

How do you create a class-based view in Django?

Steps:


1. Import View from django.views.
2. Create a class inheriting from View.
3. Define methods like get(), post(), etc., based on HTTP requests.

Example:



```
1 from django.http import HttpResponse
2 from django.views import View
3
4 class HomeView(View):
5     def get(self, request):
6         return HttpResponse("Welcome to Zamazor!")
7
```

Connecting to URL:



```
1 from django.urls import path
2 from .views import HomeView
3
4 urlpatterns = [
5     path('', HomeView.as_view(), name='home'),
6 ]
7
```

What is the role of urls.py in connecting a view to a URL?

Purpose:

urls.py maps URL patterns to specific views, acting as a **router** for incoming HTTP requests.

How It Works:

- When a request is made, Django checks urls.py for a matching pattern.
- If found, it calls the corresponding view.

Example:



```
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('about/', views.about_view, name='about'),
6 ]
7
```

What is the purpose of the context dictionary in Django views?

Definition:

The **context dictionary** holds data passed from the view to the template, allowing dynamic content rendering.

Usage:

- Keys in the dictionary become variables in the template.
- Helps in displaying data like products, user info, etc.

Example:

```
1 from django.shortcuts import render
2
3 def product_view(request):
4     context = {'product_name': 'Zamazor T-Shirt', 'price': 499}
5     return render(request, 'product.html', context)
6
```

In product.html:

```
1 <h1>{{ product_name }}</h1>
2 <p>Price: ₹{{ price }}</p>
3
```

Output:

Displays:

Zamazor T-Shirt

Price: ₹499

