What is context_object_name in django views?

Asked 11 years, 6 months ago Modified 2 years, 1 month ago Viewed 30k times

34

I'm new to django. And now I'm studying using the class-based generic views. Could someone please explain the aim and use of **context_object_name** attribute?

django-class-based-views

Share Follow

edited Oct 5, 2016 at 20:05

asked May 11, 2011 at 5:43

megido
3,885 ● 6 ● 28 ● 33

4 Answers

Sorted by: Highest score (default)

65

If you do not provide "context_object_name", your view may look like this:

```
<ul>
    {% for publisher in object_list %}
        <li>{{ publisher.name }}</li>
    {% endfor %}
</ul>
```

But if you provide like {"context_object_name": "publisher_list"}, then you can write view like:

```
<ul>
    {% for publisher in publisher_list %}
        <li>{{ publisher.name }}</li>
    {% endfor %}
</ul>
```

That means you can change the original parameter name(object_list) into any name through "context_object_name" for your view. Hope that help:)

Share Follow

answered Nov 20, 2013 at 16:11
user2395922
930 ● 1 ● 9 ● 9

24

Ok, I've got it by myself! :)

It's just a human-understandable name of variable to access from templates

https://docs.djangoproject.com/en/1.10/topics/class-based-views/generic-display/#making-friendly-template-contexts

Share Follow

edited Oct 5, 2016 at 20:05

answered May 12, 2011 at 8:05
megido
3,885 ● 6 ● 28 ● 33

2 Here is the link for the latest version (ver 1.8): docs.djangoproject.com/en/1.8/topics/class-based-views/... – Cheng Apr 27, 2015 at 13:24

10

Lets assume the following posts/views.py:

```
# posts/views.py
from django.views.generic import ListView from .models import Post

class HomePageView(ListView):
    model = Post
    template_name = 'home.html'
```

On the first line we're importing ListView and in the second line we need to explicitly define which model we're using. In the view, we subclass ListView, specify our model name and specify our template reference. Internally ListView returns an object called ***object_list*** that we want to display in our template.

In our templates file home.html we can use the Django Templating Language's for loop to list all the objects in **object_list**

Why object_list? This is the name of the variable that ListView returns to us.

Lets look at our templates/home.html

```
<!-- templates/home.html -->
<h1>Message board homepage</h1>
<ul>

     {% for post in object_list %}

          <li>{{ post }}</li>

     {% endfor %}
 </ul>
```

You see the object_list above? It is not a very friendly name? To make it more user-friendly, we can provide instead an explicit name using **context_object_name**.

This helps anyone else reading the code to understand what is variable in the template context, plus it is much easier to read and understand.

So let's go back to our posts/views.py and change it by adding the one line below:

```
context_object_name = 'all_posts_list' # <----- new
```

So our new views.py looks like this now:

```
# posts/views.py
from django.views.generic import ListView from .models import Post

class HomePageView(ListView): model = Post

    template_name = 'home.html'

    context_object_name = 'all_posts_list' # <----- new
```

And let's not forget to update our template now:

```
<!-- templates/home.html -->
<h1>Message board homepage</h1>
<ul>

   {% for post in all_posts_list %}
```

```
        <li>{{ post }}</li>

    {% endfor %}

</ul>
```

You could have left as object_list and it would still work, but you get the idea.

Share Follow

edited Oct 26, 2018 at 7:22

answered Oct 6, 2018 at 15:14

Stryker
5,422 ● 54 ● 66

2

Consider these 2 code snippet

A. Using function based view:

```
def index(request):
    product_list = Product.objects.all()
    return render(request, 'product/index.html', {'product_list': **product_list**})
```

B. Using class based view

```
class ProductListView(ListView):
    model = Product
    template_name = 'product/index.html'
    context_object_name = 'product_list'
```

In both the above method your context variable will be "product_list", and your HTML will be,

```
{% for product in product_list %}
<div class="row">
  <div class="col-md-3 offset-md-2">
    <img src="{{product.product_image}}" class="card" height="150px" />
  </div>
  <div class="col-md-4">
    <h3>{{product.product_name}}</h3>
   .......
  </div>
  <div class="col-md-2">
  .........
  </div>
</div>
{% endfor %}
```

Run code snippet   Expand snippet

Share Follow

answered Oct 23, 2020 at 11:56

Amit Baderia
3,774 ● 3 ● 23 ● 19