Performance Issue: _get_absolute_url() is taking long time with a lot of categories #3373

Edit | New issue

Closed | **Anonymous** opened this issue on 22 May 2020 · 9 comments

Anonymous commented on 22 May 2020 • edited

Issue Summary

When using a new profiling tool PyLive to profile oscar, we find function _get_absolute_url() is taking a lot of time when there are thousands of categories. And time it takes is growing with the number of categories.

Steps to Reproduce

1. Create thousands categories in sandbox
2. Load the page and it will become slower (2-3s to load main page when categories grows to 4000)

Any other relevant information. For example, why do you consider this a bug and what did you expect to happen instead?

The category url does not change over time normally. And it's intensively used in nearly every product page twice (one in side bar, another in the drop-down list 'Browse Store'). So it's naturally that we use cache to cache the url result so we can reuse it over different page.

A simplest way is use Django cache to cache the generated url:

```
def _get_absolute_url(self, parent_slug=None):
    full_url = cache.get("pk="+str(self.pk))
    if full_url is None:
        full_url = reverse('catalogue:category', kwargs={
                'category_slug': self.get_full_slug(parent_slug=parent_slug),  'pk': self.pk
            })
    cache.set("pk="+str(self.pk), full_url)
    return full_url
```

When I test it out with PyLive to profile, it can improve the speed by 3x after cache all the urls.

Technical details

- Python version: Python3.6.9
- Django version: 2.2.12
- Oscar version: 2.0
  We use PyLive for profiling.

Anonymous changed the title ~~Performance Issues:~~ Performance Issues: _get_absolute_url() is taking long time with a lot of categories on 22 May 2020

Anonymous changed the title ~~Performance Issues: _get_absolute_url() is taking long time with a lot of categories~~ Performance Issue: _get_absolute_url() is taking long time with a lot of categories on 22 May 2020

solarissmoke commented on 22 May 2020 — Member

This was fixed in #2895 (Oscar 2.1)

solarissmoke closed this on 22 May 2020

---

rik commented on 22 May 2020 — Contributor

#2895 is already in Oscar 2.0 (see `4fa6709` tag details and `9da707e`)

👍 1

solarissmoke commented on 22 May 2020 — Member

Hmm OK, reopening to investigate further then. The caching was moved to `get_full_slug` so it's not immediately obvious to me why it should have had a performance impact.

solarissmoke reopened this on 22 May 2020

Anonymous commented on 22 May 2020 • edited — Author

Seems like #2895 solves the problem partly by cache the parent slug.
However, when there are a lot of categories, it still takes a long time to generate full url. And the *reverse()* function in *_get_absolute_url()* is also expensive, as mentioned in an issue in django-rest-framework. In our profiling, it takes up to 40% of the total time for executing *_get_absolute_url()*

rik commented on 22 May 2020 — Contributor

I think this issue might be because `oscar/catalogue/browse.html` displays the whole category tree by default. You should customise that template to restrict the depth that makes sense for your shop.

Anonymous commented on 23 May 2020 — Author

> I think this issue might be because `oscar/catalogue/browse.html` displays the whole category tree by default. You should customise that template to restrict the depth that makes sense for your shop.

Thanks for the confirmation and suggestion! I agree with your suggestion.
But I just wonder can we solve this problem directly in oscar to cache the generated url. Since in #2895 oscar already step out to optimize the performance when there are a lot of categories but partially solve this problem, we can cache the url so that we make no further hit on database.

This does not have any negative effect when there is few categories but will help a lot when a lot of categories are added. Once the cache is warm up, it no longer need to access database to build such a category tree any more.

rik commented on 25 May 2020 — Contributor

I'll let the Oscar team decides if they want to cache `Category.get_absolute_url` as well. My opinion is that it is not an issue for most shops so there's no need for it in Oscar.

Remember, you can customise `Category` in your project to add this cache if you want.

😄 1

solarissmoke commented on 25 May 2020 — Member

I tend to agree with **@rik** here. Being too aggressive with caching by default has potential to cause other problems and make debugging them harder.

Also I don't think `get_absolute_url()` is the best place to do this caching. If you have a large category tree, you should probably do template caching in the places you render that tree in its entirety, which will be even more efficient than calling `get_absolute_url()` for thousands of categories (side note - would a tree containing thousands of categories really be usable/desirable to render all at once?). The respective templates in Oscar are easy to override.

😄 1

solarissmoke commented on 1 Aug 2020 — Member

I think Oscar has an appropriate level of caching here - the default category tree in the templates is intended to be used for relatively small number of categories - if you have thousands then I cannot see a situation in which it would make sense to render the whole tree on every view like that - so projects with such structures would be expected to override the applicable templates.

👍 2

solarissmoke closed this on 1 Aug 2020

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked pull requests
Successfully merging a pull request may close this issue.
None yet

Notifications — Customize
Unsubscribe
You're receiving notifications because you authored the thread.

3 participants