Django Tutorial Gists

Its a web framework

What is a web framework?

- collection of tools in one package that we can use to build web applications

Features of Django

- ORM
  - Object relational mapper
    - Helps us make DB queries
- URL routing
  - Helps us determine what logic to follow depending on URL of web request
- HTML templating
  - Allows us to format and add dynamic data into HTML

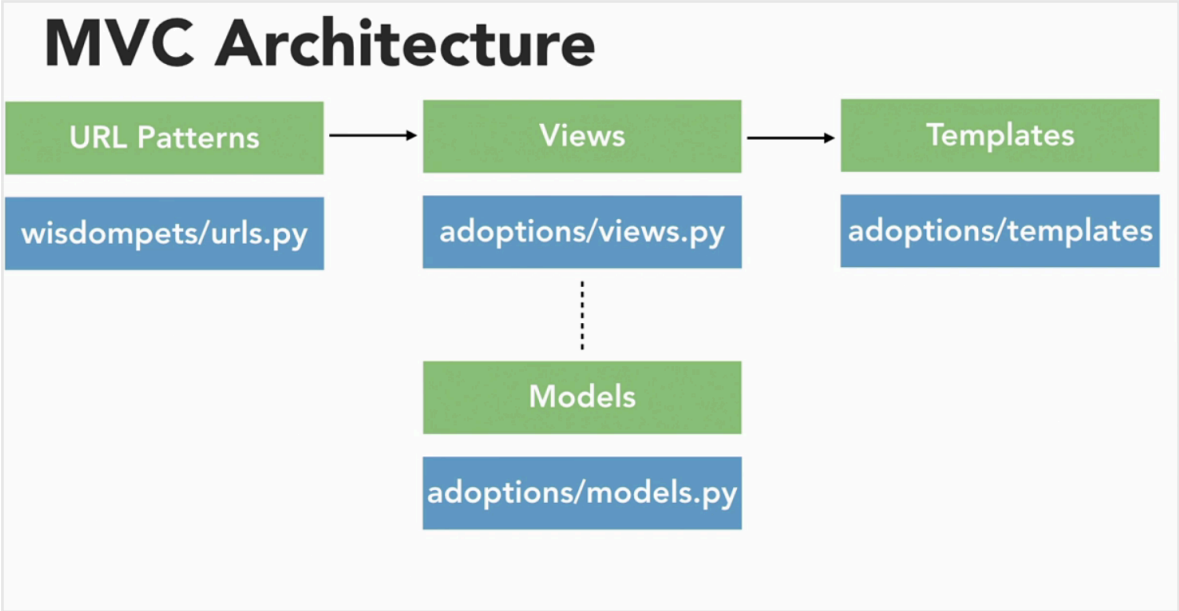
Django is not a web server

What is a Django app?

- Its a component in Django project
- Its basically just a folder with a set of file s
- Each app serves a particular purpose
  - example - a blog , a forum or a wiki

Django architecture

URL patterns
Views
Models
Templates



- URL Patterns
  - On receiving a web request, it uses the URL patterns to decide which view to pass on the request to!
    - wisdompets/urls.py
    - /adoptions/1/
      - Here 1 will be treated as a value of a variable and will be passed to a function
- Views
  - Control flow
  - Takes a HTTP req and returns a HTTP response
  - adoptions/views.py
- Models
  - To perform queries to the db, each view can use Django Models as needed
  - adoptions/models.py
- Template
  - Views can use templates
  - Presentation layer which deals with how the HTML response is displayed

Django Model

A model is a class inheriting from django.db.models.Model and defines fields as class attr
Each model is like a table in the spreadsheet
Each field in the model is like a column in the spreadsheet table

Each db record is like a row

Django fields

Textual data

CharField
TextField
EmailField
URLField

Integer data

IntegerField

```
    age = models.IntegerField(null=True)
```

    Here blank=True would have implied that age=0 while null=True implies age unknown

DecimalField

Other Fields

BooleanField
DateTimeField

Relational Fields

Foreign Key
ManyToManyField

```
class Vaccine(models.Model):
    name = models.CharField(max_length=50)

vaccinations = models.ManyToManyField('Vaccine', blank=True)
#This needs the name of the related model in quotes
```

Fields can take many attributes

models.CharField(max_length=10, blank=True)
null - No data for that field in that record
blank=True - indicated empty string

```
    sex = models.CharField(max_length=1, choices=SEX_CHOICES, blank=True)
```

    Here, blank = True implies sex unknown

choices - like sex - m/f

```
    SEX_CHOICES = [('M', 'Male'),('F','Female')]
```

https://docs.djangoproject.com/en/3.1/ref/models/fields/

Django Migrations

Model - defines the structure for the database tables
Migrations - Generates scripts to change the database structures

Every change in the models file needs a corresponding change in the migration directory

Initial migration - The first migration created for a Django app will create tables for the models that are defined
Unapplied migrations - Migrations created but not yet applied/run

##

Django Management

A Django management command is a script that is run using manage.py
In this case, we have written a script to load data from csv

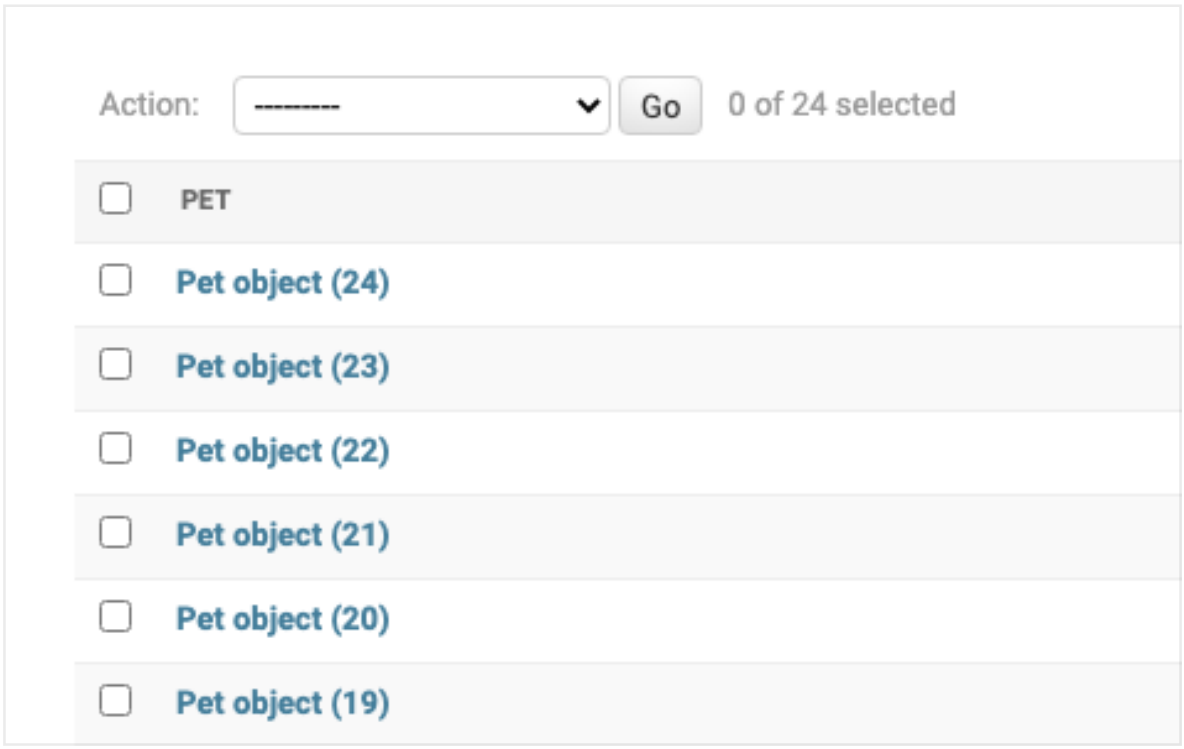
class Command(**BaseCommand**):

Django Admin

Creates an administrative interface for the project
Helps admin users to see and edit data

```
@admin.register(Pet)  #Registers the class with admin to associate it to a model
class PetAdmin(admin.ModelAdmin):
```

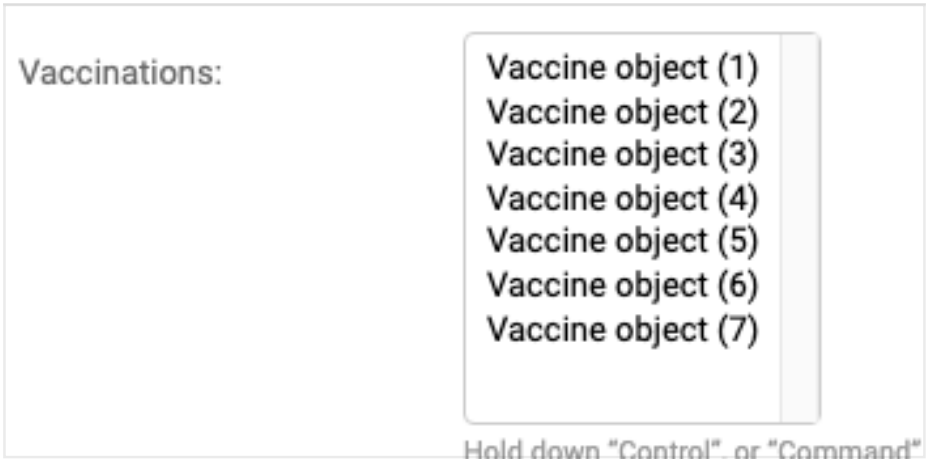
To view the changes we need to create a superuser
Once it is created then we run the server to see the changes
The display on Django admin page isnt very intuitive.



- We can handle this in admin.py using list_display

```
class PetAdmin(admin.ModelAdmin):
    list_display = ['name', 'species', 'breed', 'age', 'sex']
```

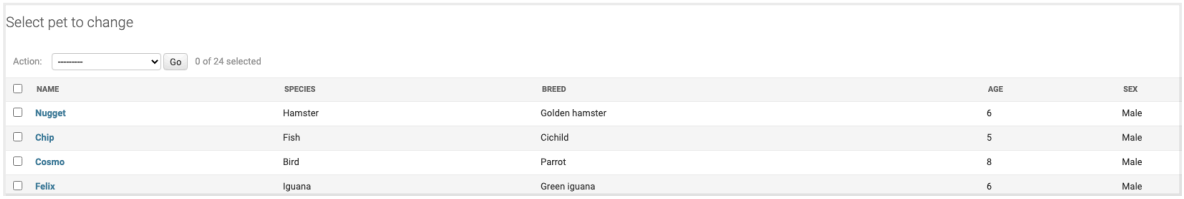
Also, Vaccinations in individual Pet object isnt displayed in a readable manner




```
class Vaccine(models.Model):
    name = models.CharField(max_length=50)

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

This tells Django the way to display Vaccines!!
After these changes - - - - >>



Django ORM thru shell

```
python manage.py shell
>>> from adoptions.models import Pet
>>> pets = Pet.objects.all() # returns a query set of
all pet objects
>>> pets
<QuerySet [<Pet: Pet object (1)>, <Pet: Pet object (2)>,
<Pet: Pet object (3)>, <Pet: Pet object (4)>, <Pet: Pet
object (5)>, <Pet: Pet object (6)>, <Pet: Pet object
(7)>, <Pet: Pet object (8)>, <Pet: Pet object (9)>,
<Pet: Pet object (10)>, <Pet: Pet object (11)>, <Pet:
Pet object (12)>, <Pet: Pet object (13)>, <Pet: Pet
object (14)>, <Pet: Pet object (15)>, <Pet: Pet object
(16)>, <Pet: Pet object (17)>, <Pet: Pet object (18)>,
<Pet: Pet object (19)>, <Pet: Pet object (20)>, '...
(remaining elements truncated)...']>
>>> pet = pets[0]
>>> pet
<Pet: Pet object (1)>
>>> pet.name
'Pepe'
>>> pet.age
0
>>> pet = Pet.objects.get(id=1)
>>> pet.name
'Pepe'

>>> Pet.objects.get(id=9999)
```

```
Traceback (most recent call last):
  File "<console>", line 1, in <module>
  File "/Users/vaibhav.singh/Desktop/knowledgeBase/
linkedinLearning/django/venv/lib/python3.7/site-
packages/django/db/models/manager.py", line 82, in
manager_method
    return getattr(self.get_queryset(), name)(*args,
**kwargs)
  File "/Users/vaibhav.singh/Desktop/knowledgeBase/
linkedinLearning/django/venv/lib/python3.7/site-
packages/django/db/models/query.py", line 417, in get
    self.model._meta.object_name
adoptions.models.Pet.DoesNotExist: Pet matching query
does not exist.
>>>
>>> Pet.objects.get(age=1)
Traceback (most recent call last):
  File "<console>", line 1, in <module>
  File "/Users/vaibhav.singh/Desktop/knowledgeBase/
linkedinLearning/django/venv/lib/python3.7/site-
packages/django/db/models/manager.py", line 82, in
manager_method
    return getattr(self.get_queryset(), name)(*args,
**kwargs)
  File "/Users/vaibhav.singh/Desktop/knowledgeBase/
linkedinLearning/django/venv/lib/python3.7/site-
packages/django/db/models/query.py", line 422, in get
    num if not limit or num < limit else 'more than %s'
% (limit - 1),
adoptions.models.Pet.MultipleObjectsReturned: get()
returned more than one Pet -- it returned 3!
>>>
```

The above two exceptions are pretty common and need to be handled in the views

URL Patterns / URL confs

wisdompets/urls.py

- First part of our application code

http://wisdompets.com
If url navigates to an empty path then we should load the homepage

http://wisdompets.com/adoptions/1
On clicking on a pet objects we should get something like the above URL

URL Pattern	View	Template
adoptions/1/	pet_detail()	pet_detail.html

wisdompets/urls.py

```
urlpatterns  = [
    path('', views.home, name='home'),
    path('adoptions/<int:pet_id>', views.pet_detail,
name='pet_detail'),
]
```

Checks each path in order
Django returns 404 in case of no match

p**ath('adoptions/<int:pet_id>/', views.pet_detail, name='pet_detail'),**

- **adoptions/<int:pet_id>'**
  - Path converter
  - **<int:pet_id>'**
    - Capture group
  - pet_id - keyword argument
  - int - path converter type
    - https://docs.djangoproject.com/en/3.1/topics/http/urls/
  -

- **name='pet_detail'**
  - will be used for links in template

Django Views

```
def home(request):
    #return HttpResponse('<p>Home view!!</p>')
    pets = Pet.objects.all()
    return render(request, 'home.html', {
        'pets':pets,
    })
```

Note -
render(request, view, dictionary of values to be passed to template)

Exception Handling -

```
def pet_detail(request, pet_id):
```

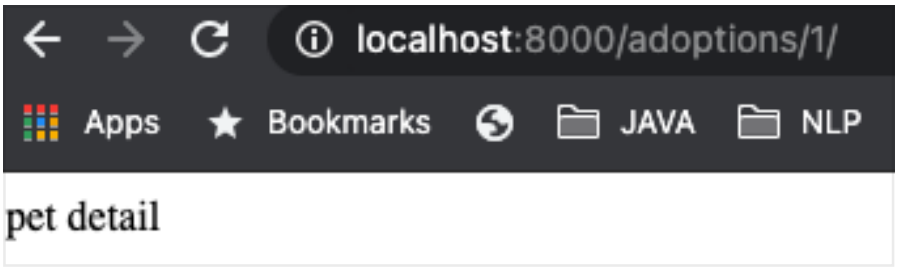
```
    #return HttpResponse(f'<p>pet detail view with id
{pet_id}</p>')
    try:
        pet = Pet.objects.get(id=pet_id)

    except Pet.DoesNotExist:
        raise Http404("Pet not found")

    return render(request, 'pet_detail.html', {
        'pet': pet,
    })
```

Django Templates

- In app/templates/
- We will keep our html files here
- add a pet_detail.html and home.html accordingly.



Template syntax

- Variable values - {{ var }}
    - <h3> The pet name is {{ pet.name }} </h3>
        - <h3>scooter</h3>

- Template tags - {% tag %} - - - — - > {% for %}

```
{% for pet in pets %}
    <li>{{ pet.name }}</li>
{% endfor %}
```

```
{% url 'pet_detail' pet_id %}
```

- /adoptions/1/
- Template filters - {{ variable | filter }}

```
<h3> The pet name is {{ pet.name| capfirst }} </h3>
```

    - <h3>Scooter</h3>
        - output has first letter capitalized

```
<p>Submitted on: {{ pet.submission_date|date:"M d Y" }}
</p>
```

- **Rendered html - Submitted on: Nov 28 2016**

Template inheritance

In base.html ->

```
{% block content %}
 _ _ _ _ _ _ _
{% endblock %}
```

- **block** is used for overriding specific parts of a template.
- In our case, we have a **block** named **content** and this is supposed to be overridden by children that inherit from this template.

In home.html, add these lines in the beginning !

```
{% extends "base.html" %}
{% block content %}
_ _
home body
{% endblock %}
```

Django static files

- Javascript
- CSS and images

in settings.py add this -

```
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, 'static')
]
```

This tells Django about the path to static files directory

- Note that it should be in the same directory as manage.py

##

Useful commands

Creates a project named wisdompets
django-admin startproject wisdompets

(venv) ~/D/k/l/d/p/wisdompets ❯❯❯ ls
adoptions manage.py wisdompets

- manage.py
  - Runs commands for the project

~/D/k/l/d/p/w/wisdompets ❯❯❯ ls
__init__.py asgi.py settings.py urls.py wsgi.py

- __init__
  - Tells python that the folder contains python code
- asgi and wsgi
  - provides hooks for web servers such as apache or nginx when django is running on a live website
- settings
  - configures the django project
- urls
  - routes web requests based on urls

==========

Running a django server
python3 manage.py runserver

==========

Starting an app
python3 manage.py startapp adoptions
(venv) ~/D/k/l/d/p/wisdompets ❯❯❯ ls
adoptions **db.sqlite3** manage.py wisdompets

- Django creates a db.sqlite3 file to have a database to work with

(venv) ~/D/k/l/d/p/w/adoptions ❯❯❯ ls
__init__.py admin.py apps.py migrations models.py tests.py
views.py

In settings.py (wisdompets i.e. projects directory)

- defines the set of apps that our django project would use

```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

Need to add adoptions app here

Lets explore this directory now -

```
(venv) ~/D/k/l/d/p/w/adoptions ❯❯❯ ls
__init__.py admin.py    apps.py     migrations  models.py   tests.py
views.py
```

- apps.py
  - Controls settings specific to this app
- models.py
  - Provides a data layer that django uses to create a database schema and queries
- admins.py
  - Provides an administrative interface for the app
  - allows us to see and edit this app
- urls.py
  - Used for url routing specific to the app
- views.py
  - Defines the logic and control flow for handling requests
  - Defines the HTTP responses to return
- tests.py
  - used to define unit tests to test the functionality of the app
- migrations folder
  - holds files that django uses to migrate the database

Migration commands

python3 manage.py makemigrations

- Generates migration files
- Reads the models.py file and checks the database to make sure that the database structure matches the current models file

- Creates numbered files and saves it in app/migrations/ directory

python3 manage.py showmigrations
python3 manage.py migrate

- Runs all the generated migrations that have not yet run
- Can run specific migrations for a specific app
    - python3 manage.py migrate adoptions 1

(venv) ~/D/k/l/d/p/wisdompets ❯❯❯ python3 manage.py makemigrations
Migrations for 'adoptions':
  adoptions/migrations/0001_initial.py
    - Create model Vaccine
    - Create model Pet
(venv) ~/D/k/l/d/p/wisdompets ❯❯❯ ls adoptions/migrations
0001_initial.py __init__.py   __pycache__

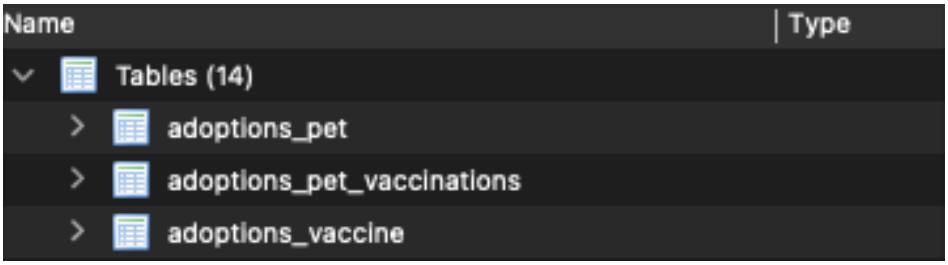
(venv) ~/D/k/l/d/p/wisdompets ❯❯❯ **python3 manage.py showmigrations**
admin
 [ ] 0001_initial **# here [ ] implies that these migrations have not yet been applied**
 [ ] 0002_logentry_remove_auto_add
 [ ] 0003_logentry_add_action_flag_choices
adoptions
 [ ] 0001_initial
auth
...
...

(venv) ~/D/k/l/d/p/wisdompets ❯❯❯ python3 manage.py migrate
Operations to perform:
  Apply all migrations: admin, adoptions, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying adoptions.0001_initial... OK
..
..

(venv) ~/D/k/l/d/p/wisdompets ❯❯❯ python3 manage.py showmigrations
admin
 [X] 0001_initial

[X] 0002_logentry_remove_auto_add
[X] 0003_logentry_add_action_flag_choices
adoptions
[X] 0001_initial
auth
[X] 0001_initial

Visualizing on sqlite db browser



adoptions_pet_vaccinations - Stores the many to many relationships

Management

Copied a management directory to the app directory - This has a script to load data from csv using manage.py
`python3 manage.py load_pet_data`

This successfully loads data to sqlite3 db

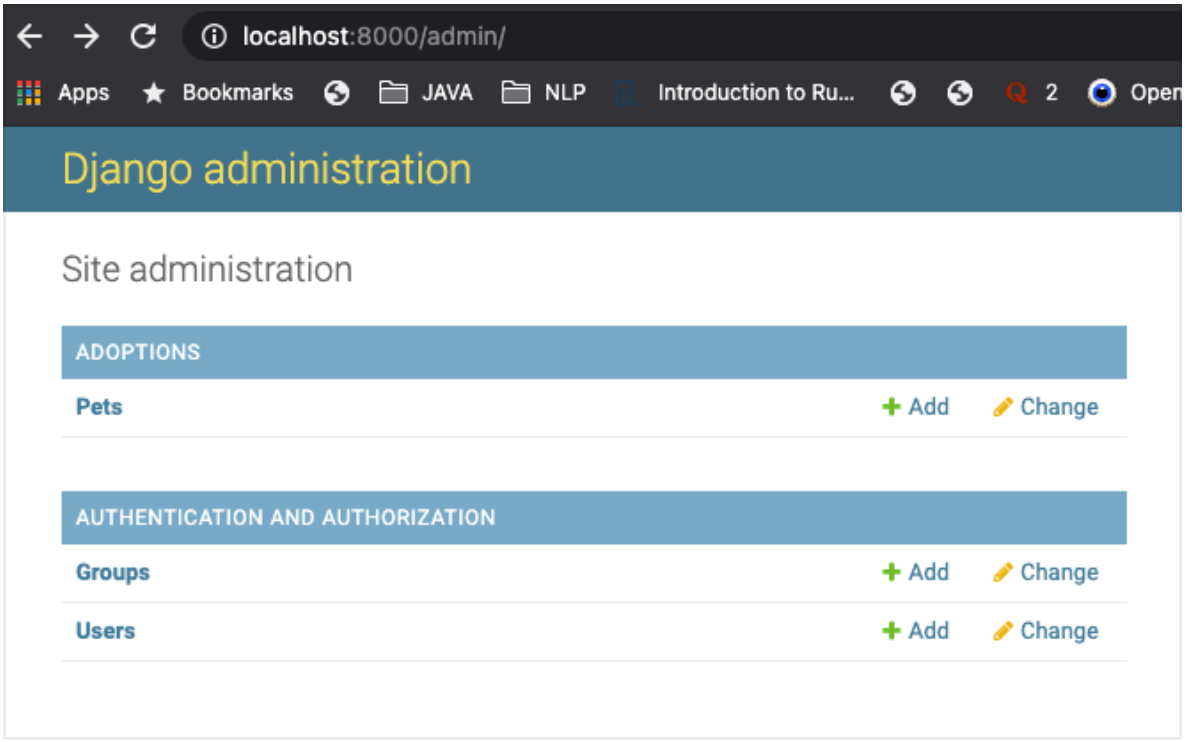
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Pepe	Reggie Tupp	Rabbit	Cinnamon rabbit	Six-month-old Pepe is ...	M	2016-11-28 13:30:00	0
2	2	Scooter	Zachary Heilyn	Hedg...	White-bellied	You have to keep an ey...	M	2016-11-28 14:45:00	2
3	3	Zera	Austin Finnagan	Iguana	Cayman brac iguana	This iguana is on the ...	F	2016-11-29 13:15:00	3
4	4	Oddball	Howie Cadell	Guine...	American guinea pig	Oddball was the runt of...	M	2016-11-29 10:00:00	1
5	5	Chyna	Sandie Gobnet	Turtle	Terrapin	Chyna got her name ...	F	2016-11-29 14:30:00	13
6	6	Rio	Philip Ransu	Dog	French bulldog	Rio, the 5-year-old ...	M	2016-11-28 10:15:00	5
7	7	Nadalee	Krystle Valerija	Dog	Chihuahua	Nadalee is a 7-year-ol...	F	2016-11-28 16:00:00	7
8	8	Scout	Nicolette Bardeau	Dog	Jack Russell terrier	Scout suffers from ...	M	2016-11-28 09:00:00	5

admin

Creating a superuser
python3 manage.py createsuperuser

```
(venv) ~/D/k/l/d/p/wisdompets ❯❯❯ python3 manage.py
createsuperuser
Username (leave blank to use 'vaibhav.singh'): vaibhavs
Email address:
Password:
Password (again):
Superuser created successfully.
```

Password used by me – learndjango



Interactive shell

python3 manage.py shell