**Per View Caching in Django 5**

Videono:59

Codefile:ch50

In per-view caching, we don’t need to add anything in the settings middleware.

Database caching:

# Database Caching

CACHES = {

    'default': {

        'BACKEND': 'django.core.cache.backends.db.DatabaseCache',

        'LOCATION': 'student\_cache',

    }

}

And run command

Python manage.py makemigrations

Python manage.py migrate

Python manage.py createcachetable

This is per-view caching, so we can implement it directly in the view. We also do this work in urls here I shows both

Student/views.py:

from django.shortcuts import render

from django.views.decorators.cache import cache\_page

def home(request):

  return render(request, 'student/home.html')

@cache\_page(30)  # here we can cache page for 30 seconds

def course(request):

  return render(request, 'student/course.html')

def result(request):

  return render(request, 'student/result.html')

Through URL SECTION

student/urls.py:

check home and result section

from django.urls import path

from student.views import home, course, result

from django.views.decorators.cache import cache\_page

urlpatterns = [

    path('', home, name="home"),

    path('home/', cache\_page(60)(home), name="home1"),

    path('index/', home, name="index"),

    path('course/', course, name="course"),

    path('result/', cache\_page(30)(result), name="result"),

]

File Based Caching:

Settings.py:

# File Based Caching

CACHES = {

    'default': {

        'BACKEND': 'django.core.cache.backends.filebased.FileBasedCache',

        'LOCATION': 'student\_cache',

        # 'LOCATION': 'C:\django5\ch49\student\_cache',

        # 'LOCATION': '/var/tmp/django5/ch49/student\_cache',

    }

}

Local Memory Cache:

# Local Memory Caching

CACHES = {

    'default': {

        'BACKEND': 'django.core.cache.backends.locmem.LocMemCache',

        'LOCATION': 'student\_cache',

    }

}

How we clear the cache :

### Clearing the Cache in Django

Django provides a cache.clear() method that removes **all stored cache**.

Step 1: Import cache

from django.core.cache import cache

from django.http import HttpResponse

Step 2: Create a view to clear cache

def clear\_cache(request):

    cache.clear()  # This will delete all cache data

    return HttpResponse("All cache cleared successfully!")

Step 3: Add URL for clearing cache

from django.urls import path

from . import views

urlpatterns = [

    # your other views

    path('clear-cache/', views.clear\_cache, name='clear\_cache'),

]

### 🔹 Example with Per-View Caching

from django.views.decorators.cache import cache\_page

from django.http import HttpResponse

from django.core.cache import cache

# Cached view for 60 seconds

@cache\_page(60)

def my\_view(request):

    return HttpResponse("This is a cached page.")

# Clear cache manually

def clear\_cache(request):

    cache.clear()

    return HttpResponse("All cache cleared successfully!")

👉 Now, when you visit /clear-cache/, Django will delete all the stored cache.