

# Django



A demo on how to get a Django web server running

## Getting Started

### Start Project

```
django-admin startproject django_project
```

### Run Server

```
python manage.py runserver
```

Visit <http://localhost:8000/>

## Starting the 'groceries' app

Ctrl+c to stop the previous **runserver** command

Start app

```
python manage.py startapp groceries
```

## Project Setup

```
vim django_projects/urls.py
```

```
from django.contrib import admin

from django.urls import path, include

urlpatterns = [

    path('admin/', admin.site.urls),

    path('groceries/', include("groceries.urls")),

]
```

## Project Setup (cont.)

```
vim django_projects/settings.py
```

```
INSTALLED_APPS = [  
    ...  
    'groceries.apps.GroceriesConfig',  
]
```

## Configure URLs

```
cd groceries  
vim urls.py
```

```
from django.urls import path  
  
from . import views  
  
app_name = "groceries"  
  
urlpatterns = [  
    path("", views.index, name="index"),  
    path("list/", views.GroceriesList.as_view(), name="grocery_list"),  
    path("item/<int:pk>", views.GroceriesDetail.as_view(), name="grocery_detail"),  
    path("item/create", views.GroceriesCreate.as_view(), name="grocery_create"),  
]
```

## Configure Views

```
vim views.py
```

```
from django.shortcuts import render

from django.views.generic import ListView, DetailView, CreateView

from django.urls import reverse_lazy

from .models import Item
```

```
def index(request):

    return render(request, "groceries/index.html")
```

## Configure Views (cont.)

```
class GroceriesList(ListView):  
    model = Item  
  
    context_object_name = "item_list"  
  
    template_name = "groceries/item_list.html"
```

```
class GroceriesDetail(DetailView):  
    model = Item  
  
    template_name = "groceries/item_detail.html"
```

```
class GroceriesCreate(CreateView):  
    model = Item  
  
    success_url = reverse_lazy('groceries:grocery_list')  
  
    template_name = "groceries/item_create.html"
```



## Configure Models

```
vim models.py
```

```
from django.db import models

class Item(models.Model):

    name = models.CharField(max_length=100)

    quantity = models.IntegerField()
```

## Configure Templates

```
mkdir templates  
  
mkdir templates/groceries  
  
cd templates/groceries  
  
cp /srv/django-demo/index.html .  
  
cp /srv/django-demo/item_list.html .  
  
cp /srv/django-demo/item_detail.html .  
  
cp /srv/django-demo/item_create.html .
```

## Configure Templates (cont.)

```
<doctype! html>

<html>

  <head>

    <title>Grocery List</title>

  </head>

  <body>

    <h1>Grocery List</h1>

    <p>This is a list containing all of your grocery items</p>

    {% if item_list %}

      {% for item in item_list %}

        <li>

          <a href="{% url 'groceries:grocery_detail' pk=item.pk %}">{{item.name}}</a>

        </li>

      {% endfor %}

    {% else %}

      <p>Oops! Looks like you don't have any items yet!</p>

    {% endif %}

    <a href="{% url 'groceries:grocery_create' %}">Add a grocery item</a>

  </body>

</html>
```

## Testing

```
cd ../../..  
  
python manage.py makemigrations groceries  
  
python manage.py migrate  
  
python manage.py runserver
```

Visit <http://localhost:8000/groceries/>

# Configure Forms

```
cd groceries  
vim forms.py
```

```
from django.forms import ModelForm  
  
from django.core.exceptions import ValidationError  
  
from .models import *  
  
class ItemForm(ModelForm):  
    class Meta:  
        model = Item  
        fields = ["name", "quantity"]  
  
    def clean(self):  
        cleaned_data = super().clean()  
        name = cleaned_data["name"]  
  
        if Item.duplicate_name(name):  
            raise ValidationError(  
                "An item with that name already exists",  
                code="duplicate name",  
                params={},  
            )
```

## Configure Models

```
vim models.py
```

```
class Item(models.Model):  
  
    name = models.CharField(max_length=100)  
    quantity = models.IntegerField()  
  
    def duplicate_name(name):  
  
        items = Item.objects.all()  
  
        for item in items:  
            if item.name == name:  
                return True  
  
        return False
```

## Configure Views

```
vim views.py
```

```
from django.shortcuts import render

from django.views.generic import ListView, DetailView, CreateView

from django.urls import reverse_lazy

from .models import Item

from .forms import ItemForm


class GroceriesCreate(CreateView):

    model = Item

    form_class = ItemForm

    success_url = reverse_lazy('groceries:grocery_list')

    template_name = "groceries/item_create.html"
```

## Testing

```
cd ..  
  
python manage.py makemigrations groceries  
  
python manage.py runserver
```

Visit <http://localhost:8000/groceries/>



## Admin Pages

```
vim groceries/admin.py
```

```
from django.contrib import admin
```

```
from .models import *
```

```
# Register your models here.
```

```
admin.site.register(Item)
```

## Admin Pages (cont.)

```
python manage.py createsuperuser
```

Choose a username/password you can remember

Example:

- user: admin
- email: admin@example.com
- pass: 12345

## Testing

Restart the development server

```
python manage.py runserver
```

Visit <http://localhost:8000/admin/>