review.wine = wine
    review.user_name = user_name
    review.rating = rating
    review.comment = comment
    review.pub_date = datetime.datetime.now()
    review.save()
```

- winerama\setting.py

```
'''  
'DIRS': [os.path.join(BASE_DIR, 'templates')],
```

- templates\registration\login.html

```
{% extends 'base.html' %}
{% load bootstrap3 %}

{% block title %}
<h2>Login</h2>
{% endblock %}

{% block content %}
<form action="{% url 'auth:login' %}"
method="post" class="form">
    {% csrf_token %}
    {% for field in form %}
        {% bootstrap_field field %}
    {% endfor %}
    {% buttons %}
        <button type="submit" class="btn btn-primary">
            {% bootstrap_icon "user" %} Login
        </button>
    {% endbuttons %}
</form>
{% endblock %}
```

- templates\base.html
 - reviews\templates\base.html 을 이동

```
<ul class="nav navbar-nav navbar-right">
{% if user.is_authenticated %}
  <li><a href="{% url 'auth:logout' %}">Logout</a></li>
{% else %}
  <li><a href="{% url 'auth:login' %}">Login</a></li>
{% endif %}
</ul>
```

User Review Page

- `setting.py`

```
LOGIN_REDIRECT_URL = '/reviews/review/user'
```

- `reviews\urls.py`

```
path('review/user/', views.user_review_list,  
name='user_review_list'),  
path('review/user/', views.user_review_list, name='user_review_list'),
```

- reviews\templates\reviews\user_review_list.html

```
{% extends 'reviews/review_list.html' %}

{% block title %}
<h2>Reviews by {{ username }}</h2>
{% endblock %}
```

- templates/base.html

```
<li><a href="{% url 'reviews:user_review_list' user.username %}">Hello {{ user.us
```


- reviews\views.py

```
def user_review_list(request, username=None):
    if not username:
        username = request.user.username
    latest_review_list = Review.objects
        .filter(user_name=username)
        .order_by('-pub_date')
    context = {'latest_review_list':latest_review_list,
        'username':username}
    return render(request, 'reviews/user_review_list.html',
        context)
```

- winerama\winerama\reviews\templates\review_detail.html

```
<h4>Rated {{ review.rating }} of 5 by  
<a href="{% url 'reviews:user_review_list'  
review.user_name %}">  
{{ review.user_name }}</a></h4>
```

- winerama\winerama\reviews\templates\review_list.html

```
<h6>rated {{ review.rating }} of 5 by <a href="{% url  
'reviews:user_review_list' review.user_name %}" >  
{{ review.user_name }}</a></h6>
```

Registration

```
(venv) pip install django-registration-redux
```

```
INSTALLED_APPS = [  
    ...  
    'registration'  
]  
  
ACCOUNT_ACTIVATION_DAYS = 7  
REGISTRATION_AUTO_LOGIN = True
```

```
(venv) python manage.py makemigrations
```

```
path('accounts/', include('registration.backends.simple.urls')),
```