# Django 2: View & Template

**IN608: Intermediate Application Development Concepts** 

**Kaiako: Tom Clark & Grayson Orr** 

### **Last Session's Content**

- Django
  - Overview
  - Pipenv
  - Installation
  - Creating a project
  - Development server
    - Creating an app
- MVC vs. MVT
- Route
- Model
- Admin Site

# Today's Content

- View
- Template

- Type of web page in your app which serves a function & has a template
- Each view is represented by a Python function or method (class-based views)
- A view is chosen by examining the requested URL
  - o Django uses a URLconf to get from a URL to a view
  - A URLconf maps URL patterns to views
- Resource: <a href="https://docs.djangoproject.com/en/3.0/topics/http/views/#writing-views">https://docs.djangoproject.com/en/3.0/topics/http/views/#writing-views</a>

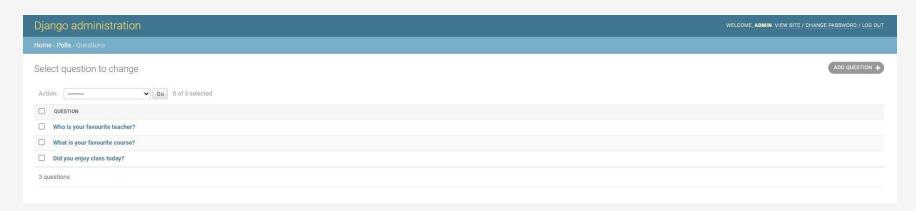
- polls/urls.py
- When someone request a page, for example, /polls/2/, Django will load the mvt.urls
   Python module pointed to by the ROOT\_URLCONF setting
- It finds the variable name urlpatterns & traverses the patterns in order
- After finding a match at polls/, it strips off the matching text polls/ & sends 2/ to the polls.url URLconf for further processing
- Results in a call to the detail() view
- Namespacing URL names
  - Larger Django projects may have more than one app
  - How does Django differentiate between them?
  - Add an app\_name to set the application namespace

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

app_name = 'polls'

urlpatterns = [
    path('', views.index, name='index'), # /polls/
    path('<int:question_id>/', views.detail, name='detail'), # /polls/2/
]
```

• Let's assume, we have created three Question objects in Django admin site



- polls/views.py
- Question.objects.order\_by('-pub\_date') get all Question objects & order by pub\_date (desc)
- render()
  - Combine a given template with a given context dictionary
  - Return an HttpResponse object with that rendered text
- get\_object\_or\_404()
  - Call get() on a given model manager
  - Raise Http404 instead of the model's DoesNotExist exception

#### Resources:

- https://docs.djangoproject.com/en/3.0/topics/db/gueries
- https://docs.djangoproject.com/en/3.0/topics/http/shortcuts/#render
- https://docs.djangoproject.com/en/3.0/topics/http/shortcuts/#get-object-or-404

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

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

- What is a template?
  - A text file, i.e., HTML, XML, CSV, etc.
  - Contains variables which get replaced with values when the template is evaluated
  - Contains tag which control the logic of the template
  - It should look very similar to EJS template language you saw in the Fundamentals of Web Development course
- In the polls/ directory, create a directory called templates/
- In the templates/ directory, create a directory called polls/
- In the polls/ directory, create two .html files index & detail
- Resource: <a href="https://docs.djangoproject.com/en/3.0/ref/templates/language">https://docs.djangoproject.com/en/3.0/ref/templates/language</a>
- App structure:

```
polls/
    __init__.py
    admin.py
    apps.py
    migrations/
    __init__.py
    models.py
    templates/
        polls/
              index.html
        detail.html
    tests.py
    views.py
```

- templates/polls/index.html
- If latest\_question\_list is not empty, loop through latest\_question\_list & display each Question object's question\_text as a link
- Run the development server & navigate to <a href="http://127.0.0.1:8000/polls">http://127.0.0.1:8000/polls</a>

- templates/polls/detail.html
- Click the **What is your favourite course?** link
- Question object's id is 2

