DP2-1920-GI-02: Sprint 4 June 1, 2020

**PROFILING AND OPTIMIZATION**

We performed 3 profilings and the corresponding optimizations, all using glowroot.

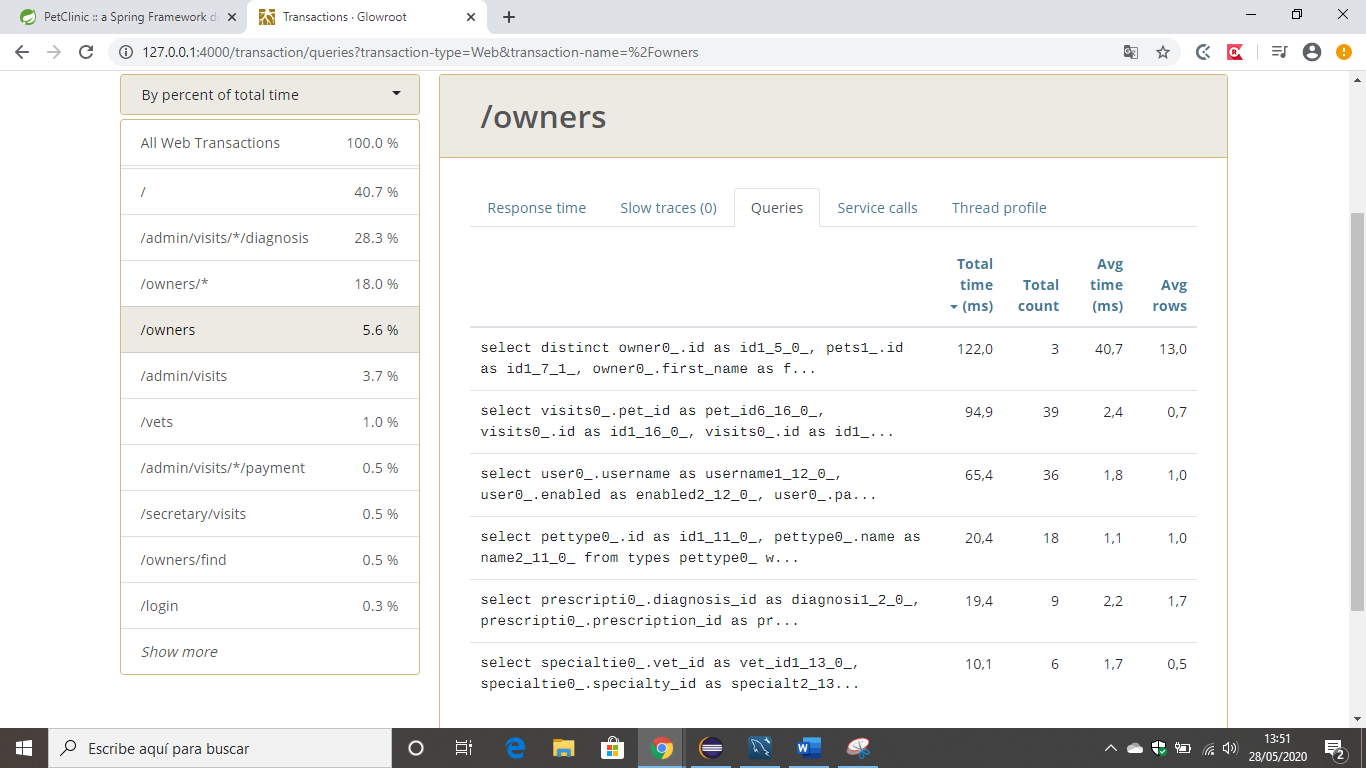
**PROFILING 1**

Description:

A N+1 Query problem has been detected when, by logging in as admin, a search is made for all the owners that exist in our system.

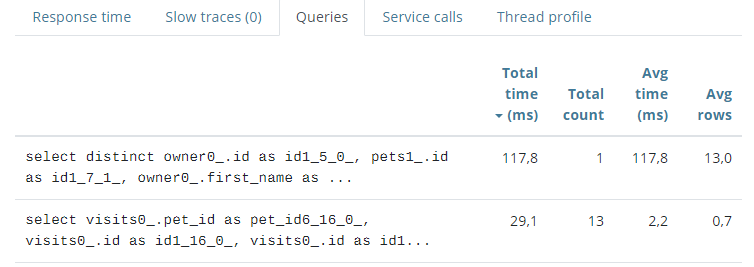
When the view of all owners (/owners) is loaded, all the owners and pets of each one appear. It has been detected that, in that view, for each pet that appears the visits of each one are loaded.

In our system we have 13 pets associated with different owners, so for each pet that we have included in our database, 13 queries are made that return the visits of each pet has.



It can be seen with:

When that view is loaded once:



When that view is loaded twice :

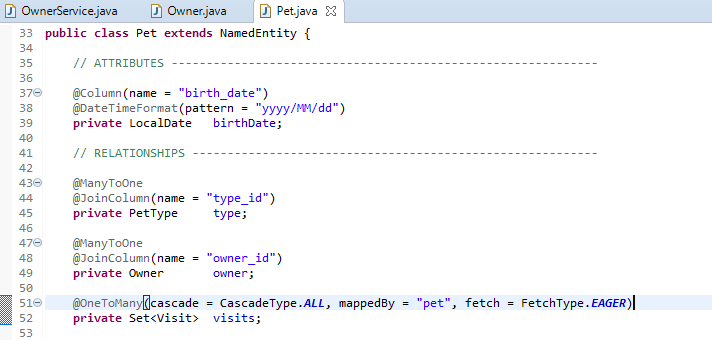


When that view is loaded three times:



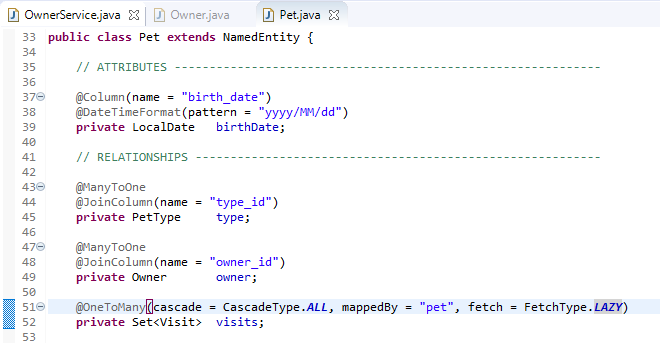
Problem:

It has been detected in the model that the relationship of pet with visits was of type **.EAGER**, which means that whenever a pet is loaded, it’s visits are loaded.

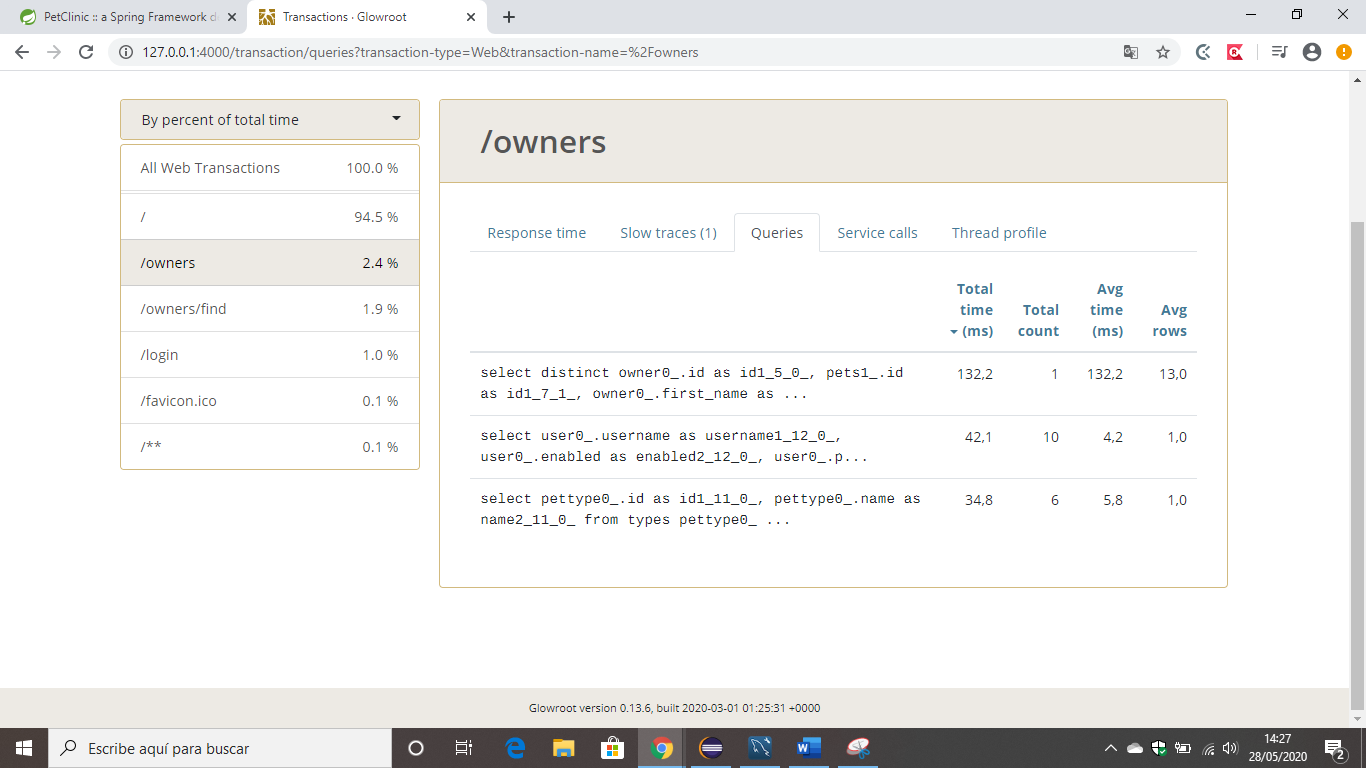


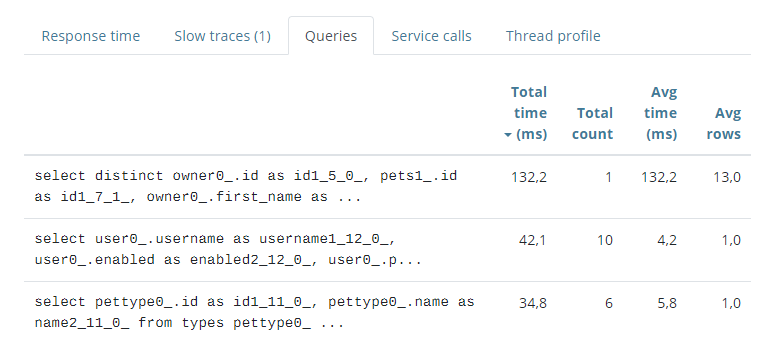
Solution:

It has been changed and not the relationship pet-visits has been set to type .LAZY so that ii only loads when necessary (since visits is something we don’t need in the /owners view we are talking about).



In this way, now in Glowroot you can see that those N Querys that were made for each pet in our database have disappeared.

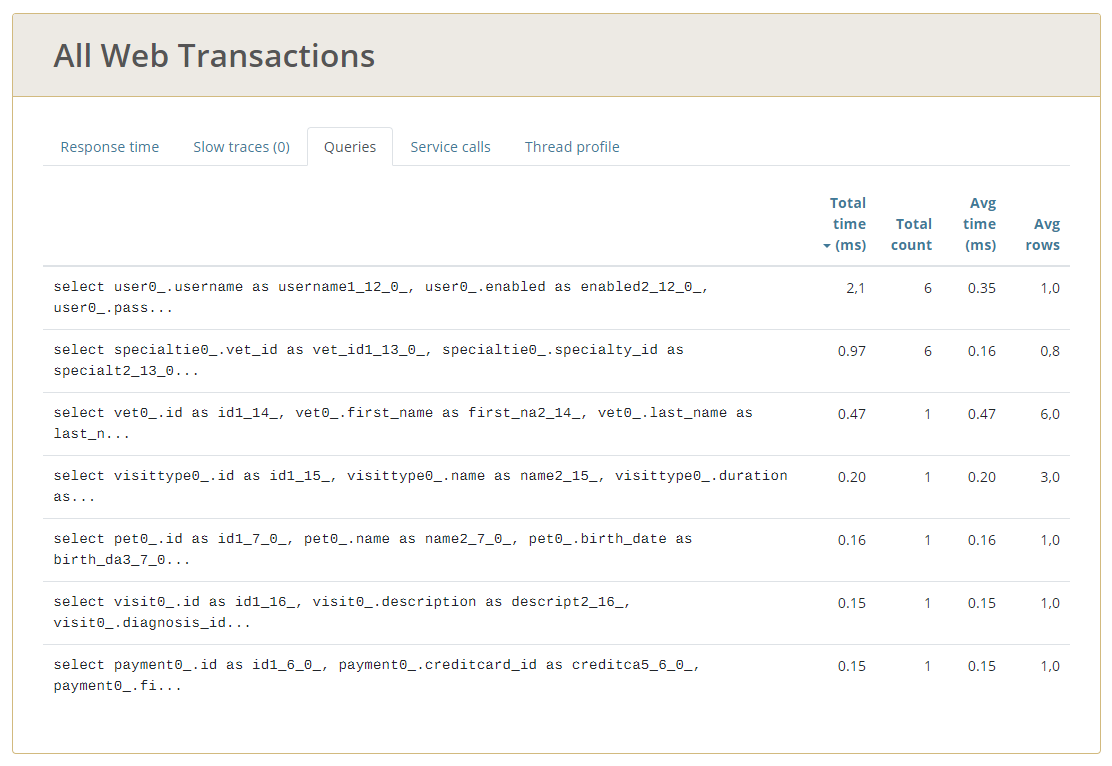




**PROFILING 2**

Situation before:

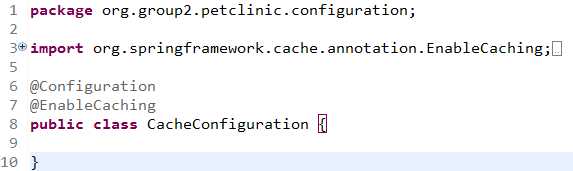
When accessing the view dp2.com/vet/visits/8 as a vet, 7 queries are made to the database, which we consider to be too many:



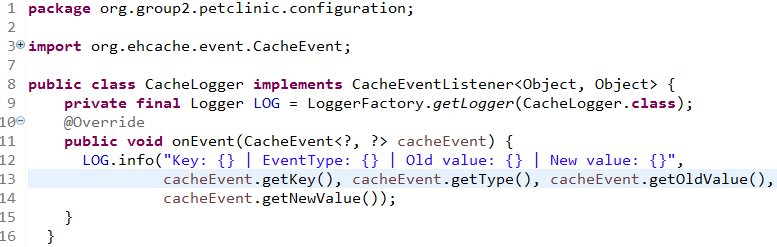
Solution:

We added a cache for findVisitById.

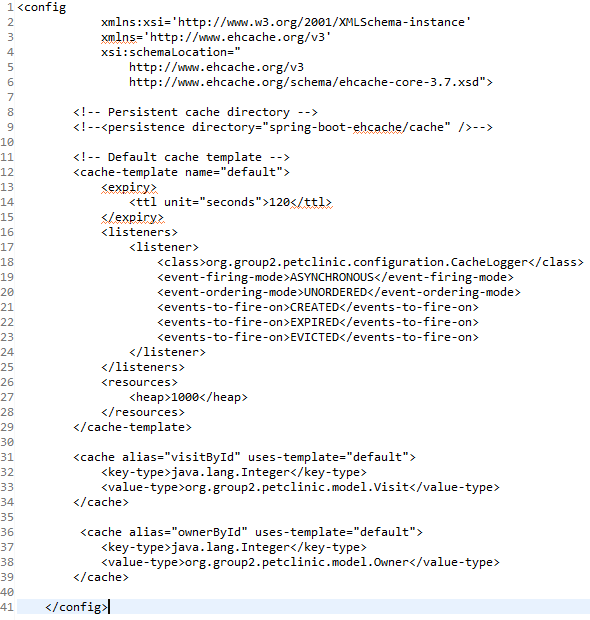
First, we added the cache configuration as explained in the video on EV:



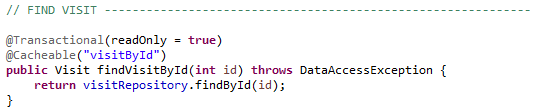
We added a cache logger:

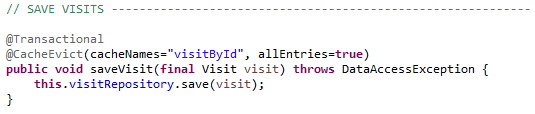


We added the ehcache3 template:



We added the necessary annotations:





Situation after:

Now, 4 queries are made to the database when the view dp2.com/vet/visits/8 is loaded, while previously it was 7. With the cache, we were able to avoid 3 queries.



**PROFILING 3**

Situation before:

When accessing the view dp2.com/owners/1 as an admin, 5 queries are made to the database, even though the data could be stored in a cache:

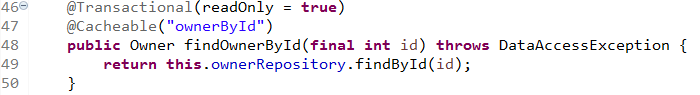


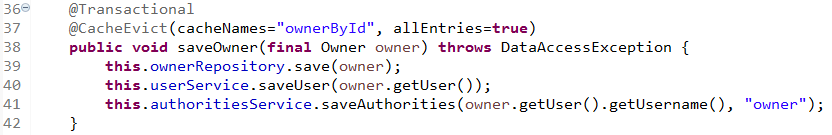
Solution:

We added a cache for findOwnerById.

We did not have to add the cache configuration as we already added it during the previous profiling (profiling 2).

We added the necessary annotations:





Situation after:

With the cache, no more queries are made to the database.

