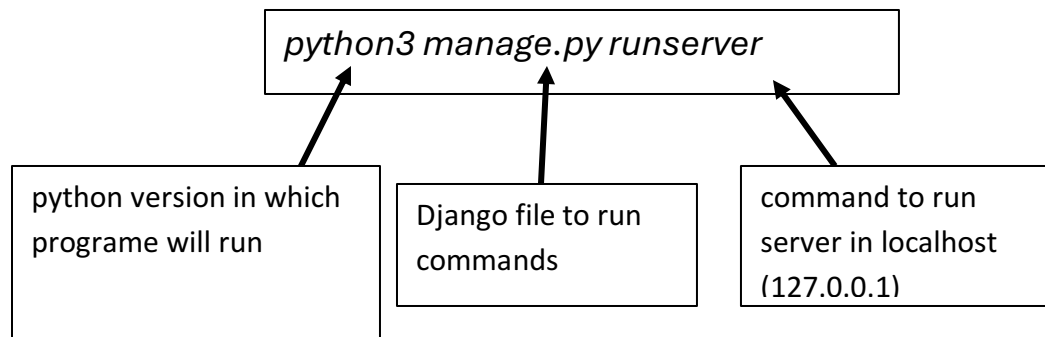


Create new project: `django-admin startproject <project name>`

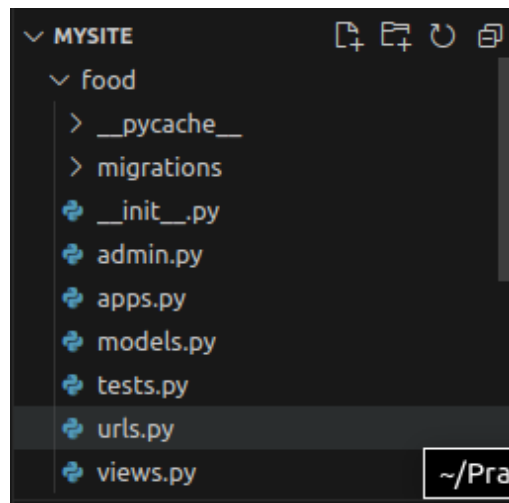
Start Django server: `python3 manage.py runserver`



App: In Django app is directory which created for special functionality of the project.

Create app: `django-admin startapp`

-> Here is an app 'food' with which will come with following files:

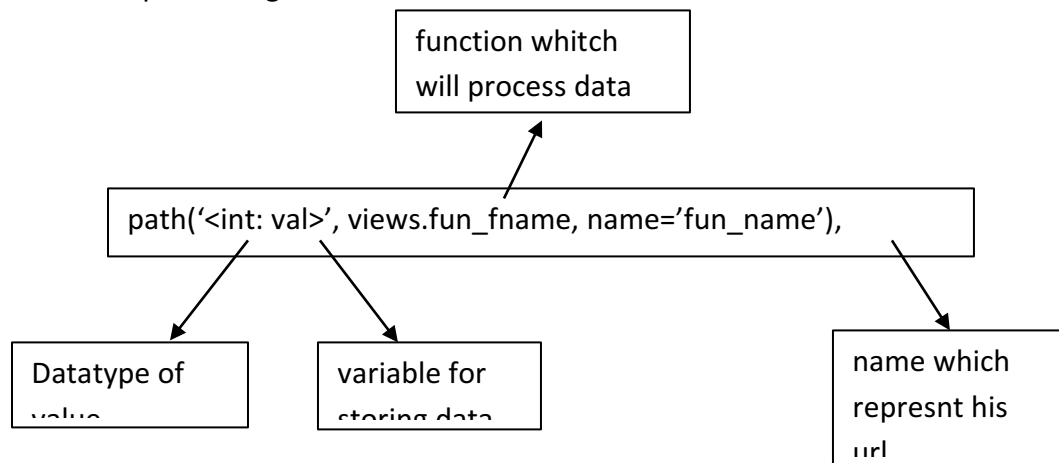


Note: it's good practice to have separate `urls.py` files for each app. When you add

Urls.py: Django first search for `ROOT_URLCONF` in `setting.py`. Which tell about where it must search for URLs

Note: use name space in each URL e.g `'name_space: url_name'`

Provide input through URLs.



In template file where you will add this URLs and another file where you will redirect to above path

Index.html

```
{% for i in item %}

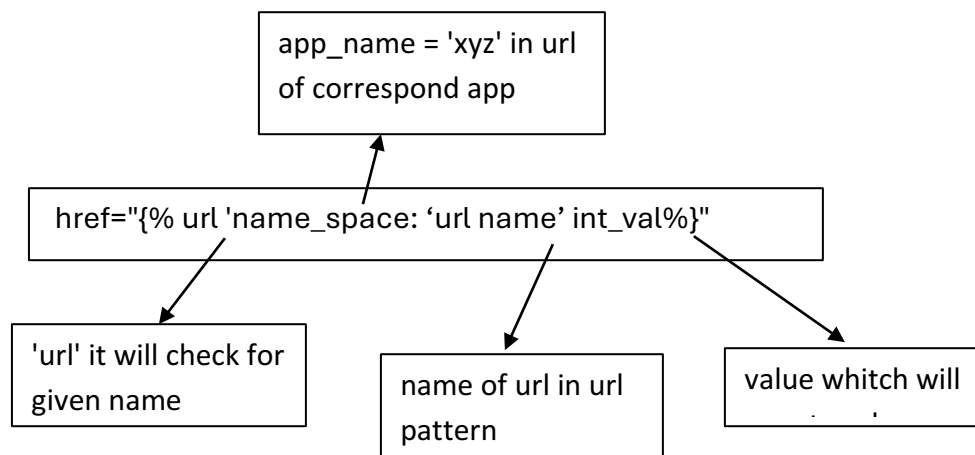
<ol>

  <a href="{% url 'name_space: 'url name' int_val%}">

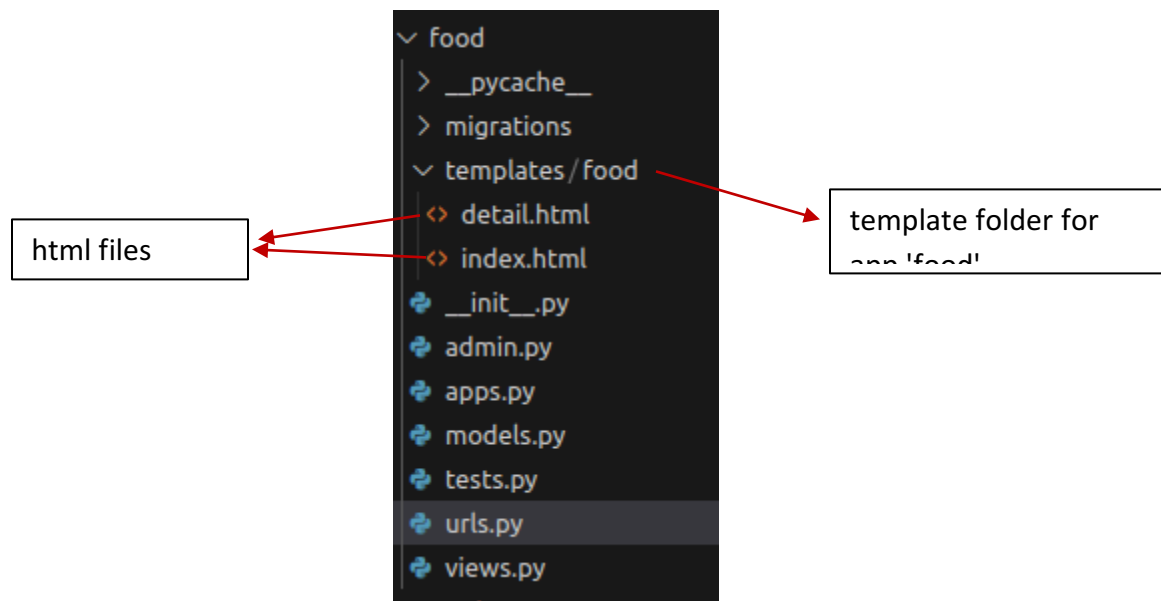
    <li>{{i.items_name}}</li></a>

</ol>

{% endfor %}
```



Template: if you add template in your app you don't have to specify path in setting.py



you need to add folder with name static in your app. If you have static assets that aren't tied to a particular app then you must add static change in setting.py in project.

Setting.py

```
STATIC_URL = "static/"
```

for defining static file other than app

```
STATICFILES_DIRS = [
    BASE_DIR / "static",
    "/var/www/static/",
]
```

Forms: It used to take Create Update operation in Django.

- 1) Create forms.py file in your app (form all ways related to some models for data)
- 2) Then with in forms.py file import `` from django import forms ``
- 3) Then create a form or meta class to inherited form models.

Authentication and Django login/logout forms:

- 1) Create a user app and CustomUser model if need
- 2) Then user `'''` from `django.contrib.auth import views '''`
 - a. Then use his views for creating url for login and logout
 - i. `''' views.LoginView.as_view(template_name='users/login.html'), '''`

above code is for login

- ii. `''' views.LogoutView.as_view(template_name='users/login.html'), '''`

above code is for logout

Setting.py:

1) add users model in setting.py

2) add `''' LOGIN_REDIRECT_URL = 'food:items' '''`

above path will redirect to specific path after login

Note: login and logout always will in 'POST'

Routing: you must use decorator from Django auth `@login_required`. This will prevent the user from accessing a particular URL.

Image:

Setting.py

Place where picture dir will create

```
MEDIA_ROOT = os.path.join(BASE_DIR, 'picture')
```

Place where Django search for images

```
MEDIA_URL = '/picture/'
```

Project urls.py

This will help to load media file to html

```
from django.conf import settings
```

```
from django.conf.urls.static import static
```

```
urlpatterns += [
```

```
# ... the rest of your URLconf goes here ...
```

```
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Signals in Django:

- 1) Create signals.py in your app.
- 2) Then import receiver, method_save, etc

```
''' from django.db.models.signals import post_save

        from django.contrib.auth.models import User

        from django.dispatch import receiver

        from .models import Profile '''
```

- 3) Then add decorator for your method e.g

```
''' @receiver(post_save, sender=User) '''
```

Above method will only call on new user will added

- 4) Then you have to add in your app_name-->app.py file

```
''' def ready(self):

        import users.signals '''
```

Django Class based views:

- 1) generic.list for Listviews, generic.detail for DetailViews, and generic.edit for Createviews
- 2) add folloing codes as-per requirements

```
model = model_name
```

```
fields = [.....]
```

```
template_name = "
```

- 3) for forms

```
''' def form_valid(self, form):

        form.instance.user = self.request.user

        return super().form_valid(form) '''
```

- 4) from Ddjango.urls import furls

```
''' def get_absolute_url(self):
```

```
return reverse('food:details', kwargs={"pk": self.pk}) ``
```