

Default Context Naming in `ListView`

In your `SubCategoryListView`, because you are using `ListView` and didn't specify the `context_object_name`, Django will automatically create a context variable named `subcategory_list`, derived from your model name (`SubCategory`). This variable will be passed to the template.

How It Works

- **`ListView`:** By default, `ListView` provides a context variable that holds the list of objects you're querying. If your model is `SubCategory`, Django will use `subcategory_list` as the default context variable name.
- **Template:** In your template, when you use `{% for category in subcategory_list %}`, Django looks for `subcategory_list`, which is provided by the `ListView`.

Example

Here's how Django provides the context automatically:

View Definition:

```
class SubCategoryListView(LoginRequiredMixin, AdminPassesTestMixin,
ListView):
    queryset = SubCategory.objects.all() # This is your queryset
    filterset_class = SubCategoryFilter # Filter class
    template_name = 'product/sub_category_list.html'
    paginate_by = 4

    def get_queryset(self):
        return self.filterset_class(self.request.GET,
queryset=self.queryset).qs

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context["title"] = "Sub-Category List"
        return context
```

Template:

```
{% if subcategory_list %}
    {% for category in subcategory_list %}
        <!-- Use 'category' here to access SubCategory fields -->
    {% endfor %}
{% else %}
    <tr>
        <td colspan="5" class="text-center text-muted">No
sub-categories found.</td>
    </tr>
{% endif %}
```

Customizing Context Variable Name

If you want to customize the context variable name, you can do so by setting the `context_object_name` attribute in your view:

```
class SubCategoryListView(LoginRequiredMixin, AdminPassesTestMixin,
ListView):
    queryset = SubCategory.objects.all()
    filterset_class = SubCategoryFilter
    template_name = 'product/sub_category_list.html'
    paginate_by = 4
    context_object_name = 'my_custom_list' # Custom context name

    def get_queryset(self):
        return self.filterset_class(self.request.GET,
queryset=self.queryset).qs

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context["title"] = "Sub-Category List"
        return context
```

In this case, you'd need to change your template to:

```
{% if my_custom_list %}
    {% for category in my_custom_list %}
        <!-- Loop content -->
    {% endfor %}
{% endif %}
```

Summary

If you haven't set a `context_object_name`, `subcategory_list` will work automatically because it's the default naming convention in Django's `ListView`. If you want a different name, you can customize it by using the `context_object_name` attribute in your view.