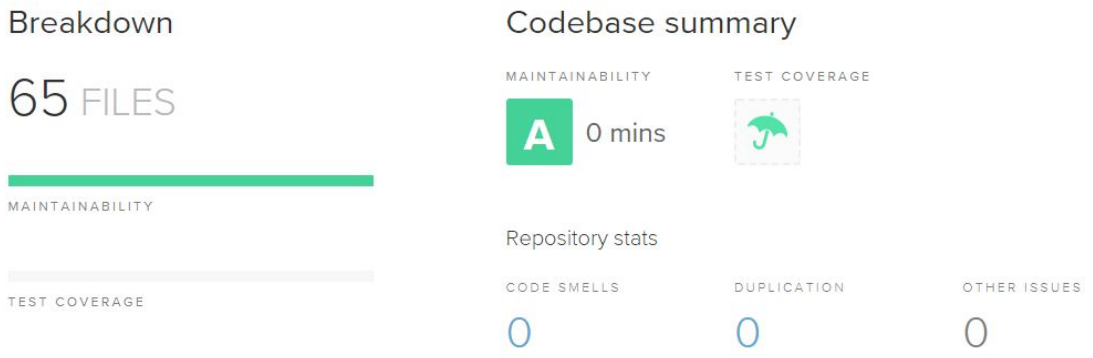
Quality and performance in application

Quality Review (using codeclimate)

We use CodeClimate.com for analyse a code quality. if the quality of the code is not good enough for our needs and for the improvement of the project, we refactor our code.
This is useful for

- the Sustainability of application
- easy development
- Performance (duplicate code)
- security breaches
- clean code architecture

CodeClimate assigns a score between A and D according to the code quality of the project.
In this project, we started from minimum note D. We arrived at note A.
We basically reduced the duplication of the code, and updated the obsolete code, then we factored the code by reducing the lines, and removing the unnecessary functions.
Here is the codeClimate report of the TodoList project



However, we have voluntarily hidden 2 errors messages: messages due to migrations and fixtures, which are not reproducible in a production environment.

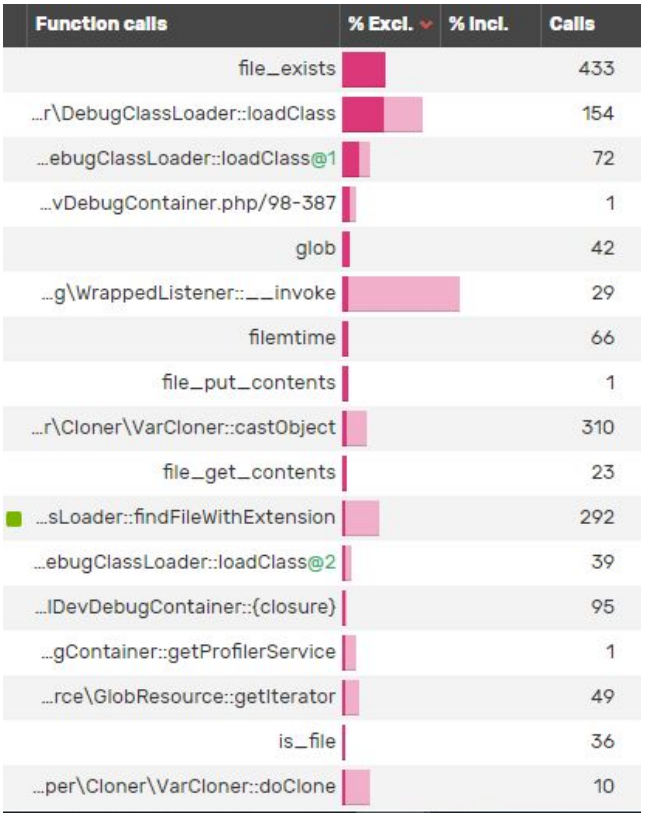
Performance Review (using Blackfire)

We use blackfire because it is a very important utility which works well with the symfony framework.
Blackfire is able to analyze all the routes of our application and give a very detailed report of the performance of the application

Here is a rapport for homepage
https://blackfire.io/profiles/4721b95f-8e93-4f78-9628-ef7bcda98871/graph
For homepage, Here is a basic information:

Wall Time	106 ms		
I/O Wait	n/a		
CPU Time	n/a		
Memory	4.49MB		
Network	n/a	n/a	n/a
SQL	n/a	n/a	

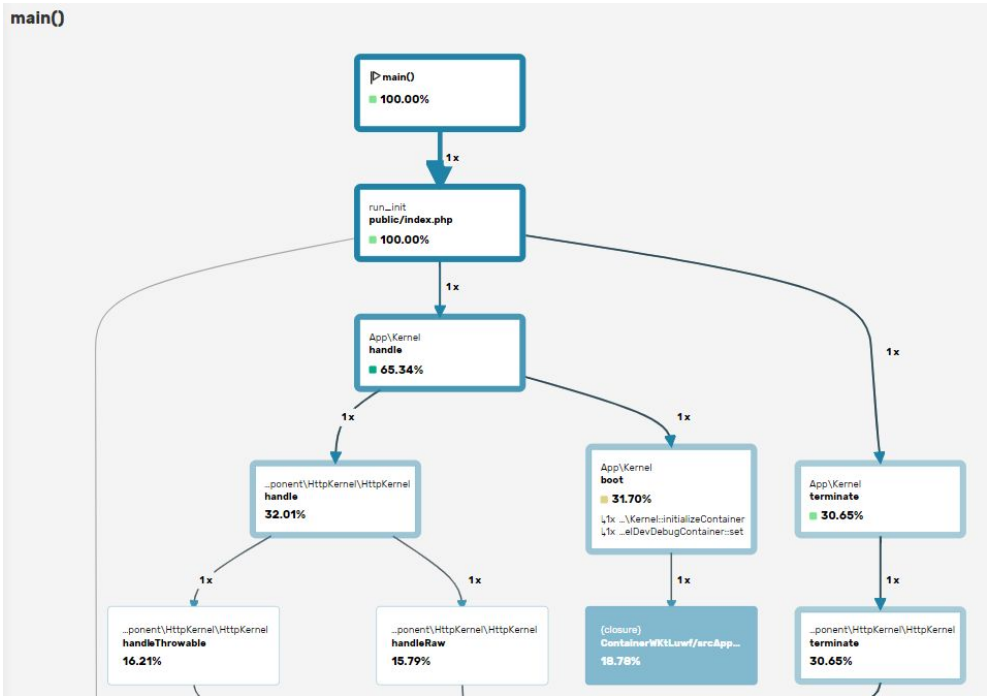
we can get more precise data and diagram. let's take a closer look at the performance of our Todo List website. Let's look at the loading time of our classes

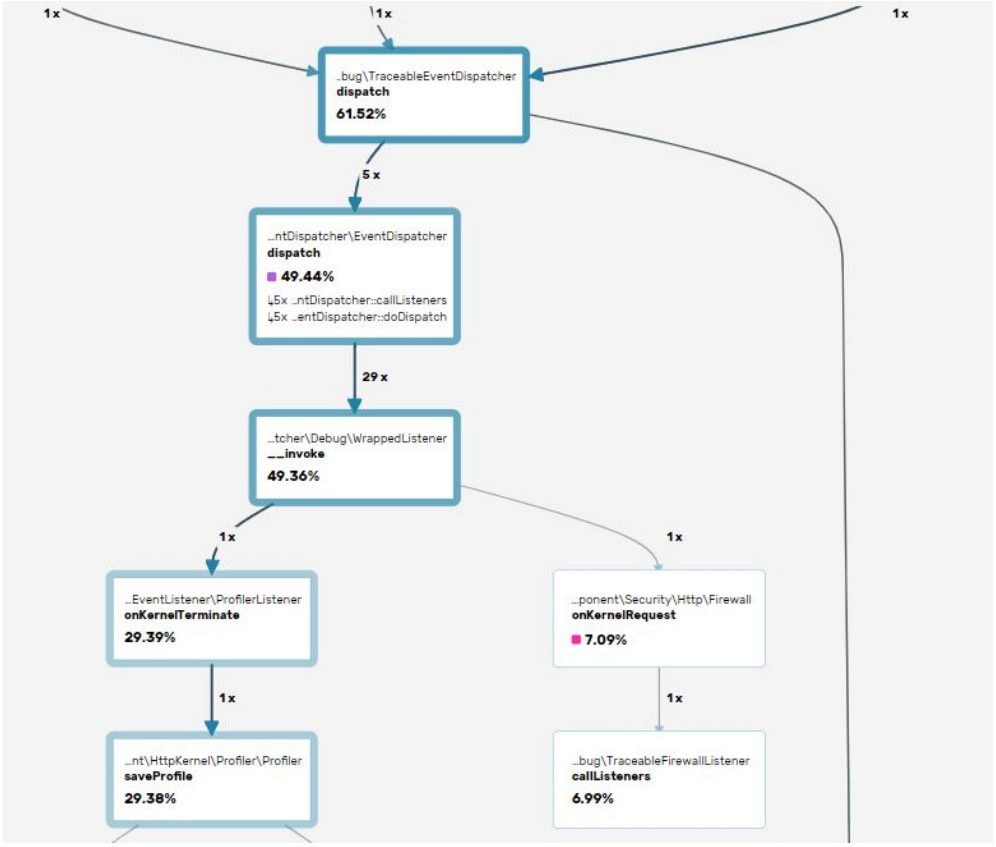


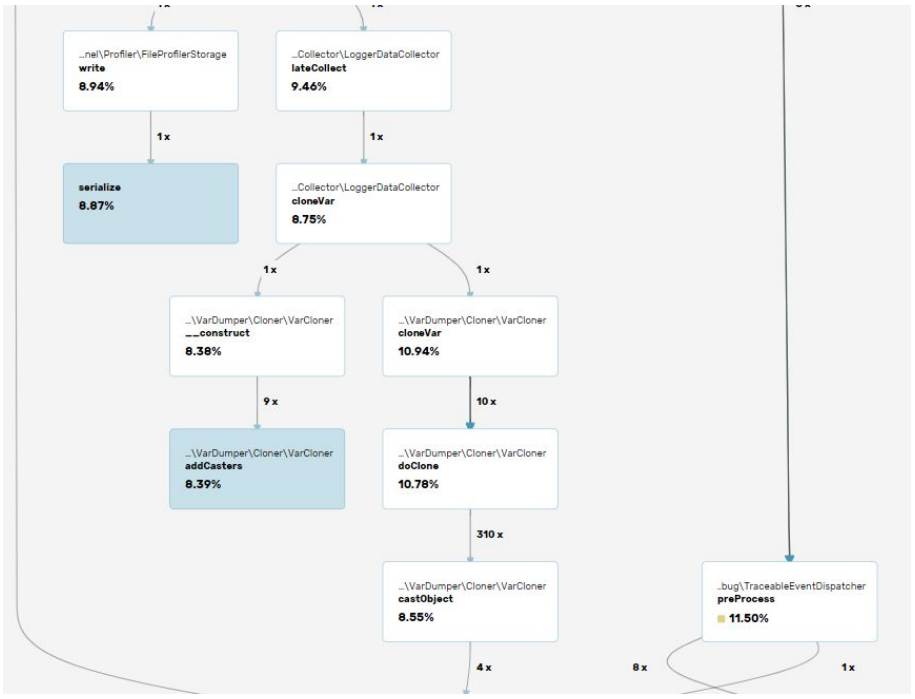
Function calls	% Excl.	% Incl.	Calls
file_exists			433
…r\DebugClassLoader::loadClass			154
…ebugClassLoader::loadClass@1			72
…vDebugContainer.php/98-387			1
glob			42
…g\WrappedListener::__invoke			29
filemtime			66
file_put_contents			1
…r\Cloner\VarCloner::castObject			310
file_get_contents			23
…sLoader::findFileWithExtension			292
…ebugClassLoader::loadClass@2			39
…lDevDebugContainer::{closure}			95
…gContainer::getProfilerService			1
…rce\GlobResource::getIterator			49
is_file			36
…per\Cloner\VarCloner::doClone			10

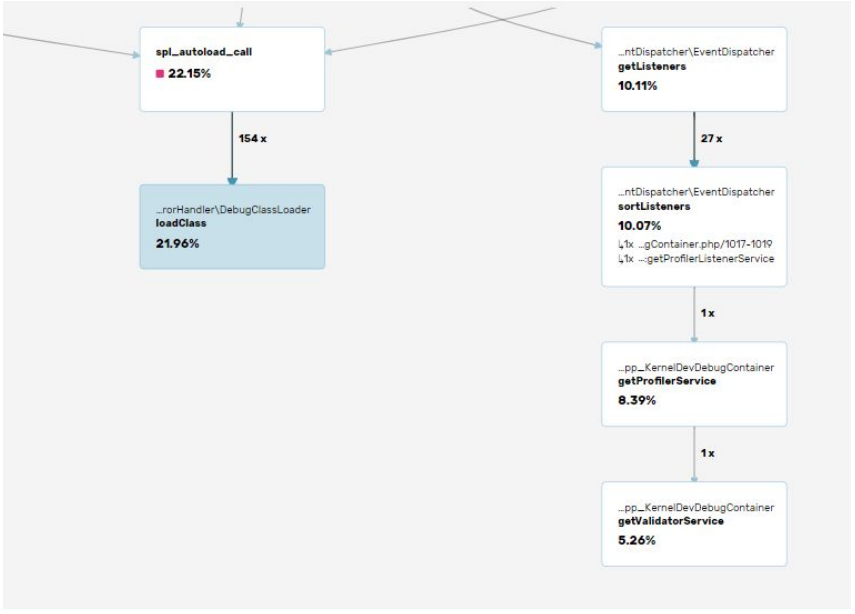
Here is the memory allocated to load the application

We can also look at the loading diagram of the website






...nel\Profiler\FileProfilerStorage
write
8.94%
...Collector\LoggerDataCollector
lateCollect
9.46%
1 x
1 x
serialize
8.87%
...Collector\LoggerDataCollector
cloneVar
8.75%
1 x
1 x
...\VarDumper\Cloner\VarCloner
__construct
8.38%
...\VarDumper\Cloner\VarCloner
cloneVar
10.94%
9 x
10 x
...\VarDumper\Cloner\VarCloner
addCasters
8.39%
...\VarDumper\Cloner\VarCloner
doClone
10.78%
310 x
...\VarDumper\Cloner\VarCloner
castObject
8.55%
...bug\TraceableEventDispatcher
preProcess
11.50%
4 x
8 x
1 x


spl_autoload_call
22.15%
154 x
...rorHandler\DebugClassLoader
loadClass
21.96%
...ntDispatcher\EventDispatcher
getListeners
10.11%
27 x
...ntDispatcher\EventDispatcher
sortListeners
10.07%
↳1x ...gContainer.php/1017-1019
↳1x ...::getProfilerListenerService
1 x
...pp_KernelDevDebugContainer
getProfilerService
8.39%
1 x
...pp_KernelDevDebugContainer
getValidatorService
5.26%