Django2.1 Tutorial Lab Documentation

Release latest

Jan 04, 2019

Contents

1 Simple is Power **3**

2 Terminal Prompt **5**

3 Table of Contents **7**

3.1 Writing your first Django app, part 1 . . . 7
3.2 Writing your first Django app, part 2 . . . 9
3.3 Writing your first Django app, part 3 . . . 14
3.4 Writing your first Django app, part 4 . . . 22
3.5 Writing your first Django app, part 5 . . . 26
3.6 Writing your first Django app, part 6 . . . 27
3.7 Want List . . . 29

Hosting:

- Read The Docs, Django2.1 Tutotrial Lab

The main purpose of this project is to practice Django 2.1 Getting started effectively.

Like Yoga's daily practice, I need a routine to awaken my body and mind. Same as programming, I need lab to maintain skills to invoke idea.

CHAPTER 1

Simple is Power

Repeat practice and find out your weakness.

You have your right to be an individual thinker.

CHAPTER 2

Terminal Prompt

Before activate virtual environment:

```
export PS1='$ '
```

After activate virtual environment:

```
export PS1='(venv)$ '
```

To determine which python:

```
which python
```

CHAPTER 3

Table of Contents

3.1 Writing your first Django app, part 1

- Django Getting started, part 1
- Read The Docs, part 1

3.1.1 1-1. Virtual Envrionment

Lab:

```
$ python3.6 -v venv venv
$ source venv/bin/activate
(venv)$ pip install django
(venv)$ pip freeze
```

Note: $ To check installed packages.

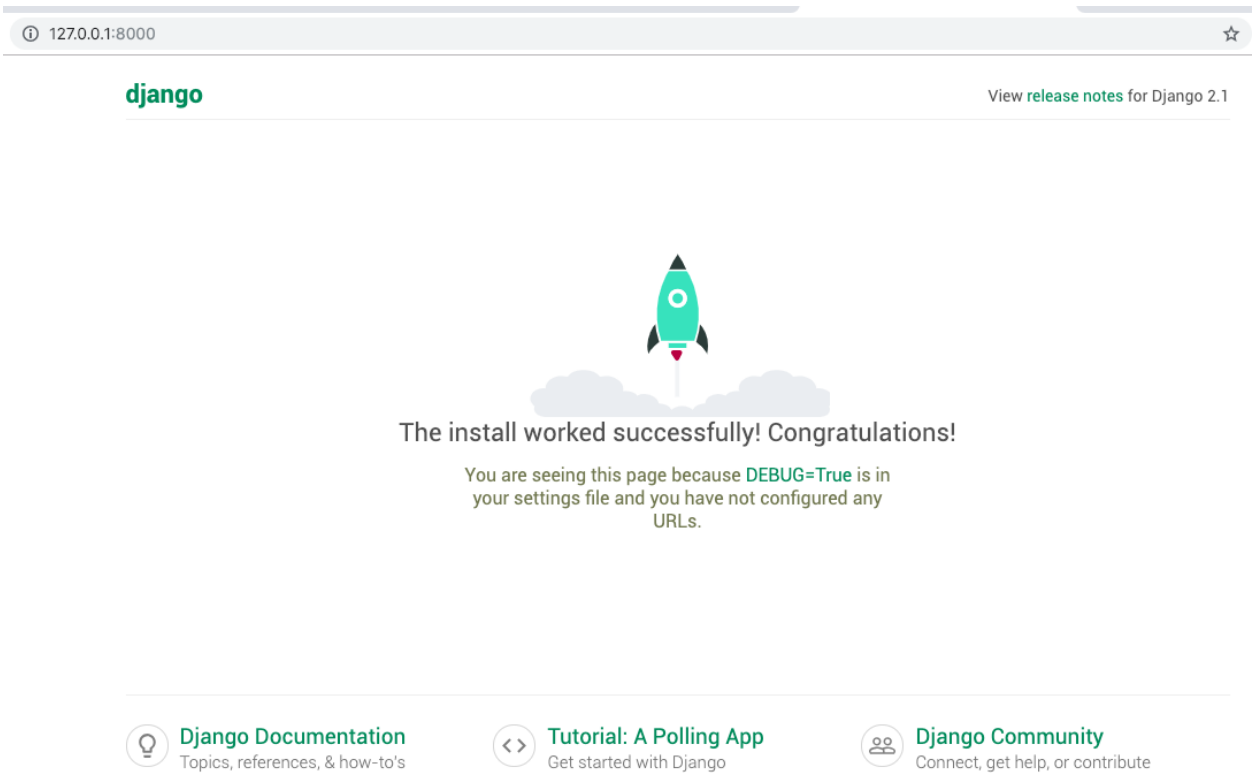
```
Django==2.1.4
pytz==2018.7
```

3.1.2 1-2. Start Project

Lab:

```
(venv)$ django-admin startproject mysite
(venv)$ cd mysite
(venv)$ python manage.py runserver
```

Note: To see a rocket!



3.1.3 1-3. Start App

Lab:

```
(venv)$ python manage.py startapp polls
*** edit mysite/urls.py
*** add polls/urls.py
*** edit polls/views.py
*** add go.py
(venv)$ . go
```

- mysite/urls.py:

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

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

- polls/urls.py:

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

urlpatterns = [
  path('', views.index, name='index'),
]
```

- polls/views.py:

```
from django.http import HttpResponse
def index(request):
    return HttpResponse("Hello, world. You're at the polls index.")
```

- go:

```
python manage.py runserver
```

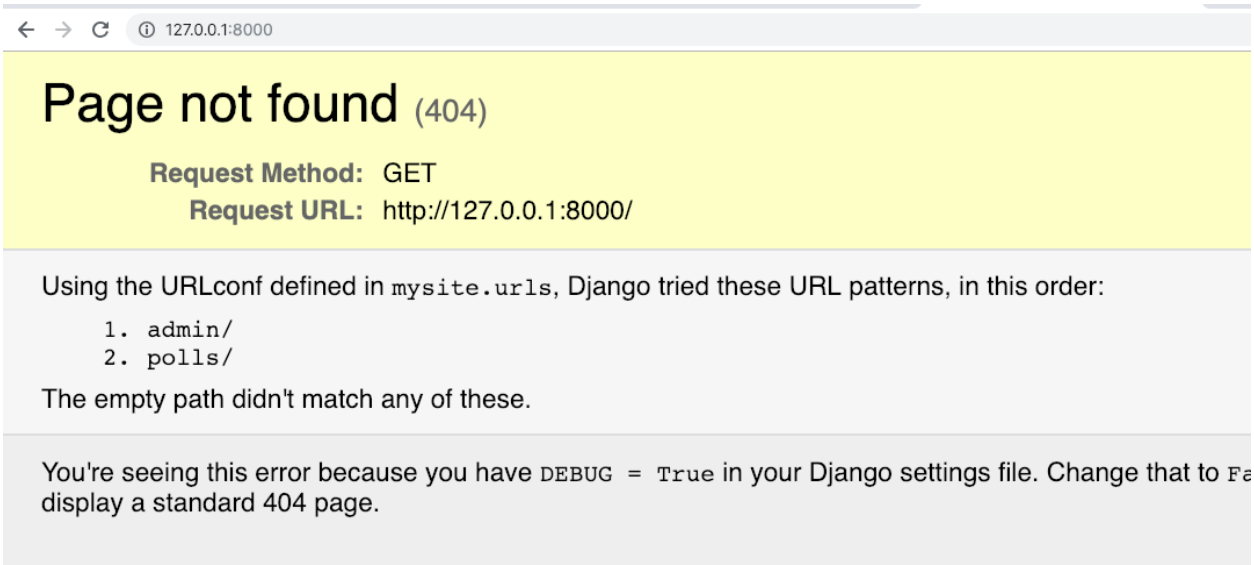
Note: To ensure http://127.0.0.1:8000/polls/ is working.

Hello, world. You're at the polls index.

Warning: Be aware http://127.0.0.1:8000 is damaged!

3.2 Writing your first Django app, part 2

- Django Getting started, part 2
- Writing custom django-admin commands
- Read The Docs, part 2

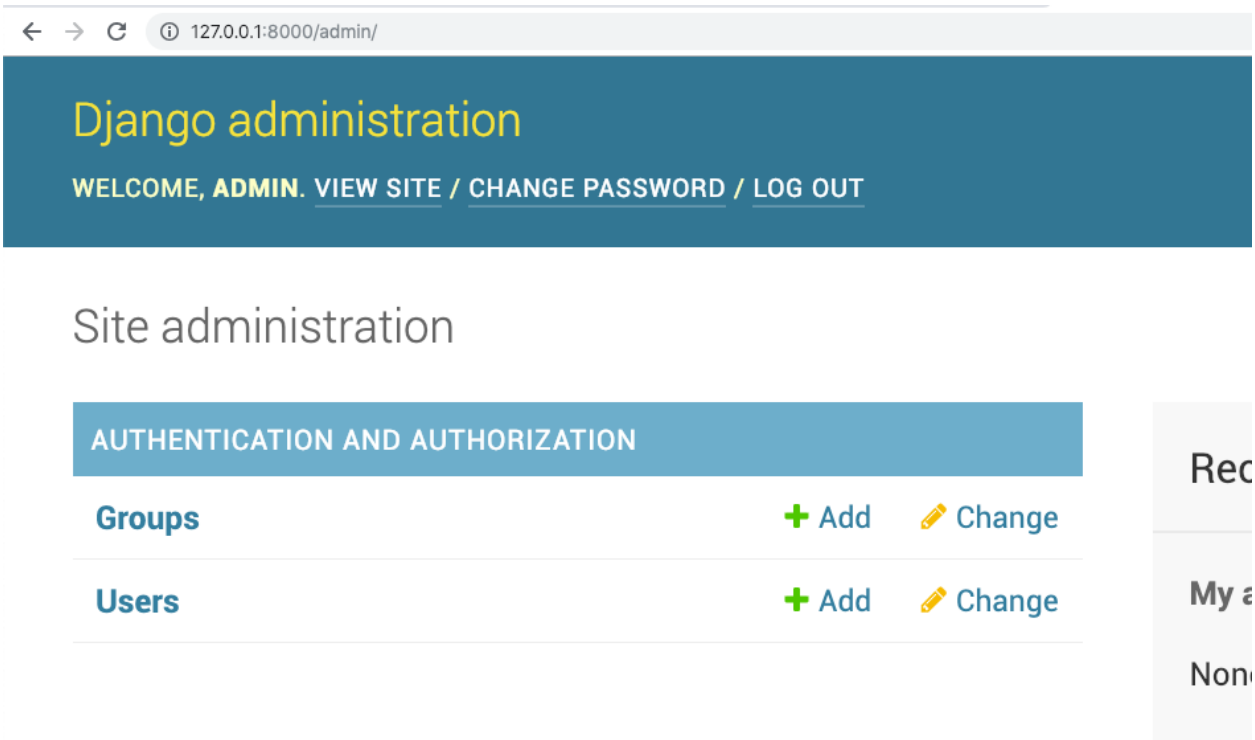


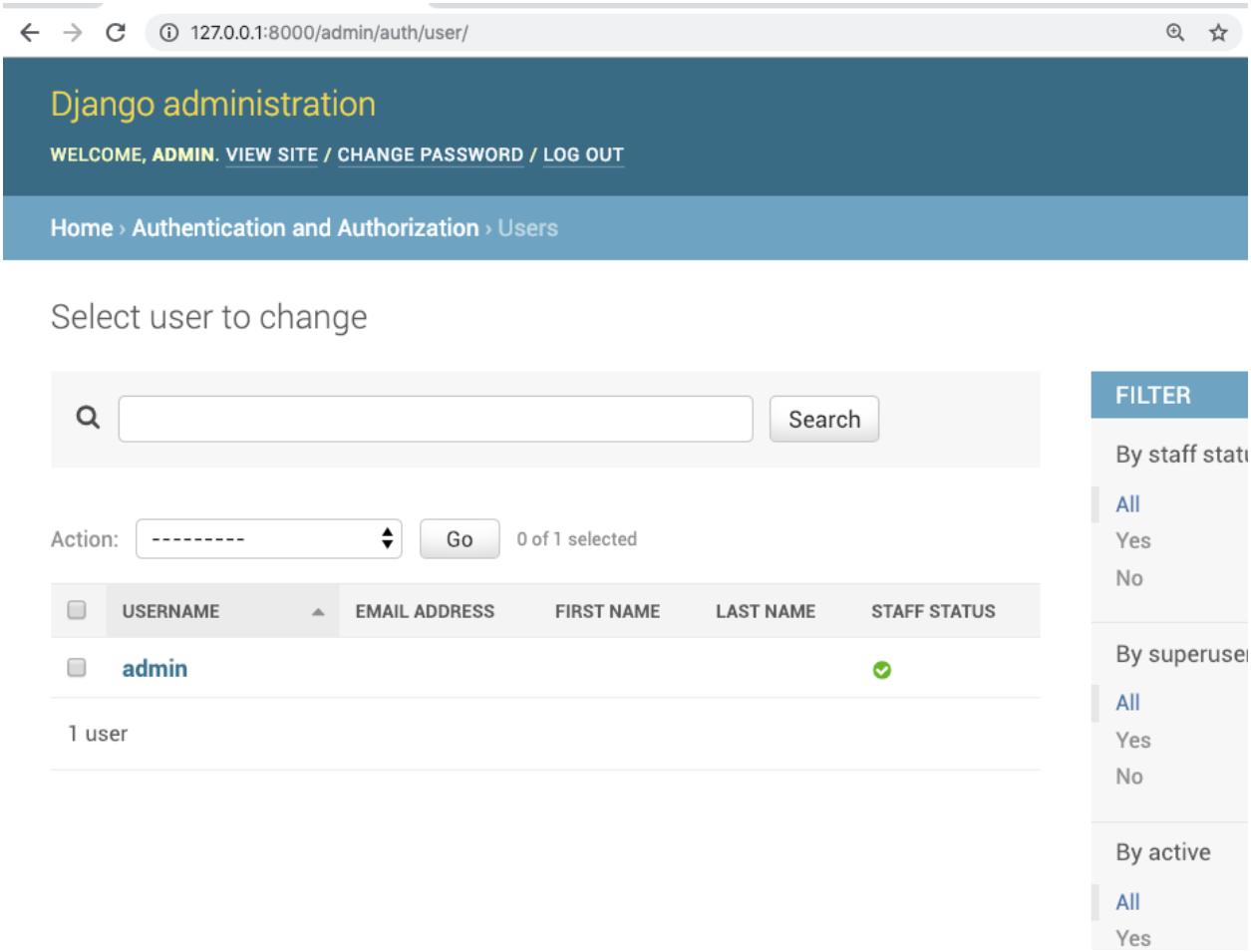
3.2.1 2-1. Admin

Lab:

```
(venv)$ python manage.py migrate
(venv)$ python manage.py createsuperuser
(venv)$ . go
```

Note: http://127.0.0.1:8000/admin/, login to maintain user/group.




127.0.0.1:8000/admin/auth/user/
Django administration
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Authentication and Authorization › Users
Select user to change
Search
Action:

Go
0 of 1 selected
USERNAME
EMAIL ADDRESS
FIRST NAME
LAST NAME
STAFF STATUS
admin
1 user
FILTER
By staff stat
All
Yes
No
By superuse
All
Yes
No
By active
All
Yes

3.2.2 2-2. Model

Lab:

```
*** edit mysite/settings.py
*** edit poll/models.py
*** edit poll/admin.py
*** edit go
. go
```

- mysite/settings.py:

```
INSTALLED_APPS = [
   'polls',
   'django.contrib.admin',
   ...
```

- polls/models.py:

```
from django.db import models


class Question(models.Model):
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')
    def __str__(self):
        return self.question_text

class Choice(models.Model):
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)
    def __str__(self):
        return self.choice_text
```

- polls/admin.py:

```
from django.contrib import admin
from .models import Question,Choice
admin.site.register(Question)
admin.site.register(Choice)
```

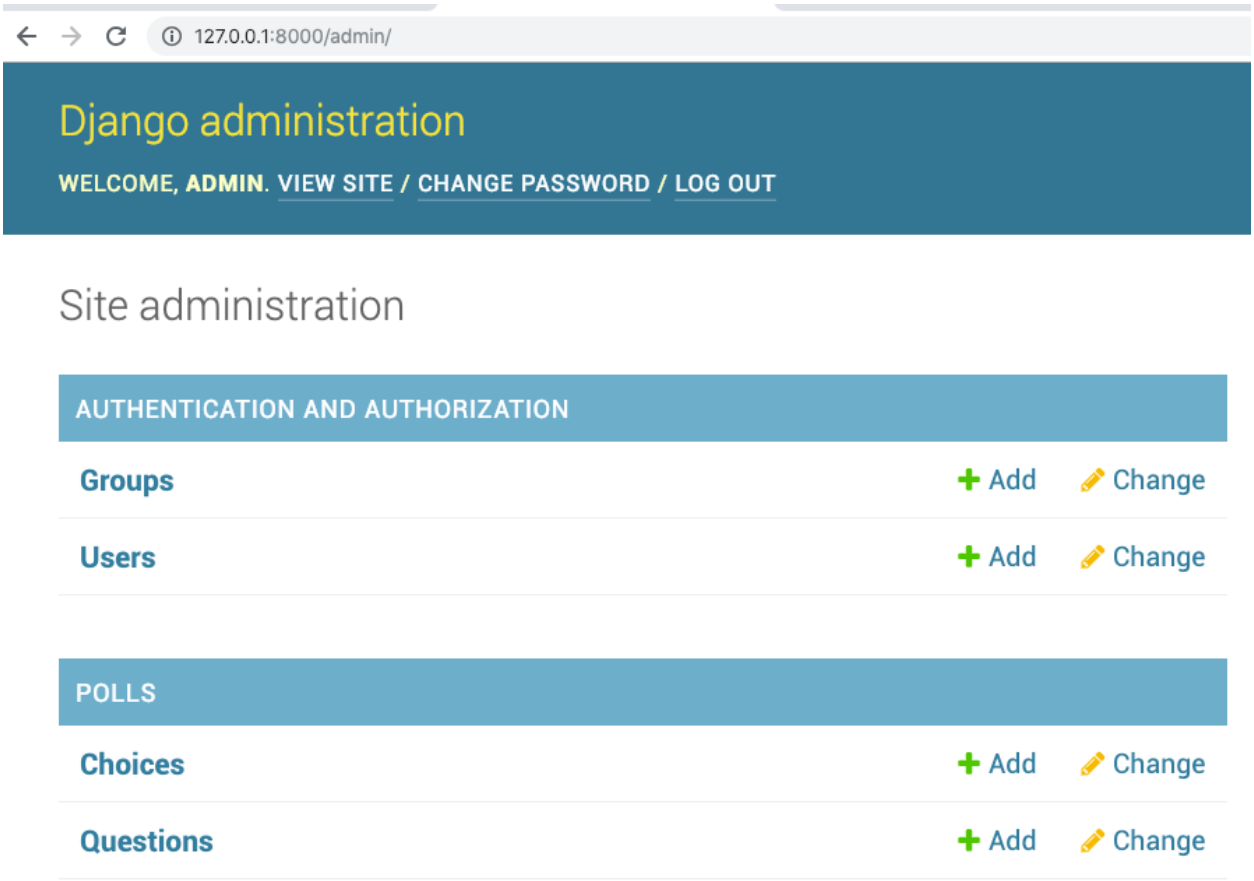
- go.py:

```
python manage.py makemigrations
python manage.py migrate
python manage.py runserver
```

Note: Able to maintain Question and Choice.

3.2.3 2-3. Command

Lab:


127.0.0.1:8000/admin/
Django administration
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Site administration
AUTHENTICATION AND AUTHORIZATION
Groups
Add
Change
Users
Add
Change
POLLS
Choices
Add
Change
Questions
Add
Change

```
*** add polls/management/commands/initpolls.py
python manage.py
python manage.py initpolls
. go
```

- polls/management/commands/initpolls.py:

```
from django.core.management.base import BaseCommand, CommandError
from polls.models import Question,Choice
from django.utils import timezone

class Command(BaseCommand):
    help = 'Create sample questions and choices.'

    def add_arguments(self, parser):
        parser.add_argument('question_num',type=int)

    def handle(self, *args, **options):
        q = Question.objects.all()
        q.delete()

        cnt = 0
        question_num = options['question_num']
        while (cnt < question_num):
            cnt += 1
            if cnt > 12:
                self.stdout.write(self.style.WARNING('Max number was set to 12'))
                break
            q = Question(question_text="Question #"+str(cnt),pub_date=timezone.
↪now())
            q.save()
            q.choice_set.create(choice_text='Choice A for Question #'+str(cnt),
↪votes=0)
            q.choice_set.create(choice_text='Choice B for Question #'+str(cnt),
↪votes=0)
            q.choice_set.create(choice_text='Choice C for Question #'+str(cnt),
↪votes=0)

        q = Question.objects.all()
        c = Choice.objects.all()

        self.stdout.write(self.style.SUCCESS('Questions "%s"' % q))
        self.stdout.write(self.style.SUCCESS('Choices "%s"' % c))
```

3.3 Writing your first Django app, part 3

References:

- Django Getting started, part 3
- Django Grils Tutorial, STATIC_ROOT
- Django , Django import / export

Hosting:

- Read The Docs, part 3

```
(venv)$ python manage.py help

Type 'manage.py help <subcommand>' for help on a specific subcommand.

Available subcommands:

[auth]
    changepassword
    createsuperuser

[contenttypes]
    remove_stale_contenttypes

[django]
    check
    compilemessages
    createcachetable
    dbshell
    diffsettings
    dumpdata
    flush
    inspectdb
    loaddata
    makemessages
    makemigrations
    migrate
    sendtestemail
    shell
    showmigrations
    sqlflush
    sqlmigrate
    sqlsequencereset
    squashmigrations
    startapp
    startproject
    test
    testserver

[polls]
    initpolls
```

```
(venv)$ python manage.py initpolls 99
Max number was set to 12
Questions "<QuerySet [<Question: Question #1>, <Question: Question #2>, <Question: Q
uestion #3>, <Question: Question #4>, <Question: Question #5>, <Question: Question #
6>, <Question: Question #7>, <Question: Question #8>, <Question: Question #9>, <Ques
tion: Question #10>, <Question: Question #11>, <Question: Question #12>]>"
Choices "<QuerySet [<Choice: Choice A for Question #1>, <Choice: Choice B for Questi
on #1>, <Choice: Choice C for Question #1>, <Choice: Choice A for Question #2>, <Cho
ice: Choice B for Question #2>, <Choice: Choice C for Question #2>, <Choice: Choice
A for Question #3>, <Choice: Choice B for Question #3>, <Choice: Choice C for Questi
on #3>, <Choice: Choice A for Question #4>, <Choice: Choice B for Question #4>, <Cho
ice: Choice C for Question #4>, <Choice: Choice A for Question #5>, <Choice: Choice
B for Question #5>, <Choice: Choice C for Question #5>, <Choice: Choice A for Questi
on #6>, <Choice: Choice B for Question #6>, <Choice: Choice C for Question #6>, <Cho
ice: Choice A for Question #7>, <Choice: Choice B for Question #7>, '...(remaining e
lements truncated)...']>"
```

3.3.1 3-1. Polls Index

Lab:

```
*** edit polls/views.py
*** add polls/templates/polls/index.html
```

- polls/views.py:

```
from django.shortcuts import render
from .models import Question

def index(request):
    latest_question_list = Question.objects.order_by('-pub_date')[:5]
    context = {'latest_question_list': latest_question_list}
    return render(request, 'polls/index.html', context)
```

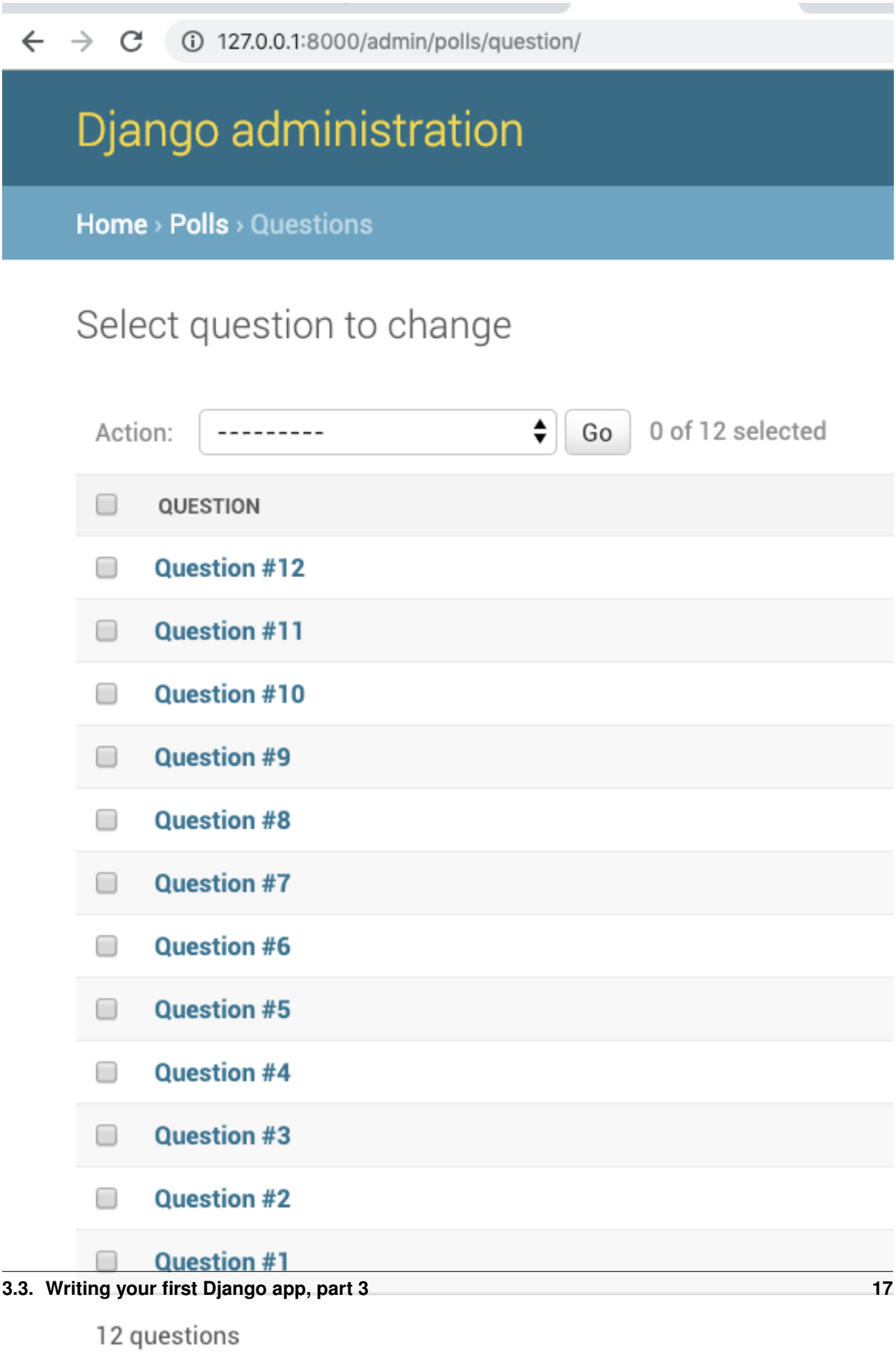
- polls/templates/polls/index.html:

```
{% if latest_question_list %}
  <ul>
  {% for question in latest_question_list %}
    <li><a href="/polls/{{ question.id }}/">{{ question.question_text }}</a></li>
  {% endfor %}
  </ul>
{% else %}
  <p>No polls are available.</p>
{% endif %}
```

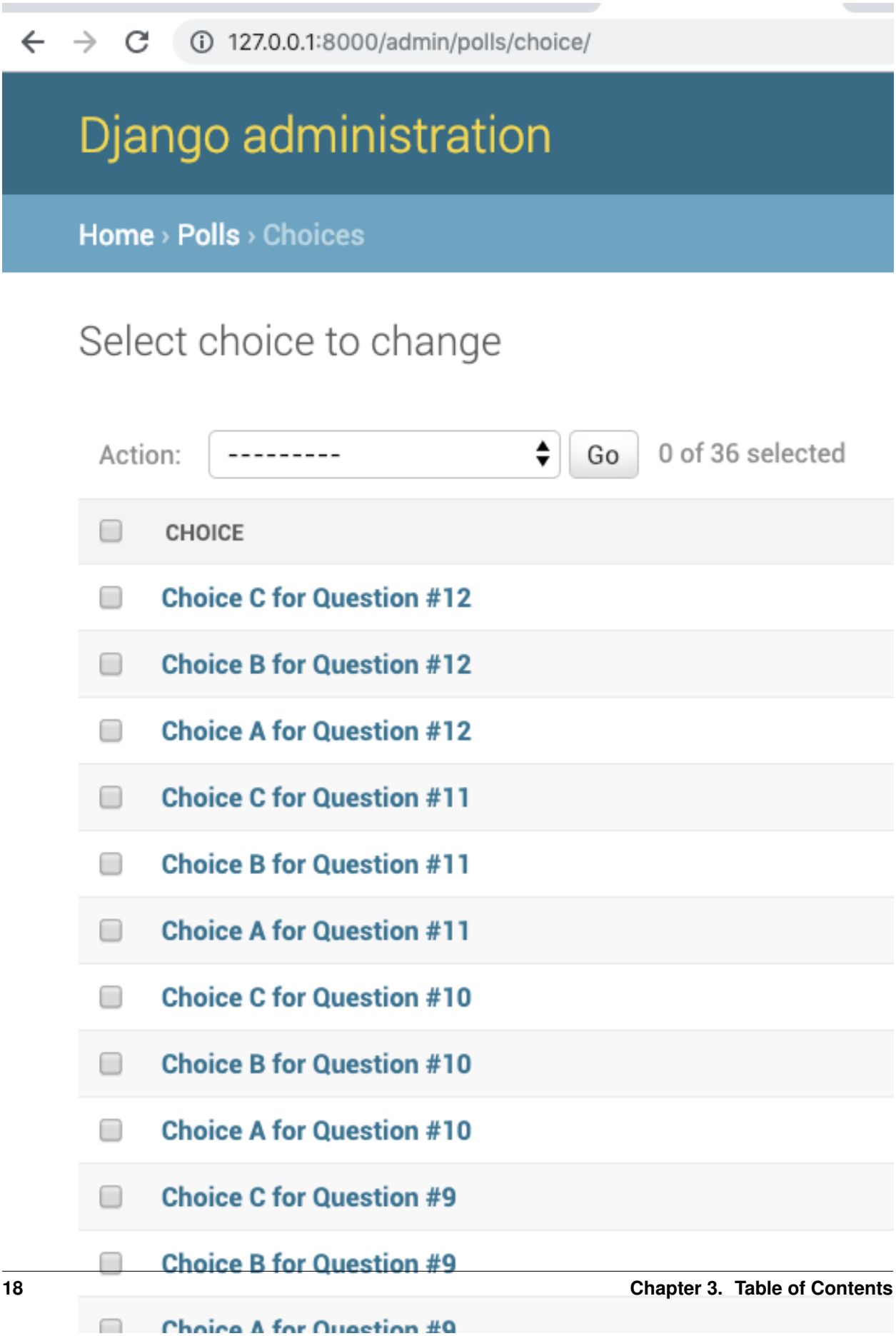
Note: Show questions on our polls page.

3.3.2 3-2. Polls Detail

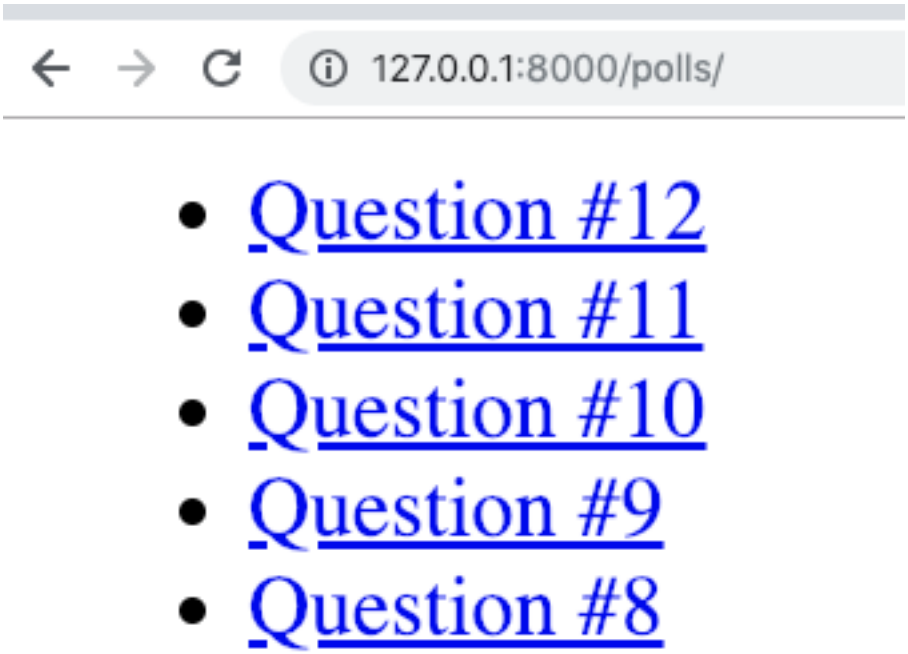
Lab:


127.0.0.1:8000/admin/polls/question/
Django administration
Home › Polls › Questions
Select question to change
Action:

Go
0 of 12 selected
QUESTION
Question #12
Question #11
Question #10
Question #9
Question #8
Question #7
Question #6
Question #5
Question #4
Question #3
Question #2
Question #1
12 questions


127.0.0.1:8000/admin/polls/choice/
Django administration
Home › Polls › Choices
Select choice to change
Action:

Go
0 of 36 selected
CHOICE
Choice C for Question #12
Choice B for Question #12
Choice A for Question #12
Choice C for Question #11
Choice B for Question #11
Choice A for Question #11
Choice C for Question #10
Choice B for Question #10
Choice A for Question #10
Choice C for Question #9
Choice B for Question #9




```
*** edit polls/urls.py
*** edit polls/models.py
*** edit polls/views.py
*** edit polls/templates/polls/index.html
*** add polls/templates/polls/detail.html
. go
```

- polls/urls.py:

```
from django.urls import path
from . import views
app_name = 'polls'

urlpatterns = [
  path('', views.index, name='index'),
  path('<int:question_id>/', views.detail, name='detail')
]
```

- polls/models.py:

```
from django.db import models

class Question(models.Model):
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')
    def __str__(self):
        return self.question_text
```

(continues on next page)

(continued from previous page)

```
class Choice(models.Model):
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)
    def __str__(self):
        return self.question.question_text+" "+self.choice_text
```

- polls/views.py:

```
from django.shortcuts import render,get_object_or_404
from .models import Question

def index(request):
    latest_question_list = Question.objects.order_by('-pub_date')[:5]
    context = {'latest_question_list': latest_question_list}
    return render(request, 'polls/index.html', context)

def detail(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    return render(request, 'polls/detail.html', {'question': question})
```

- polls/templates/polls/index.html:

```
{% if latest_question_list %}
  <ul>
  {% for question in latest_question_list %}
    <li><a href="{% url 'polls:detail' question.id %}">{{ question.question_text }
↪}</a></li>
  {% endfor %}
  </ul>
{% else %}
  <p>No polls are available.</p>
{% endif %}
```

- polls/templates/polls/detail.html:

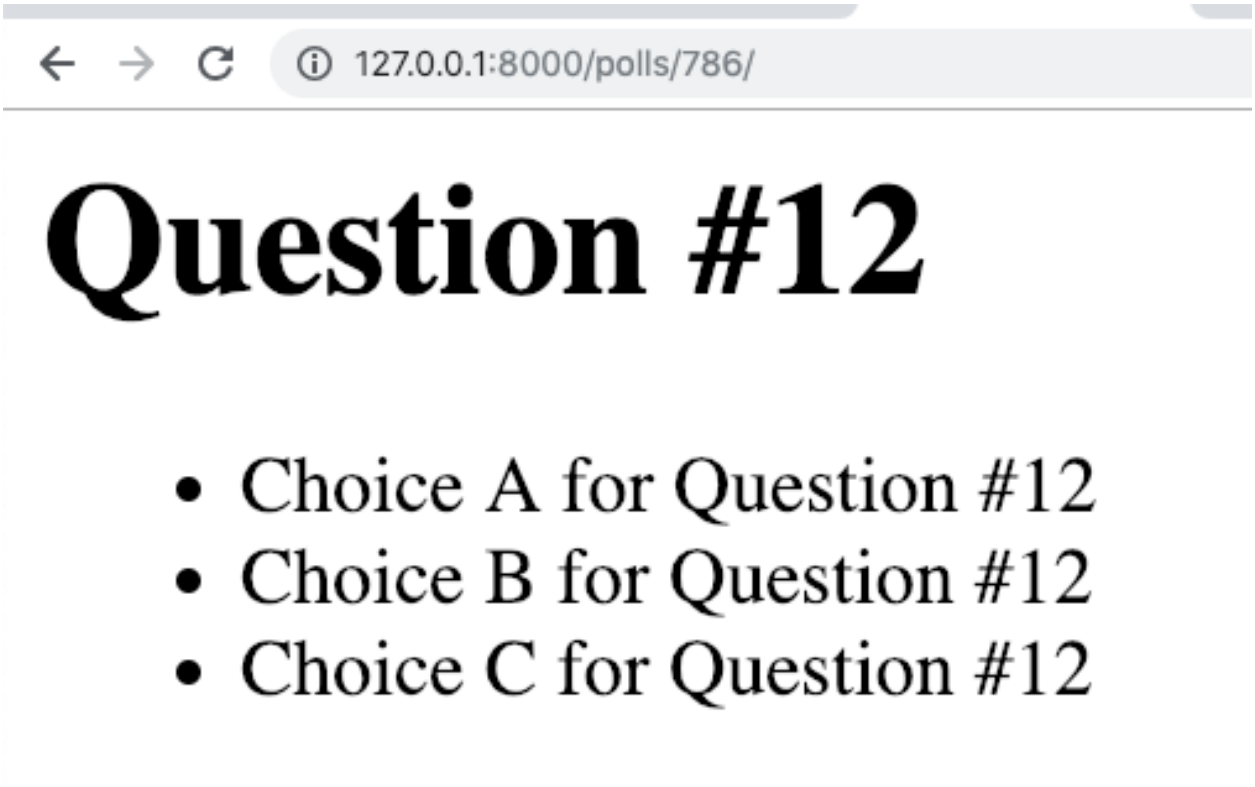
```
<h1>{{ question.question_text }}</h1>
<ul>
{% for choice in question.choice_set.all %}
    <li>{{ choice.choice_text }}</li>
{% endfor %}
</ul>
```

3.3.3 3-3. Django import / export

Lab:

```
pip install django-import-export
pip freeze
*** edit mysite/setting.py
*** edit polls/admin.py
. go
```

- mysite/setting.py:




```
INSTALLED_APPS = (
  'import_export',
  'polls',
  ...

...
STATIC_URL = '/static/'
# https://tutorial.djangogirls.org/en/django_start_project/
STATIC_ROOT = os.path.join(BASE_DIR, 'static')
```

- polls/admin.py:

```
from django.contrib import admin

from import_export import resources
from import_export.admin import ImportExportModelAdmin

from .models import Question,Choice


class QuestionResource(resources.ModelResource):
    class Meta:
        model = Question

class QuestionAdmin(ImportExportModelAdmin):
    resource_class = QuestionResource

admin.site.register(Question,QuestionAdmin)
```

(continues on next page)

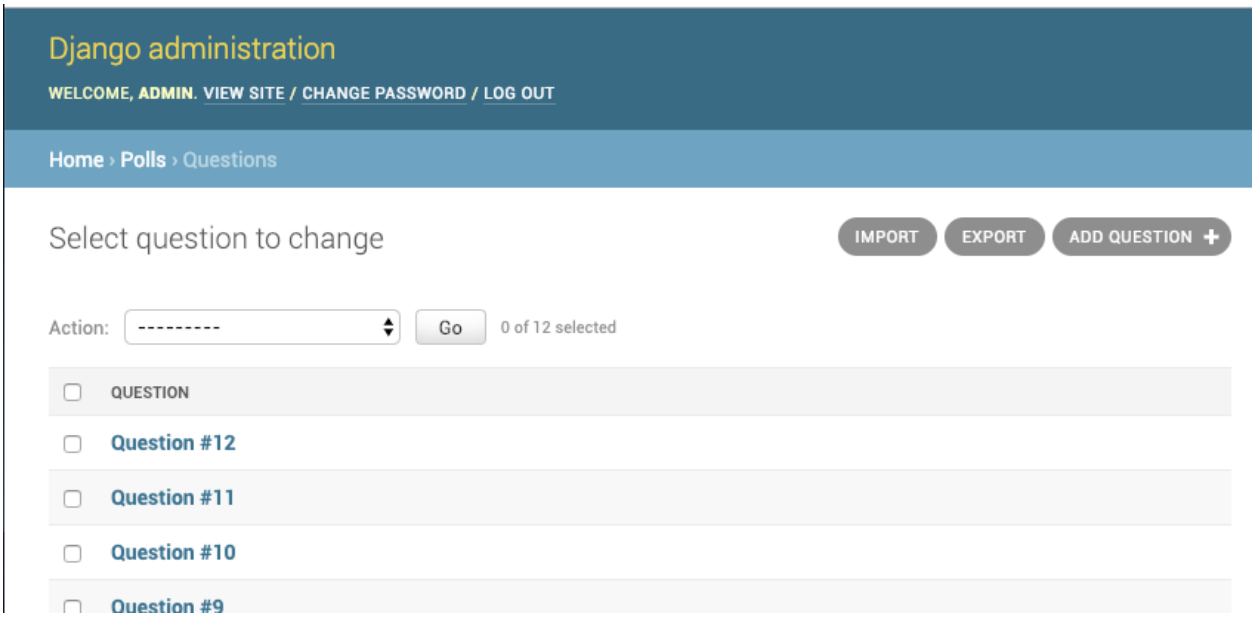
(continued from previous page)

```
class ChoiceResource(resources.ModelResource):
    class Meta:
        model = Choice

class ChoiceAdmin(ImportExportModelAdmin):
    resource_class = ChoiceResource

admin.site.register(Choice,ChoiceAdmin)
```

```
defusedxml==0.5.0
diff-match-patch==20181111
Django==2.1.4
django-import-export==1.1.0
et-xmlfile==1.0.1
jdcal==1.4
odfpy==1.4.0
openpyxl==2.5.12
pytz==2018.7
PyYAML==3.13
tablib==0.12.1
unicodecsv==0.14.1
xlrd==1.2.0
xlwt==1.3.0
```

3.4 Writing your first Django app, part 4

- Django Getting started, part 4
- Read The Docs, part 4



3.4.1 4-1. Forms

Lab:

```
*** edit polls/urls.py
*** edit polls/views.py
*** edit polls/templates/polls/detail.html
*** add polls/templates/polls/vote.html
*** add polls/templates/polls/result.html
. go
```

- polls/urls.py:

```
from django.urls import path
from . import views
app_name = 'polls'

urlpatterns = [
  path('', views.index, name='index'),
  path('<int:question_id>/', views.detail, name='detail'),
  path('<int:question_id>/vote/', views.vote, name='vote'),
  path('<int:question_id>/results/', views.results, name='results'),
]
```

- polls/views.py:

```
from django.http import HttpResponse, HttpResponseRedirect
from django.shortcuts import get_object_or_404, render
from django.urls import reverse

from .models import Choice, Question

def index(request):
    latest_question_list = Question.objects.order_by('-pub_date')[:5]
    context = {'latest_question_list': latest_question_list}
```

(continues on next page)

(continued from previous page)

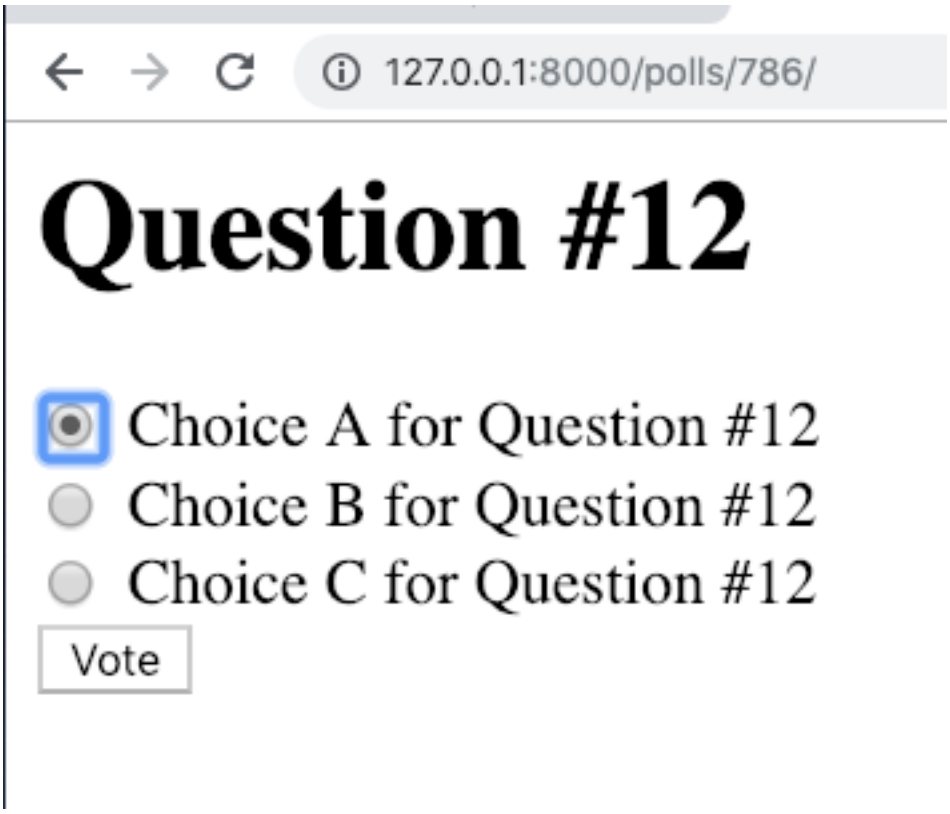
```
    return render(request, 'polls/index.html', context)

def detail(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    return render(request, 'polls/detail.html', {'question': question})

def vote(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    try:
        selected_choice = question.choice_set.get(pk=request.POST['choice'])
    except (KeyError, Choice.DoesNotExist):
        # Redisplay the question voting form.
        return render(request, 'polls/detail.html', {
            'question': question,
            'error_message': "You didn't select a choice.",
        })
    else:
        selected_choice.votes += 1
        selected_choice.save()
        # Always return an HttpResponseRedirect after successfully dealing
        # with POST data. This prevents data from being posted twice if a
        # user hits the Back button.
        return HttpResponseRedirect(reverse('polls:results', args=(question.id,)))

def results(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    return render(request, 'polls/results.html', {'question': question})
```

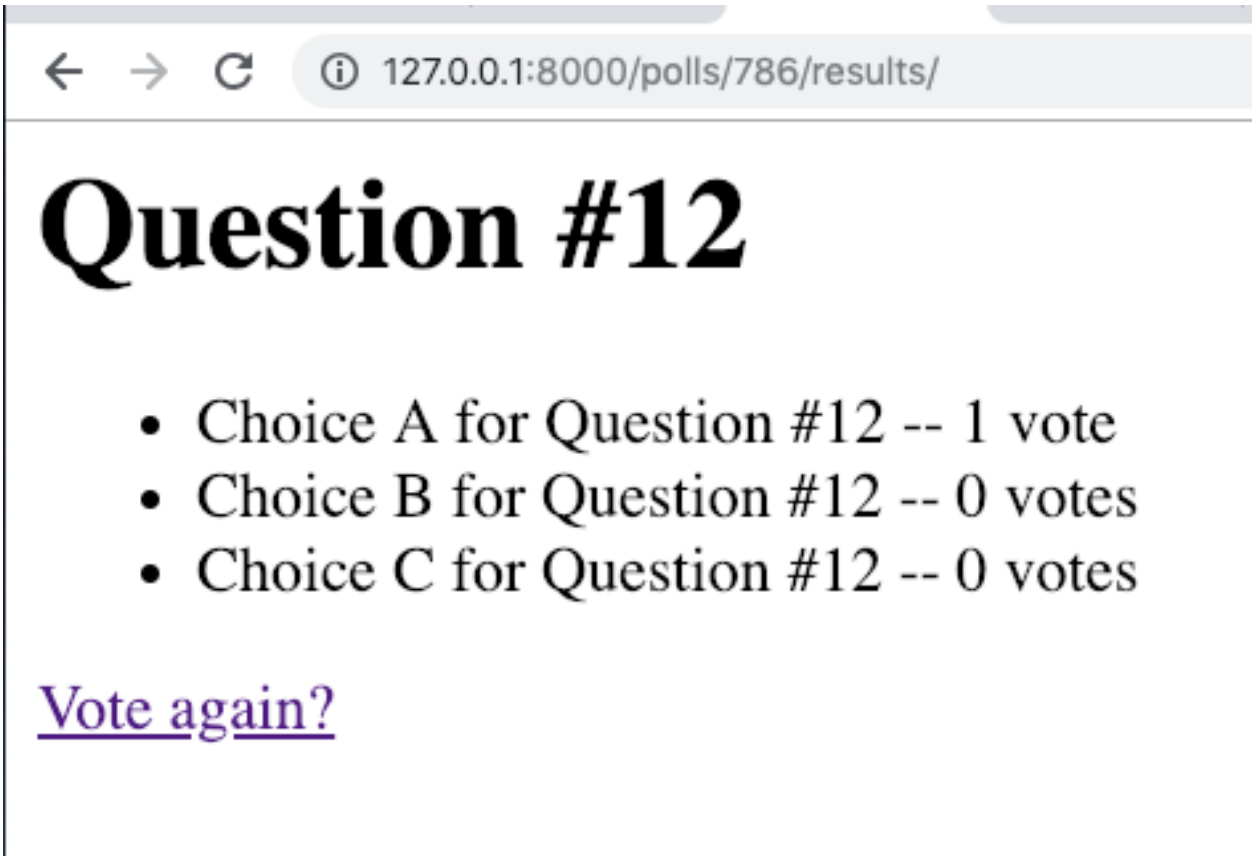
- polls/templates/polls/detail.html:

```
<h1>{{ question.question_text }}</h1>

{% if error_message %}<p><strong>{{ error_message }}</strong></p>{% endif %}

<form action="{% url 'polls:vote' question.id %}" method="post">
{% csrf_token %}
{% for choice in question.choice_set.all %}
    <input type="radio" name="choice" id="choice{{ forloop.counter }}" value="{{␣
↪choice.id }}">
    <label for="choice{{ forloop.counter }}">{{ choice.choice_text }}</label><br>
{% endfor %}
<input type="submit" value="Vote">
</form>
```

- polls/templates/polls/results.html:

```
<h1>{{ question.question_text }}</h1>

<ul>
{% for choice in question.choice_set.all %}
    <li>{{ choice.choice_text }} -- {{ choice.votes }} vote{{ choice.
↪votes|pluralize }}</li>
{% endfor %}
</ul>

<a href="{% url 'polls:detail' question.id %}">Vote again?</a>
```


127.0.0.1:8000/polls/786/
Question #12
Choice A for Question #12
Choice B for Question #12
Choice C for Question #12
Vote


127.0.0.1:8000/polls/786/results/
Question #12
Choice A for Question #12 -- 1 vote
Choice B for Question #12 -- 0 votes
Choice C for Question #12 -- 0 votes
Vote again?

Note: Able to vote

3.5 Writing your first Django app, part 5

- Django Getting started, part 5
- Read The Docs, part 5

3.5.1 5-1. Test

Lab:

```
*** edit mysite/urls.py
*** edit polls/tests.py
*** add polls/templates/index.html
(venv)$ python manage.py test
```

- polls/urls.py:

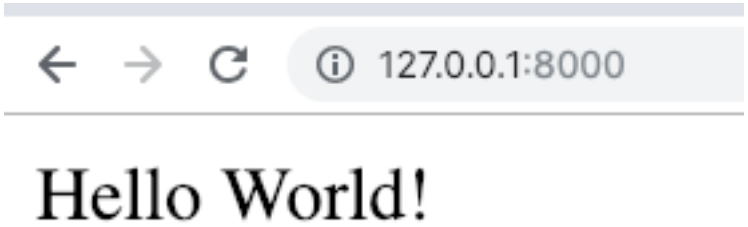
```
from django.contrib import admin
from django.urls import path,include
from django.views.generic import TemplateView

urlpatterns = [
    path('admin/', admin.site.urls),
    path('polls/', include('polls.urls')),
    path('', TemplateView.as_view(template_name="index.html"),name='index'),
]
```

- polls/tests.py:

```
from django.test import TestCase
from django.utils import timezone
from django.urls import reverse
import datetime

from .models import Question

class QuestionModelTests(TestCase):

    def test_was_published_recently_with_future_question(self):
        """
        was_published_recently() returns False for questions whose pub_date
        is in the future.
        """
        time = timezone.now() + datetime.timedelta(days=30)
        future_question = Question(pub_date=time)
        self.assertIs(future_question.was_published_recently(), False)

    def test_hello_world(self):
        response = self.client.get(reverse('index'))
        self.assertEqual(response.status_code, 200)
        self.assertContains(response, "Hello World!")
```

- polls/templates/index.html:

```
Hello World!
```




```
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..
----------------------------------------------------------------------
Ran 2 tests in 0.015s

OK
Destroying test database for alias 'default'...
```

Note: Fix / with 'Hello World!' using CBV.

3.6 Writing your first Django app, part 6

- Django Getting started, part 6
- Django Girls Tutorials, Template extending
- Boostrap v4.2, Starter template
- Read The Docs, part 6

3.6.1 6-1. Template Extending

Lab:

```
*** edit polls/static/polls/style.css
*** add polls/templates/base.html
*** edit polls/templates/index.html
*** edit polls/templates/detail.html
*** edit polls/templates/results.html
(venv)$ python manage.py test
```

- polls/static/polls/style.css:

```
body{
  margin-top: 12px;
}

li a {
    color: black;
}
```

- polls/templates/base.html:

```
{% load static %}
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
↪fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.
↪1/css/bootstrap.min.css" integrity="sha384-
↪GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS" crossorigin=
↪"anonymous">
    <link rel="stylesheet" type="text/css" href="{% static 'polls/style.css' %}">
    <title>Lab</title>
  </head>
  <body>
    <div class="container">
      <h1><a href='/polls/'>Django2.1 Tutotrial Lab</a></h1>
          <div class="row">
            <div class="col-md-8">
            {% block content %}
            {% endblock %}
            </div>
        </div>
    </div>
    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity=
↪"sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"␣
↪crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/
↪popper.min.js" integrity="sha384-wHAiFfRlMFy6i5SRaxvfOCifBUQy1xHdJ/
↪yoi7FRNXMRBu5WHdZYu1hA6ZOblgut" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.
↪min.js" integrity="sha384-B0UglyR+jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/
↪mkkaduKaBhlAXv9k" crossorigin="anonymous"></script>
  </body>
</html>
```

- polls/templates/index.html:

```
{% extends 'polls/base.html' %}
{% block content %}
```

(continues on next page)

(continued from previous page)

```
{% if latest_question_list %}
  <h3><ul>
  {% for question in latest_question_list %}
    <li><a href="{% url 'polls:detail' question.id %}">{{ question.question_text }
↪}</a></li>
  {% endfor %}
</ul></h3>
{% else %}
  <p>No polls are available.</p>
{% endif %}
{% endblock %}
```

- polls/templates/detail.html:

```
{% extends 'polls/base.html' %}
{% block content %}
<h3>{{ question.question_text }}</h3>

{% if error_message %}<p><strong>{{ error_message }}</strong></p>{% endif %}

<form action="{% url 'polls:vote' question.id %}" method="post">
{% csrf_token %}
{% for choice in question.choice_set.all %}
    <input type="radio" name="choice" id="choice{{ forloop.counter }}" value="{{␣
↪choice.id }}">
    <label for="choice{{ forloop.counter }}">{{ choice.choice_text }}</label><br>
{% endfor %}
<input class='btn btn-success' type="submit" value="Vote">
</form>
{% endblock %}
```

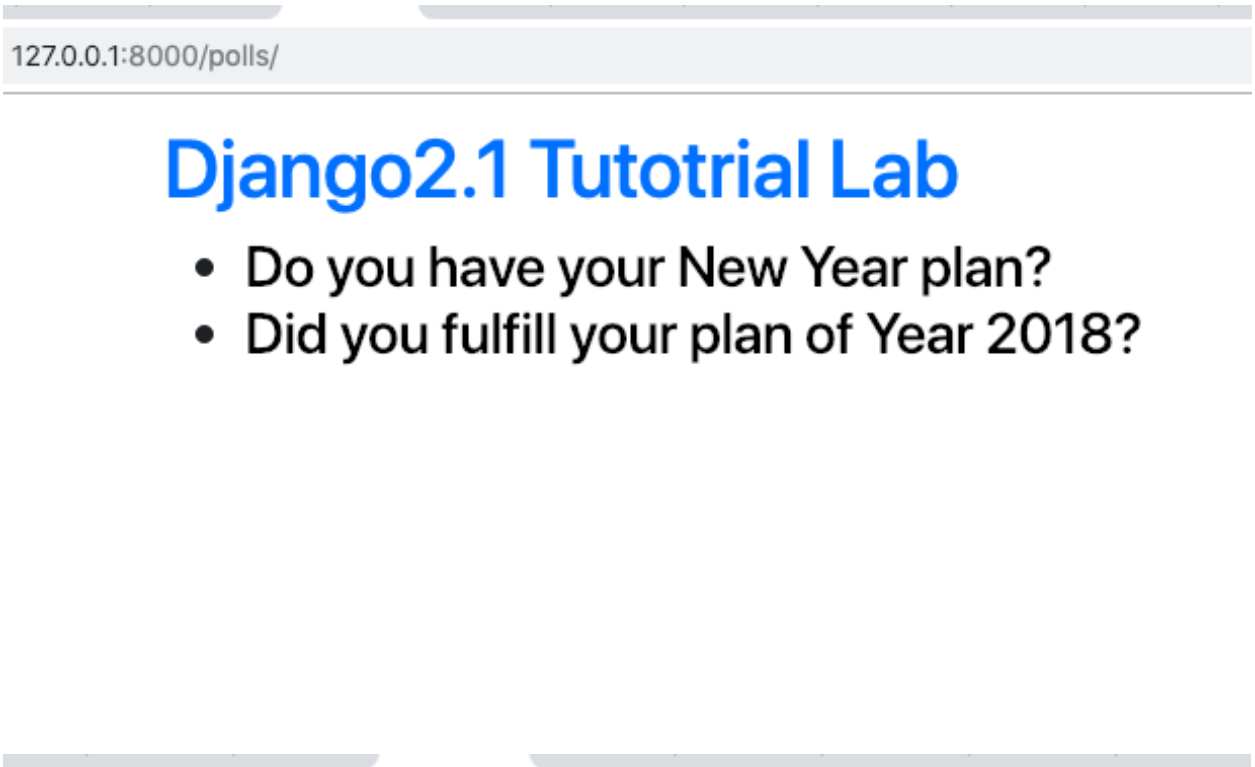
- polls/templates/results.html:

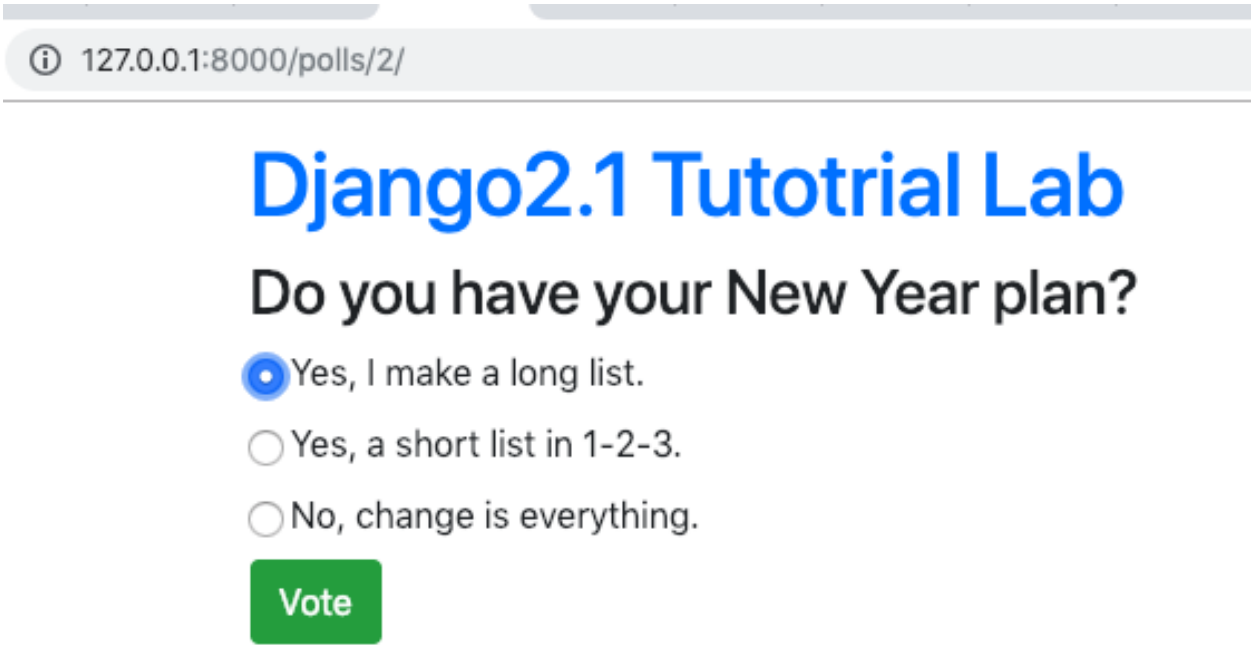
```
{% extends 'polls/base.html' %}
{% block content %}
<h2>{{ question.question_text }}</h2>

<ul>
{% for choice in question.choice_set.all %}
    <li>{{ choice.choice_text }} -- {{ choice.votes }} vote{{ choice.
↪votes|pluralize }}</li>
{% endfor %}
</ul>

<a class='btn btn-success' href="{% url 'polls:detail' question.id %}">Vote again?
↪</a>
{% endblock %}
```

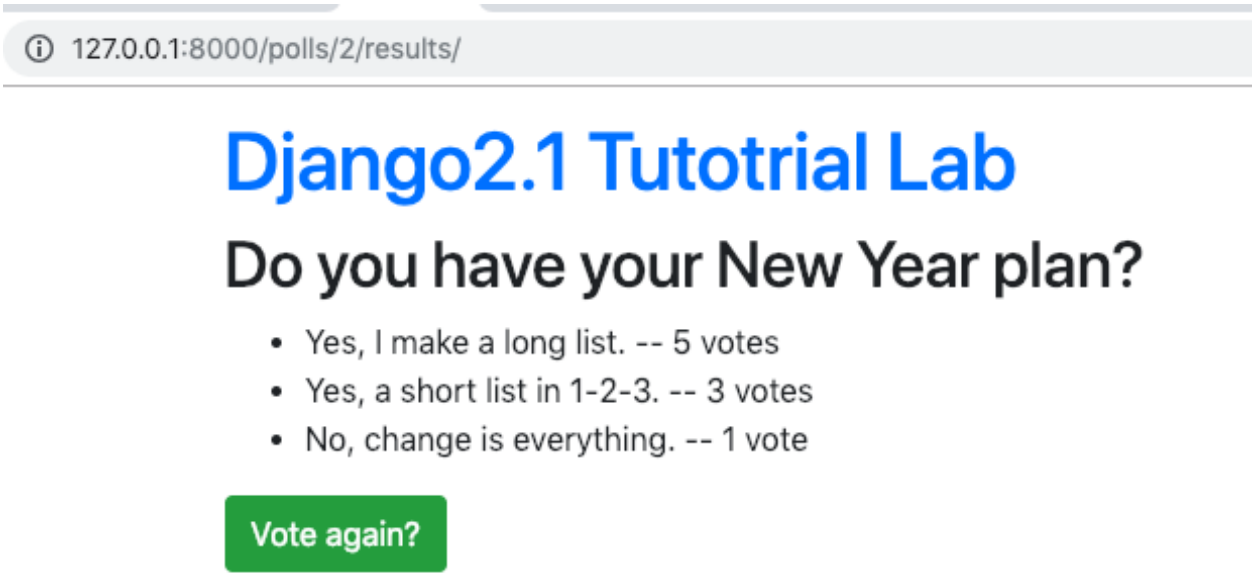
Warning: You might need to 'Clear Browsing Data' to let css working during development.

3.7 Want List

Hosting:


127.0.0.1:8000/polls/
Django2.1 Tutotrial Lab
Do you have your New Year plan?
Did you fulfill your plan of Year 2018?


127.0.0.1:8000/polls/2/
Django2.1 Tutotrial Lab
Do you have your New Year plan?
Yes, I make a long list.
Yes, a short list in 1-2-3.
No, change is everything.
Vote



- Read The Docs, Want List

3.7.1 Why?

From Django tutorial, we learned

> python manage.py startapp polls

Later on we still need to create urls.py for route. And, to be production ready, we also need to have templates/polls/base.html, as well as static/polls/style.css. Why not to provide a much ready templates for startapp?

Learn lower level python

Something we need to guess to assume first. Then to verify and to adjust.

My desired startapp

References:

- Django ref, django-admin and manage.py

python manage.py startapp

urls.py, templates/{{app_name}}/base.html,

What's standard startapp doing?

111

create the folder

https://docs.djangoproject.com/en/2.1/ref/django-admin/#startapp

startapp

`django-admin startapp name [directory]`

Creates a Django app directory structure for the given app name in the current directory or the given destination.

By default the directory created contains a **`models.py`** file and other app template files. (See the source for more details.) If only the app name is given, the app directory will be created in the current working directory.

If the optional destination is provided, Django will use that existing directory rather than creating a new one. You can use '.' to denote the current working directory.

For example:

```
django-admin startapp myapp /Users/jezdez/Code/myapp
```

GitHub, Inc. [US] | https://github.com/django/django/tree/master/django/conf/app_template

Search or jump to... / Pull requests Issues Marketplace Explore

django / django

Watch 2,147 Star 38,623 Fork 16,676

Code Pull requests 184 Insights

Branch: master | django / django / conf / app_template / | Create new file | Upload files | Find file | History

claudep Refs #23919 -- Removed encoding preambles and future imports — Latest commit d7b9aaa on Nov 20, 2016

..		
migrations	Fixed #25968 -- Changed project/app templates to use a "py-tpl" suffix.	3 years ago
__init__.py-tpl	Fixed #25968 -- Changed project/app templates to use a "py-tpl" suffix.	3 years ago
admin.py-tpl	Refs #23919 -- Removed encoding preambles and future imports	2 years ago
apps.py-tpl	Refs #23919 -- Removed encoding preambles and future imports	2 years ago
models.py-tpl	Refs #23919 -- Removed encoding preambles and future imports	2 years ago
tests.py-tpl	Refs #23919 -- Removed encoding preambles and future imports	2 years ago
views.py-tpl	Refs #23919 -- Removed encoding preambles and future imports	2 years ago

222

copy files

333

replace variable with app name

To understand Django

Prepare envrionment to study Django:

```
$ pwd
/Users/pinglingchen/ksndjs/django001/django
$ ls
django
$ cd django
$ git remote -v
origin      https://github.com/twoutlook/django.git (fetch)
origin      https://github.com/twoutlook/django.git (push)
$ cd ..
$ ls
django
$ python3.6 -m venv venv
$ . venv/bin/activate
(venv) $ pip freeze
(venv) $ pip install -e django
Obtaining file:///Users/pinglingchen/ksndjs/django001/django/django
Collecting pytz (from Django==2.2.dev20190102231945)
  Using cached https://files.pythonhosted.org/packages/f8/0e/
↪2365ddc010afb3d79147f1dd544e5ee24bf4ece58ab99b16fbb465ce6dc0/pytz-2018.7-py2.py3-
↪none-any.whl
Collecting sqlparse (from Django==2.2.dev20190102231945)
  Using cached https://files.pythonhosted.org/packages/65/85/
↪20bdd72f4537cf2c4d5d005368d502b2f464ede22982e724a82c86268eda/sqlparse-0.2.4-py2.py3-
↪none-any.whl
Installing collected packages: pytz, sqlparse, Django
  Running setup.py develop for Django
Successfully installed Django pytz-2018.7 sqlparse-0.2.4
You are using pip version 10.0.1, however version 18.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(venv) $ pip freeze
-e git+https://github.com/twoutlook/django.
↪git@b5fe97a34ea527d4254b58c2e828450e7c32157f#egg=Django
pytz==2018.7
sqlparse==0.2.4
You are using pip version 10.0.1, however version 18.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(venv) $
```

python manage.py startapp having urls.py

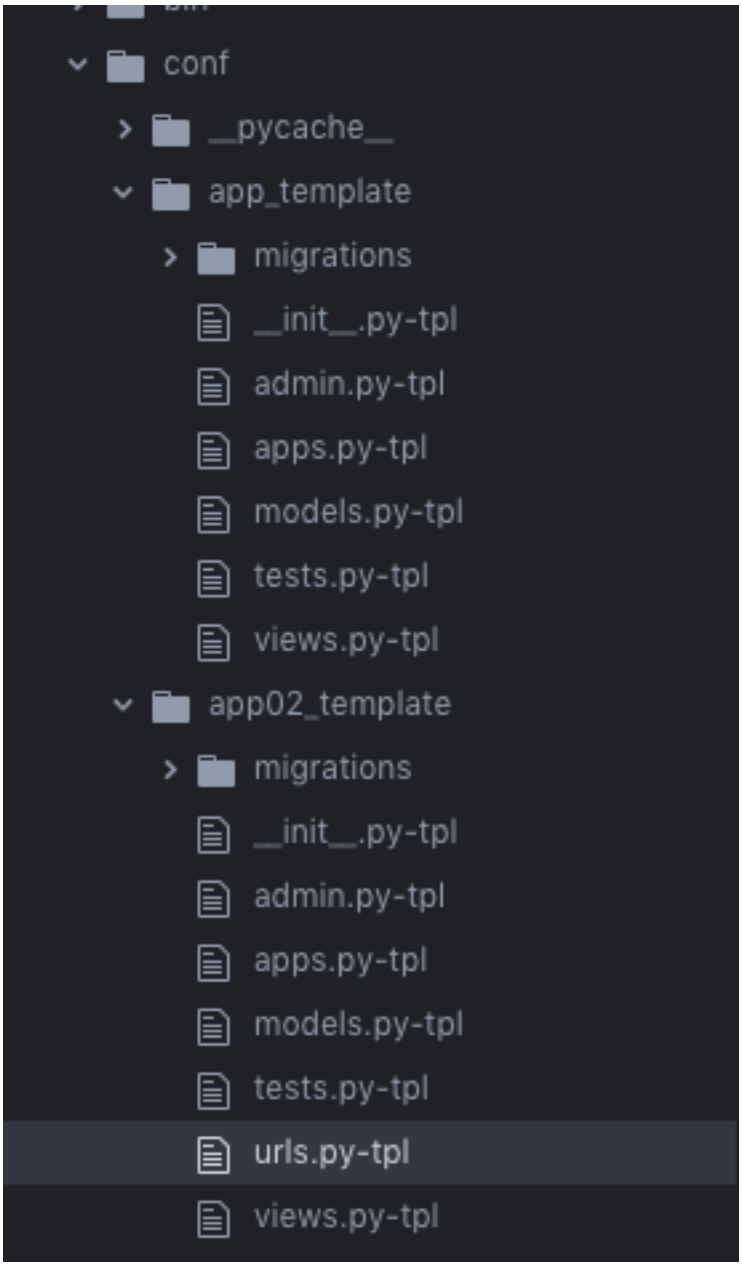
```
$ ls
django
$ cd django
$ git remote -v
origin  https://github.com/twoutlook/django.git (fetch)
origin  https://github.com/twoutlook/django.git (push)
$ cd ..
$ ls
django
$ python3.6 -m venv venv
$ . venv/bin/activate
(venv) $ pip freeze
(venv) $ pip install -e django
Obtaining file:///Users/pinglingchen/ksndjs/django001/django/django
Collecting pytz (from Django==2.2.dev20190102231945)
  Using cached https://files.pythonhosted.org/packages/f8/0e/2365ddc010afb3d79147f1d
d544e5ee24bf4ece58ab99b16fbb465ce6dc0/pytz-2018.7-py2.py3-none-any.whl
Collecting sqlparse (from Django==2.2.dev20190102231945)
  Using cached https://files.pythonhosted.org/packages/65/85/20bdd72f4537cf2c4d5d005
368d502b2f464ede22982e724a82c86268eda/sqlparse-0.2.4-py2.py3-none-any.whl
Installing collected packages: pytz, sqlparse, Django
  Running setup.py develop for Django
Successfully installed Django pytz-2018.7 sqlparse-0.2.4
You are using pip version 10.0.1, however version 18.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(venv) $ pip freeze
-e git+https://github.com/twoutlook/django.git@b5fe97a34ea527d4254b58c2e828450e7c321
57f#egg=Django
pytz==2018.7
sqlparse==0.2.4
You are using pip version 10.0.1, however version 18.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(venv) $
```

want/_static/want002-02.png

```
(venv) $ python manage.py startapp02 app005
(venv) $ tree app005
app005
├── __init__.py
├── admin.py
├── apps.py
├── migrations
│   └── __init__.py
├── models.py
├── tests.py
├── urls.py
└── views.py

1 directory, 8 files
(venv) $ cat app005/urls.py

# ... by Chenpingling, 2019-01-04
(venv) $
```

startapp02.py | urls.py-tpl | views.py-tpl

```python
from django.core.management.templates import TemplateCommand


class Command(TemplateCommand):
    help = (
        "Creates a Django app directory structure for the given app name in "
        "the current directory or optionally in the given directory."
    )
    missing_args_message = "You must provide an application name."

    def handle(self, **options):
        app_name = options.pop('name')
        target = options.pop('directory')
        super().handle('app02', app_name, target, **options)

```



want/_static/want003-04.png