Let's explore few

Shortcut

Functions In Django

The package django.shortcuts collects helper functions and classes that simplify common tasks when working with web applications.

These shortcut functions are designed to streamline the development process and increase code readability. Here are some commonly used Django shortcut functions:

1. render()

It renders a template with a given context and returns an HttpResponse.

views.py

```python
from django.shortcuts import render

def my_view(request):
    context = {'key': 'value'}
    return render(request,
                  template_name = 'my_template.html',
                  context = context,
                  content_type=None,
                  status=None,
                  using=None)
```

pass the context data, you want to use in templates

pass the status default is 200

pass the engine name to load the template

pass the content type default is text/html

pass the templates name that you want to render.

2. redirect()

It redirects the user to a different URL, base on the arguments passed.

The arguments could be:

1. A model: the model's get_absolute_url() function will be called.

```
from django.shortcuts import redirect
from .models import Student

def my_view(request):
    ...
    obj = Student.objects.get(id=1)
    return redirect(obj)
```

By passing some object; that object's get_absolute_url() method will be called to figure out the redirect URL:

2. A view name, possibly with arguments: reverse() will be used to reverse-resolve the name.

```
def my_view(request):
    ...
    return redirect("some-view-name", foo="bar")
```

pass the view name

if your view have any arguments you have to pass the arguments as well.

3. By passing a hardcoded URL to redirect to:

```
def my_view(request):
    ...
    return redirect("/some/url/")
```

you can also pass the hardcoded url and the full url as well

```
return redirect(""https://example.com/"")
```

3. get_object_or_404()

It call the get method on the model, if an object does not found, it simply return the 404 http response, otherwise return an object.

```python
from django.shortcuts import get_object_or_404

def my_view(request):
    obj = get_object_or_404(MyModel, pk=1)
```

**see the difference

pass the model name

pass the primary key

This example is equivalent to this:

```python
from django.http import Http404

def my_view(request):
    try:
        obj = MyModel.objects.get(pk=1)
    except MyModel.DoesNotExist:
        raise Http404("No MyModel matches the given query.")
```

4. get_list_or_404()

It call the filter method on the model, if no object found, it simply return the 404 http response, otherwise return an objects.

```python
from django.shortcuts import get_list_or_404

def my_view(request):
    my_objects = get_list_or_404(MyModel, published=True)
```

pass the model name

pass the field name that you want to apply filter

**see the difference

This example is equivalent to this:

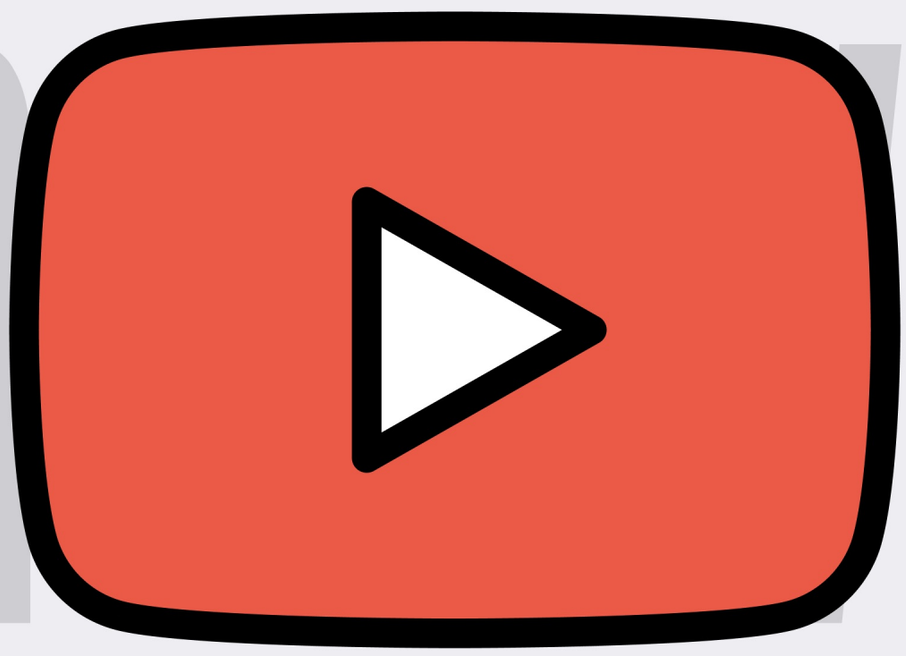
```python
from django.http import Http404

def my_view(request):
    my_objects = list(MyModel.objects.filter(published=True))
    if not my_objects:
        raise Http404("No MyModel matches the given query.")
```

These are just a few examples of Django shortcut functions. The Django documentation provides a comprehensive list and detailed explanations for each function: Django Shortcuts.

https://docs.djangoproject.com/en/5.0/topics/http/shortcuts/

For more django related content

Subscribe To My Youtube Channel



Link in bio

DID YOU FIND THIS HELPFUL

Follow for more