Audit du projet

Contenu

[Technologies actuellement utilisées 2](#_Toc488609409)

[Sources d’informations utiles 2](#_Toc488609410)

[Améliorations techniques 2](#_Toc488609411)

[Création d’un projet 2](#_Toc488609412)

[Rendre les contrôleurs plus légers 2](#_Toc488609413)

[Accélérateur PHP 3](#_Toc488609414)

[Optimiser l’autoloader 3](#_Toc488609415)

[Utilisation d’un cache Doctrine 3](#_Toc488609416)

[Utitilisation de Monolog 3](#_Toc488609417)

[Personnaliser les pages d’erreurs 4](#_Toc488609418)

[Améliorations fonctionnelles 4](#_Toc488609419)

[Pouvoir consulter la liste des utilisateurs 4](#_Toc488609420)

[Pouvoir revenir à la page d’accueil 4](#_Toc488609421)

[Mesure actuelle des performances du site 4](#_Toc488609422)

[Le détail complet des métriques 4](#_Toc488609423)

[Login 4](#_Toc488609424)

[Les métriques de performance : 4](#_Toc488609425)

[Les métriques « Twig » 5](#_Toc488609426)

[Les métriques « doctrine » 6](#_Toc488609427)

[Lister les utilisateurs 6](#_Toc488609428)

[Les métriques de performance : 6](#_Toc488609429)

[Les métriques « Twig » 7](#_Toc488609430)

[Les métriques « doctrine » 8](#_Toc488609431)

[Lister les tâches 8](#_Toc488609432)

[Les métriques de performance : 8](#_Toc488609433)

[Les métriques « Twig » 9](#_Toc488609434)

[Les métriques « doctrine » 9](#_Toc488609435)

[Ajouter une tâche 10](#_Toc488609436)

[Les métriques de performance : 10](#_Toc488609437)

[Les métriques « Twig » 11](#_Toc488609438)

[Les métriques « doctrine » 12](#_Toc488609439)

# Technologies actuellement utilisées

Le projet utilise le framework « Symfony 3.3 » et les composants suivants :

* Doctrine (gestion des échanges avec la base de données)
* Security (composant qui prend en charge la sécurité du site)
* Twig (pour la gestion des vues)
* Monolog (pour produire des logs en production, non utilisés pour le moment)
* SwiftMailer

Il est important d’utiliser les dernières versions stables de chacun de ces composants ou, a minima, les dernières versions encore supportées par les équipes de développements.

# Sources d’informations utiles

Pour maintenir le site et éviter toute dette technique, il est utile de suivre les sources d’informations suivantes :

<http://symfony.com/blog/>

<http://symfony.com/blog/category/security-advisories> (être très attentif aux aspects Sécurité du site).

https://blog.blackfire.io/ (orienté performance)

Le site de Sensio et le fils twitter (<https://twitter.com/phpc>) sont également des sources d’informations intéressantes pour se maintenir à jour sur les technologies PHP.

Les messages de log de type « dépréciations » sont également une source d’informations sur le fait que l’application n’utilise pas les API de dernière génération.

# Améliorations techniques

## Création d’un projet

Tout projet doit être créé avec la commande « symfony » suivante :

php symfony new todolist

Les installations de bundle externes doivent être faites avec l’utilitaire « Composer », afin de s’affranchir de tout problème de dépendance entre composants « Symfony ».

## Gestion de la soumission d’un formulaire

Avant l’appel à la méthode « isValid() », il est préférable d’appeler la méthode « isSubmitted() »,. L’appel de la méthode « isValid() » avec un formulaire non soumis est obsolète depuis la version 3.2 de « Symfony » et lancera une exception en version 4.

## 

## Personnaliser les pages d’erreurs

Il est important de personnaliser les pages d’erreurs, en créant des fichiers twig par code erreur html et un fichier twig générique.

Dans le répertoire App/Resources/TwigBundle/views/Exceotion créer les fichiers suivants :

error[code\_erreur].html.twig

error.html.twig

## Utilisation du composant Cache de symfony

Afin de limiter les accès base de données, il peut intéressant d’utiliser un cache de données :

https://symfony.com/blog/new-in-symfony-3-1-cache-component

## 

## Eviter la création de variable

Pour améliorer la performance de l’application, il faut éviter de créer des variables si cela n’est pas nécessaire.

## Rendre les contrôleurs plus légers

Il est préférable de créer deux classes services avec injection de dépendance afin de déplacer le code métier des contrôleurs vers des services métiers.

## 

## Accélérateur PHP

Pour accélérer les performances de l’application en production, il est recommandé d’installer un accélérateur PHP comme APC :

https://doc.ubuntu-fr.org/php-apc

## Optimiser l’autoloader

Par défaut, l’autoloader n’est pas optimisé. Pour utiliser un autoloader optimisé il faut utiliser la commande suivante :

php composer.phar dump-autoload --optimize

## 

## Utilisation d’un cache Doctrine

L’utilisation du bundle Symfony « DoctrineCacheBundle » permet d’optimiser les accès base de données (attention ce bundle n’est pas encore compatible avec la version 3.3.\* de Symfony).

Il faut installer le bundle avec la commande suivante :

composer require doctrine/doctrine-cache-bundle

puis activer le bundle en ajoutant dans le fichier app/AppKernel.php la ligne suivante :

new Doctrine\Bundle\DoctrineCacheBundle\DoctrineCacheBundle(),

et enfin configurer le bundle (app/config/config.yml):

# app/config/config.yml

doctrine\_cache:

providers:

my\_apc\_metadata\_cache:

type: apc

namespace: metadata\_cache\_ns

my\_apc\_query\_cache:

namespace: query\_cache\_ns

apc: ~

## 

## Utitilisation de Monolog

Le composant Symfony « monolog » est utile pour suivre et gérer l’application en production :

https://symfony.com/doc/current/logging.html

# Améliorations fonctionnelles

## Pouvoir consulter la liste des utilisateurs

Pour afficher la page d’accès à la liste des utilisateurs il faut saisir l’adresse dans la barre d’adresse du navigateur

## Pouvoir revenir à la page d’accueil

Le lien sur ‘To do List App’ ne renvoie vers rien, alors que l’utilisateur s’attendrait à revenir à la page d’accueil.

# Mesure actuelle des performances du site

La mesure des performances est faite avec l’outil ‘Symfony Profiler’ accessible en mode dev ; le diagramme d’exécution est accessible depuis l’onglet ‘Performance’.

J’ai établi les métriques du site à partir de quatre cas d’utilisation : la connexion, la consultation des utilisateurs, la consultation des tâches et la création d’une tâche.

Sur chacun de ces scenarii, j’ai retenu différentes métriques : les métriques de performance, les métriques liées à la génération des vue (Twig) et les métriques liées aux accès base de données (Doctrine).

Les métriques de performance : le temps d’exécution, le pic de consommation mémoire, la liste des objets instanciés.

Les métriques « Twig » : le temps d’exécution, le nombre de template utilisés, le nombre de blocks appelés et enfin le nombre de macro utilisé.

Les métriques « Doctine » : le nombre de requêtes exécutées, le nombre de requêtes différentes exécutées, le temps d’exécution global, le temps d’exécution par requête.

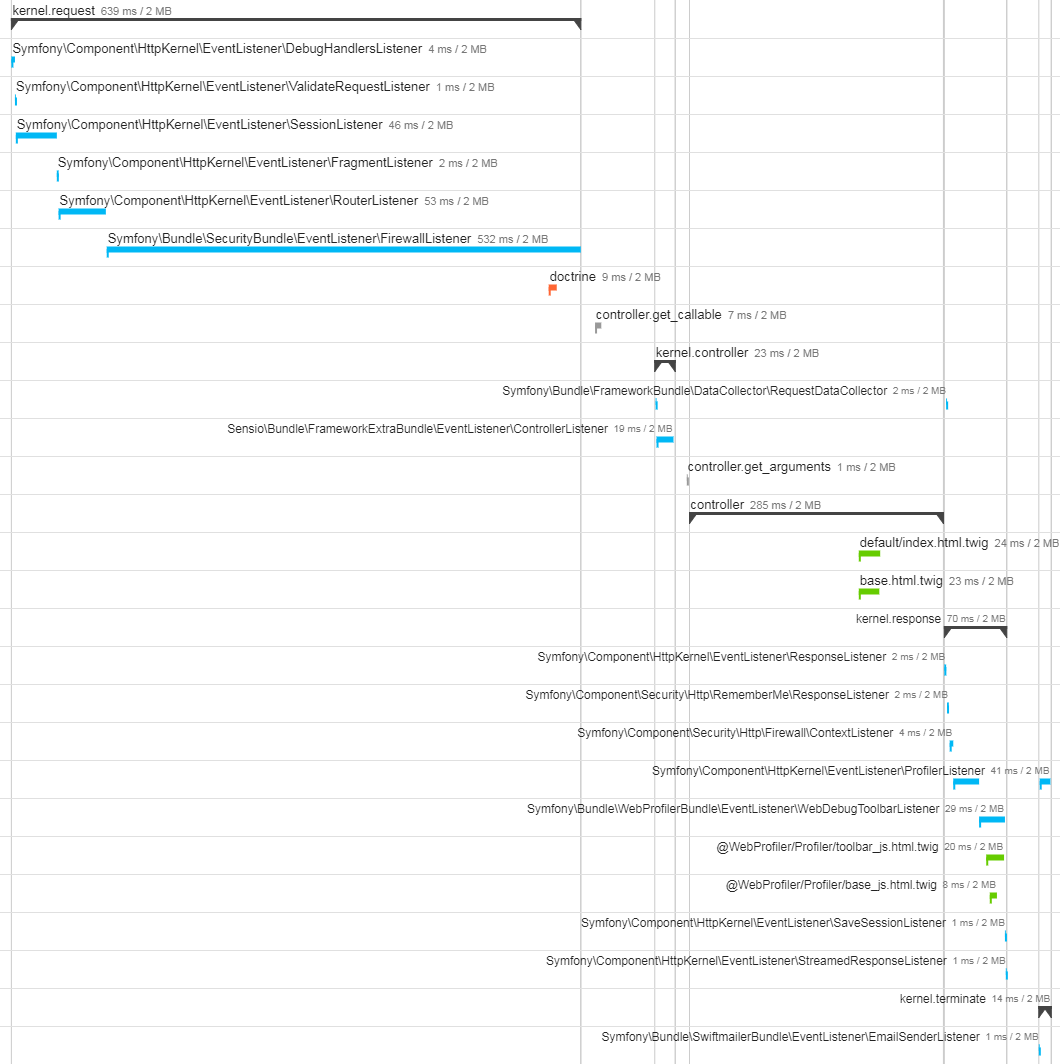
# Le détail complet des métriques

## Login

### Les métriques de performance :

|  |  |
| --- | --- |
| Temps total d’exécution | 1398ms |
| Pic mémoire | 2.00MB |

Graphe d’appel :



### Les métriques « Twig »

|  |  |
| --- | --- |
| Temps d’exécution | 52ms |
| Nombre de vues « Twig » utilisées | 5 |
| Nombre de blocs « twig » utilisés | 5 |
| Nombre de macros | 0 |

Graphe d’appel :

main 52.01ms/100%

└ default/index.html.twig 29.00ms/56%

│ └ base.html.twig 28.00ms/54%

│ └ default/index.html.twig::block(header\_title)

│ └ base.html.twig::block(header\_img)

│ └ default/index.html.twig::block(body)

└ @WebProfiler/Profiler/toolbar\_js.html.twig 23.00ms/44%

└ @WebProfiler/Profiler/base\_js.html.twig 8.00ms/15%

└ @WebProfiler/Profiler/toolbar.css.twig

### Les métriques « doctrine »

|  |  |
| --- | --- |
| Nombre de requêtes exécutées | 1 |
| Nombre de requêtes différentes exécutées | 1 |
| Le temps d’exécution global | 4.00ms |

Le temps d’exécution par requête.

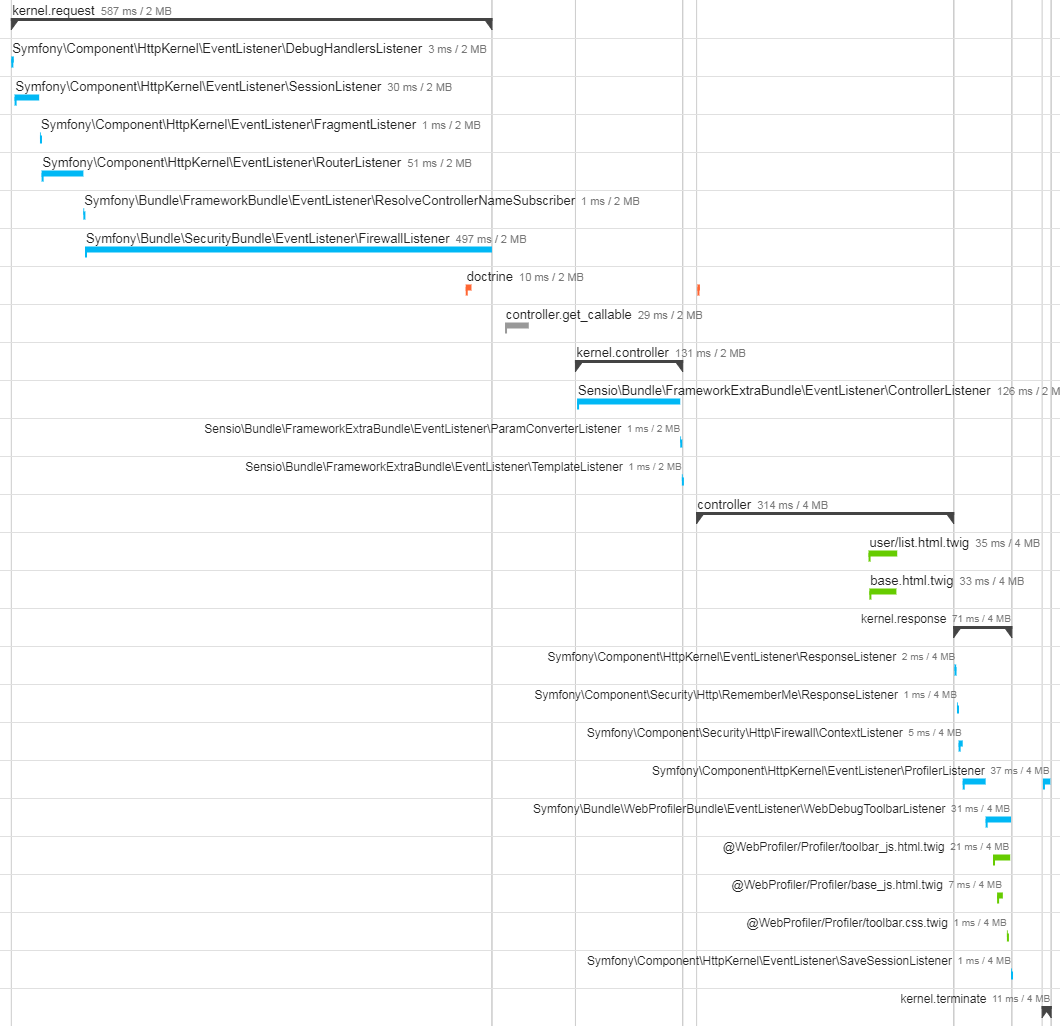
|  |  |
| --- | --- |
| Temps | Requêtes |
| 4.00ms | **SELECT** t0.id **AS** id\_1, t0.username **AS** username\_2, t0.password **AS** password\_3, t0.email **AS** email\_4, t0.roles **AS** roles\_5 **FROM** user t0 **WHERE** t0.id = ?  Parameters: [0 => 1] |

## Lister les utilisateurs

### Les métriques de performance :

|  |  |
| --- | --- |
| Temps total d’exécution | 1506ms |
| Pic mémoire | 4.00MB |

Graphe d’appel :



### Les métriques « Twig »

|  |  |
| --- | --- |
| Temps d’exécution | 55ms |
| Nombre de vues « Twig » utilisées | 5 |
| Nombre de blocs « twig » utilisés | 3 |
| Nombre de macros | 0 |

Graphe d’appel :

main 55.01ms/100%

└ user/list.html.twig 34.00ms/62%

│ └ base.html.twig 33.00ms/60%

│ └ user/list.html.twig::block(header\_title)

│ └ user/list.html.twig::block(header\_img)

│ └ user/list.html.twig::block(body) 6.00ms/11%

└ @WebProfiler/Profiler/toolbar\_js.html.twig 21.00ms/38%

└ @WebProfiler/Profiler/base\_js.html.twig 8.00ms/15%

└ @WebProfiler/Profiler/toolbar.css.twig

### Les métriques « doctrine »

|  |  |
| --- | --- |
| Nombre de requêtes exécutées | 2 |
| Nombre de requêtes différentes exécutées | 2 |
| Le temps d’exécution global | 4.00ms |

Le temps d’exécution par requête.

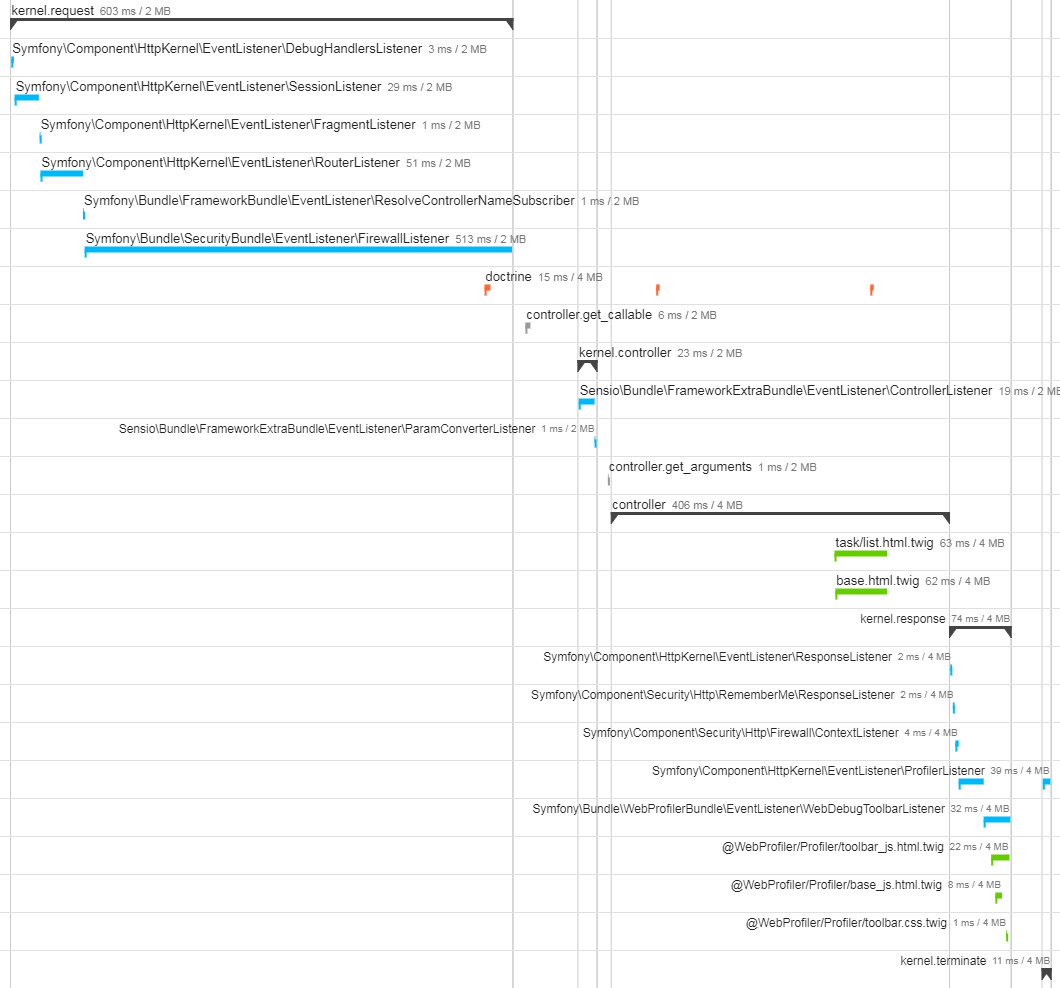
|  |  |
| --- | --- |
| Temps | Requêtes |
| 4.00ms | **SELECT** t0.id **AS** id\_1, t0.username **AS** username\_2, t0.password **AS** password\_3, t0.email **AS** email\_4, t0.roles **AS** roles\_5 **FROM** user t0 **WHERE** t0.id = ?  Parameters: [0 => 1] |
| 0.00ms | **SELECT** t0.id **AS** id\_1, t0.username **AS** username\_2, t0.password **AS** password\_3, t0.email **AS** email\_4, t0.roles **AS** roles\_5 **FROM** user t0  Parameters: [] |

## Lister les tâches

### Les métriques de performance :

|  |  |
| --- | --- |
| Temps total d’exécution | 1481ms |
| Pic mémoire | 4.00MB |

Graphe d’appel :



### Les métriques « Twig »

|  |  |
| --- | --- |
| Temps d’exécution | 85ms |
| Nombre de vues « Twig » utilisées | 5 |
| Nombre de blocs « twig » utilisés | 3 |
| Nombre de macros | 0 |

Graphe d’appel :

main 85.01ms/100%

└ task/list.html.twig 63.01ms/74%

│ └ base.html.twig 62.01ms/73%

│ └ base.html.twig::block(header\_title)

│ └ task/list.html.twig::block(header\_img) 1.00ms/1%

│ └ task/list.html.twig::block(body) 33.00ms/39%

└ @WebProfiler/Profiler/toolbar\_js.html.twig 22.00ms/26%

└ @WebProfiler/Profiler/base\_js.html.twig 8.00ms/9%

└ @WebProfiler/Profiler/toolbar.css.twig

### Les métriques « doctrine »

|  |  |
| --- | --- |
| Nombre de requêtes exécutées | 3 |
| Nombre de requêtes différentes exécutées | 2 |
| Le temps d’exécution global | 6.00ms |

Le temps d’exécution par requête.

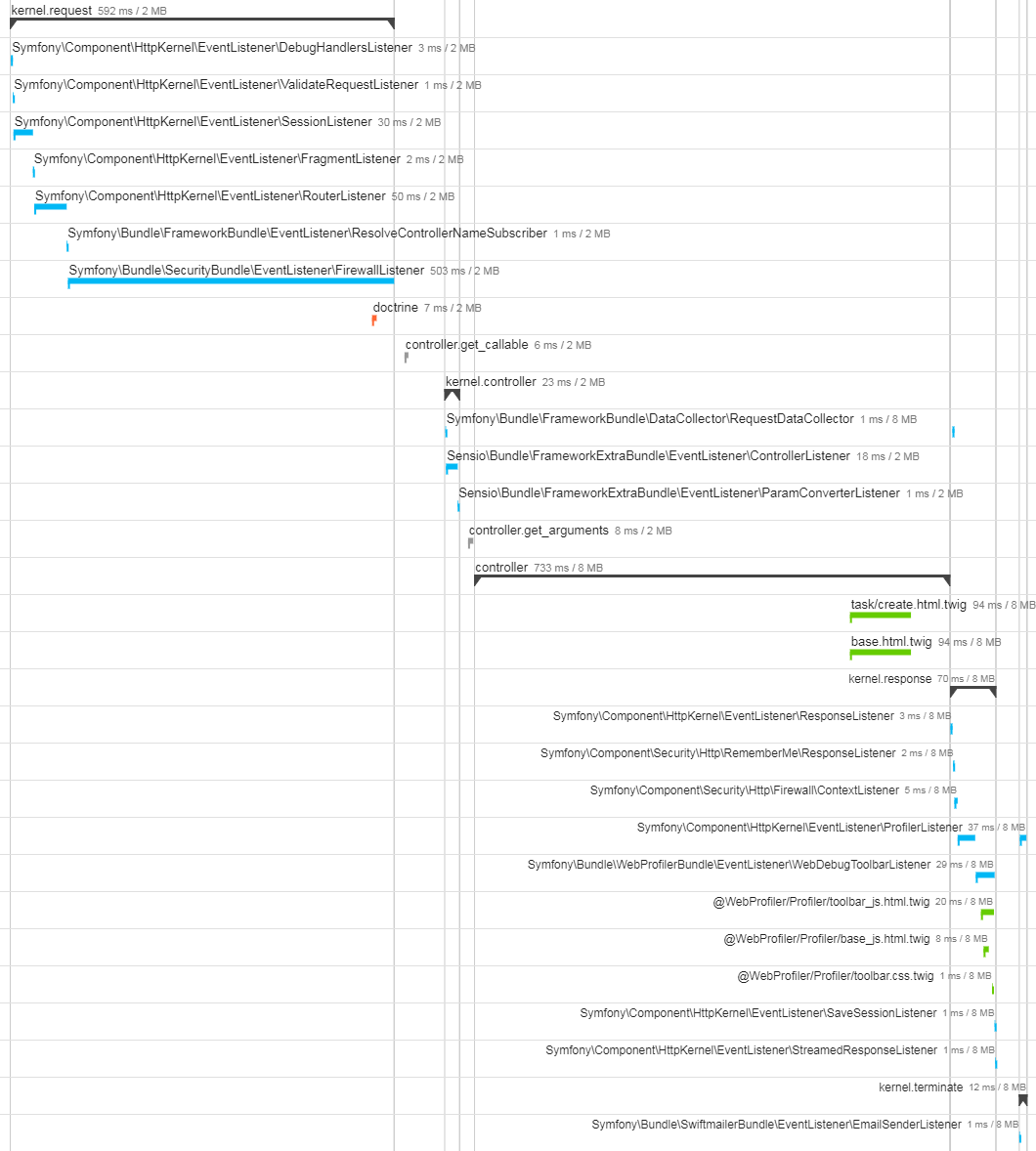
|  |  |
| --- | --- |
| Temps | Requêtes |
| 3.00ms | **SELECT** t0.id **AS** id\_1, t0.username **AS** username\_2, t0.password **AS** password\_3, t0.email **AS** email\_4, t0.roles **AS** roles\_5 **FROM** user t0 **WHERE** t0.id = ?  Parameters: [0 => 1] |
| 1.00ms | **SELECT** t0.id **AS** id\_1, t0.created\_at **AS** created\_at\_2, t0.title **AS** title\_3, t0.content **AS** content\_4, t0.is\_done **AS** is\_done\_5, t0.user\_id **AS** user\_id\_6 **FROM** task t0  Parameters: [] |
| 2.00ms | **SELECT** t0.id **AS** id\_1, t0.username **AS** username\_2, t0.password **AS** password\_3, t0.email **AS** email\_4, t0.roles **AS** roles\_5 **FROM** user t0 **WHERE** t0.id = ?  Parameters: [0 => 4] |

## Ajouter une tâche

### Les métriques de performance :

|  |  |
| --- | --- |
| Temps total d’exécution | 1797ms |
| Pic mémoire | 8.00MB |

Graphe d’appel :



### Les métriques « Twig »

|  |  |
| --- | --- |
| Temps d’exécution | 116ms |
| Nombre de vues « Twig » utilisées | 5 |
| Nombre de blocs « twig » utilisés | 38 |
| Nombre de macros | 0 |

Graphe d’appel :

main 116.01ms/100%

└ task/create.html.twig 95.01ms/82%

│ └ base.html.twig 94.01ms/81%

│ └ base.html.twig::block(header\_title)

│ └ base.html.twig::block(header\_img)

│ └ task/create.html.twig::block(body) 68.01ms/59%

│ └ form\_div\_layout.html.twig::block(form\_start)

│ └ form\_div\_layout.html.twig::block(form\_widget) 33.00ms/28%

│ │ └ form\_div\_layout.html.twig::block(form\_widget\_compound) 32.00ms/28%

│ │ └ form\_div\_layout.html.twig::block(widget\_container\_attributes)

│ │ │ └ form\_div\_layout.html.twig::block(attributes)

│ │ └ bootstrap\_3\_layout.html.twig::block(form\_errors)

│ │ └ form\_div\_layout.html.twig::block(form\_rows) 26.00ms/22%

│ │ │ └ bootstrap\_3\_layout.html.twig::block(form\_row) 10.00ms/9%

│ │ │ │ └ bootstrap\_3\_layout.html.twig::block(form\_label) 3.00ms/3%

│ │ │ │ │ └ form\_div\_layout.html.twig::block(form\_label) 2.00ms/2%

│ │ │ │ │ └ form\_div\_layout.html.twig::block(attributes)

│ │ │ │ └ form\_div\_layout.html.twig::block(form\_widget) 3.00ms/3%

│ │ │ │ │ └ bootstrap\_3\_layout.html.twig::block(form\_widget\_simple) 3.00ms/3%

│ │ │ │ │ └ form\_div\_layout.html.twig::block(form\_widget\_simple) 2.00ms/2%

│ │ │ │ │ └ form\_div\_layout.html.twig::block(widget\_attributes)

│ │ │ │ │ └ form\_div\_layout.html.twig::block(attributes)

│ │ │ │ └ bootstrap\_3\_layout.html.twig::block(form\_errors)

│ │ │ └ bootstrap\_3\_layout.html.twig::block(form\_row) 9.00ms/8%

│ │ │ │ └ bootstrap\_3\_layout.html.twig::block(form\_label) 3.00ms/3%

│ │ │ │ │ └ form\_div\_layout.html.twig::block(form\_label) 2.00ms/2%

│ │ │ │ │ └ form\_div\_layout.html.twig::block(attributes)

│ │ │ │ └ bootstrap\_3\_layout.html.twig::block(textarea\_widget) 2.00ms/2%

│ │ │ │ │ └ form\_div\_layout.html.twig::block(textarea\_widget) 2.00ms/2%

│ │ │ │ │ └ form\_div\_layout.html.twig::block(widget\_attributes)

│ │ │ │ │ └ form\_div\_layout.html.twig::block(attributes)

│ │ │ │ └ bootstrap\_3\_layout.html.twig::block(form\_errors)

│ │ │ └ form\_div\_layout.html.twig::block(hidden\_row) 4.00ms/3%

│ │ │ └ form\_div\_layout.html.twig::block(hidden\_widget) 3.00ms/3%

│ │ │ └ bootstrap\_3\_layout.html.twig::block(form\_widget\_simple) 2.00ms/2%

│ │ │ └ form\_div\_layout.html.twig::block(form\_widget\_simple) 2.00ms/2%

│ │ │ └ form\_div\_layout.html.twig::block(widget\_attributes) 1.00ms/1%

│ │ │ └ form\_div\_layout.html.twig::block(attributes) 1.00ms/1%

│ │ └ form\_div\_layout.html.twig::block(form\_rest) 1.00ms/1%

│ └ form\_div\_layout.html.twig::block(form\_end) 3.00ms/3%

│ └ form\_div\_layout.html.twig::block(form\_rest) 1.00ms/1%

└ @WebProfiler/Profiler/toolbar\_js.html.twig 21.00ms/18%

└ @WebProfiler/Profiler/base\_js.html.twig 8.00ms/7%

└ @WebProfiler/Profiler/toolbar.css.twig

└ @WebProfiler/Profiler/base\_js.html.twig 7.00ms/8%

└ @WebProfiler/Profiler/toolbar.css.twig

### Les métriques « doctrine »

|  |  |
| --- | --- |
| Nombre de requêtes exécutées | 1 |
| Nombre de requêtes différentes exécutées | 1 |
| Le temps d’exécution global | 4.00ms |

Le temps d’exécution par requête.

|  |  |
| --- | --- |
| Temps | Requêtes |
| 4.00ms | **SELECT** t0.id **AS** id\_1, t0.username **AS** username\_2, t0.password **AS** password\_3, t0.email **AS** email\_4, t0.roles **AS** roles\_5 **FROM** user t0 **WHERE** t0.id = ?  Parameters: [0 => 1] |