



# [AUDIT DE QUALITE DU CODE & PERFORMANCE DE L'APPLICATION]

---

[Améliorez une application existante de ToDo & Co]

Créé par  
ALKHALLOUFI Fatima  
Avril 2024

ToDo & Co

## Table des matières

[Améliorez une application existante de ToDo & Co] .....	1
I. Analyse du code initial.....	3
II. Correction d'anomalies et amélioration de l'application.....	4
III. Résultats de l'audit final.....	8
IV. audit de performance Symfony profiler .....	9

## I. Analyse du code initial

### 1. Déploiement de l'application

Une fois l'application récupérée depuis github, nous avons constaté:

- des bugs au lancement de l'application :

Whoops, looks like something went wrong.

1/1

FatalThrowableError in ErrorHandler.php line 364:

Type error: Too few arguments to function  
Symfony\Component\Debug\ErrorHandler::handleError(), 4 passed in C:\wamp\www\projet8-TodoList\app\AppKernel.php on line 6 and at least 5 expected

1. in ErrorHandler.php line 364
2. at ErrorHandler->handleError('8192', 'AppKernel implements the Serializable interface, which is deprecated. Implement \_\_serialize() and \_\_unserialize() instead (or in addition, if support for old PHP versions is necessary)', 'C:\wamp\www\projet8-TodoList\app\AppKernel.php', '6') in AppKernel.php line 6
3. at require\_once('C:\wamp\www\projet8-TodoList\app\AppKernel.php') in DebugClassLoader.php line 142
4. at DebugClassLoader->loadClass('AppKernel') in app\_dev.php line 25

- la dépréciation de la version du Symfony et des composants du framework :

```
"php": ">=5.5.9",
"symfony/symfony": "3.1.*",
"doctrine/orm": "^2.5",
"doctrine/doctrine-bundle": "^1.6",
"doctrine/doctrine-cache-bundle": "^1.2",
"symfony/swiftmailer-bundle": "^2.3",
"symfony/monolog-bundle": "^2.8",
"symfony/polyfill-apcu": "^1.0",
"sensio/distribution-bundle": "^5.0",
"sensio/framework-extra-bundle": "^3.0.2",
"incenteev/composer-parameter-handler": "^2.0"
```

la version 3.1 utilisé dans le cadre de notre projet n'est plus maintenue depuis 2017.



**Symfony 3.1**

UNMAINTAINED

Requires: **PHP 5.5.9** or higher    Latest patch version: 3.1.10

Released on: May 2016    End of support: July 2017



Source Code



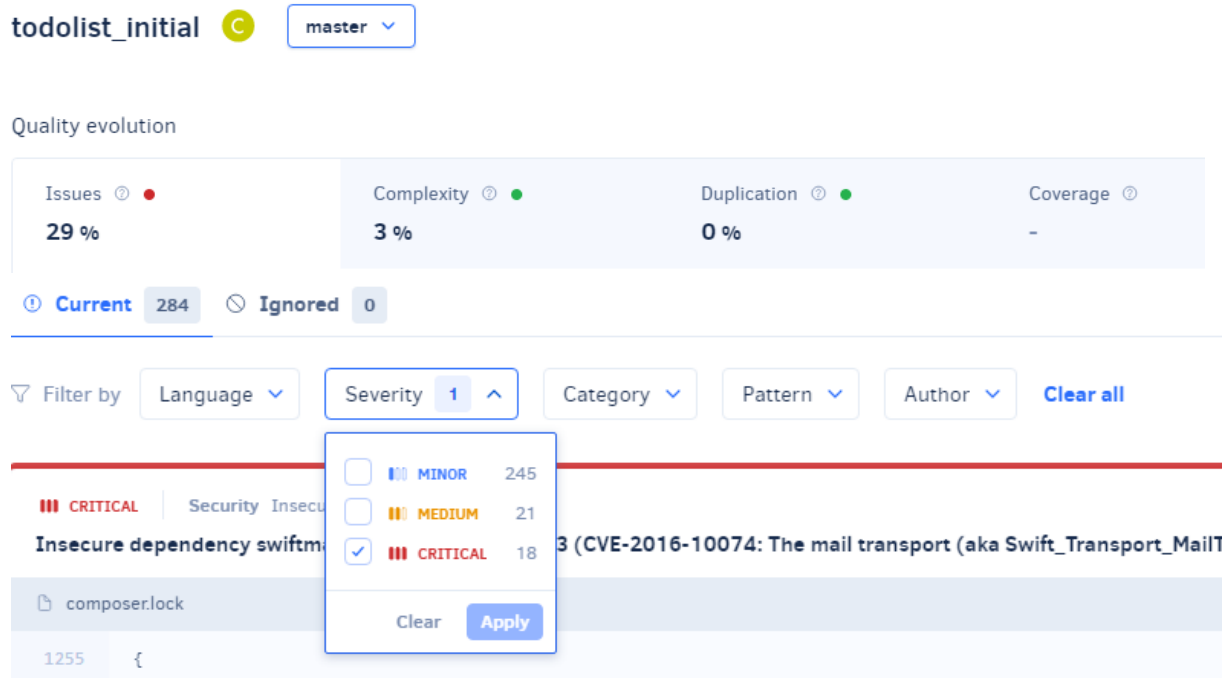
Documentation



MIT License

Source : <https://symfony.com/releases>

## 2. Analyse de la qualité de code avec l'outil Codacy



L'audit Codacy montre un niveau de qualité équivalent à la note **C** et un **pourcentage d'issues de 29%** avec **un total de 284 issues** dont **18 critiques**.

## II. Correction d'anomalies et amélioration de l'application

### 1. Migration vers Symfony 6.4

Afin de corriger les anomalies détectées et répondre aux objectifs du projet, il a été nécessaire de faire une migration vers la dernière **version 6.4 Long-Term Support Release**.

Vu que cette nouvelle version de Symfony nécessite PHP 8.1 ou une version supérieure. Il était plus simple de créer une nouvelle application avec la commande :

```
symfony new --webapp Todolist
```

et y importer les dossiers **src** et **web** qui devient **public** dans la

nouvelle version.

## 2. Correction des bugs

Une fois cet import réalisé, il fallait apporter les modifications nécessaires aux fichiers entity, controller, form et leur namespace.

- Les Controllers « extends » désormais l'AbstractController au lieu de la classe Controller;
- Mise à jour des annotations;
- Ajout des **use** nécessaires au fonctionnement de chaque classe

**Exemple du DefaultController :**

**AVANT :**

```
namespace AppBundle\Controller;

use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;

class DefaultController extends Controller
{
    /**
     * @Route("/", name="homepage")
     */
    public function indexAction()
    {
        return $this->render('default/index.html.twig');
    }
}
```

**APRÈS :**

```
namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

class DefaultController extends AbstractController
{
    #[Route('/', name: 'homepage')]
    public function indexAction(): Response
    {
```

```

    return $this->render('default/index.html.twig');
}
}

```

### 3. Sécuriser les méthodes d'enregistrement et d'authentification

- configurer le fichier security.yaml;
- créer la méthode register() dans le registrationController et supprimer la méthode user\_create;
- corriger la méthode login & créer la méthode logout dans le securityController
- ajouter les types de propriétés à la User entity
- ajouter l'input caché \_csrf\_token au login template
- remplacer le path url dans base.html.twig

### 4. Attacher une tâche à un utilisateur

- Ajouter les types de propriétés à l'entité Task;
- Ajouter la relation ManyToOne à la table Task pour attacher chaque tâche à un utilisateur :

```

- #[ORM\OneToMany(targetEntity: Task::class, mappedBy: 'user')]
- private Collection $tasks;

```

- Ajouter la ligne :

```
$task->setUser($this->getUser());
```

à la méthode task\_create

### 5. Choisir un rôle pour un utilisateur

- Modifier le **RegistrationFormType** et le **UserFormType** en ajoutant **roles checkbox** :

```

->add('roles', CheckboxType::class, [
    'label' => 'Admin',
    'required' => false
])

```

### 6. Améliorations: Mise en place des autorisations

- Ajouter l'annotation #[IsGranted('ROLE\_ADMIN')] à toutes les routes de gestion des utilisateurs afin de restreindre l'accès à ces dernières (voir exemple ci-dessous):

```
#[IsGranted('ROLE_ADMIN')]
```

```
#[Route('/register', name: 'app_register')]
```

- Correction de la méthode deleteTaskAction() dans le TaskController en ajoutant une condition if, afin de permettre la suppression des tasks uniquement par les utilisateurs ayant créé ses tasks et ne permettre la suppression des tâches anonymes que par les utilisateurs ayant le ('ROLE\_ADMIN')

```
#[Route('/tasks/{id}/delete', name: 'task_delete')]
public function deleteTaskAction(Task $task, EntityManagerInterface $em)
{
    .....

    if (($author == $connectedUser) || (in_array('ROLE_ADMIN', $roles) &&
$username == 'anonyme')) {

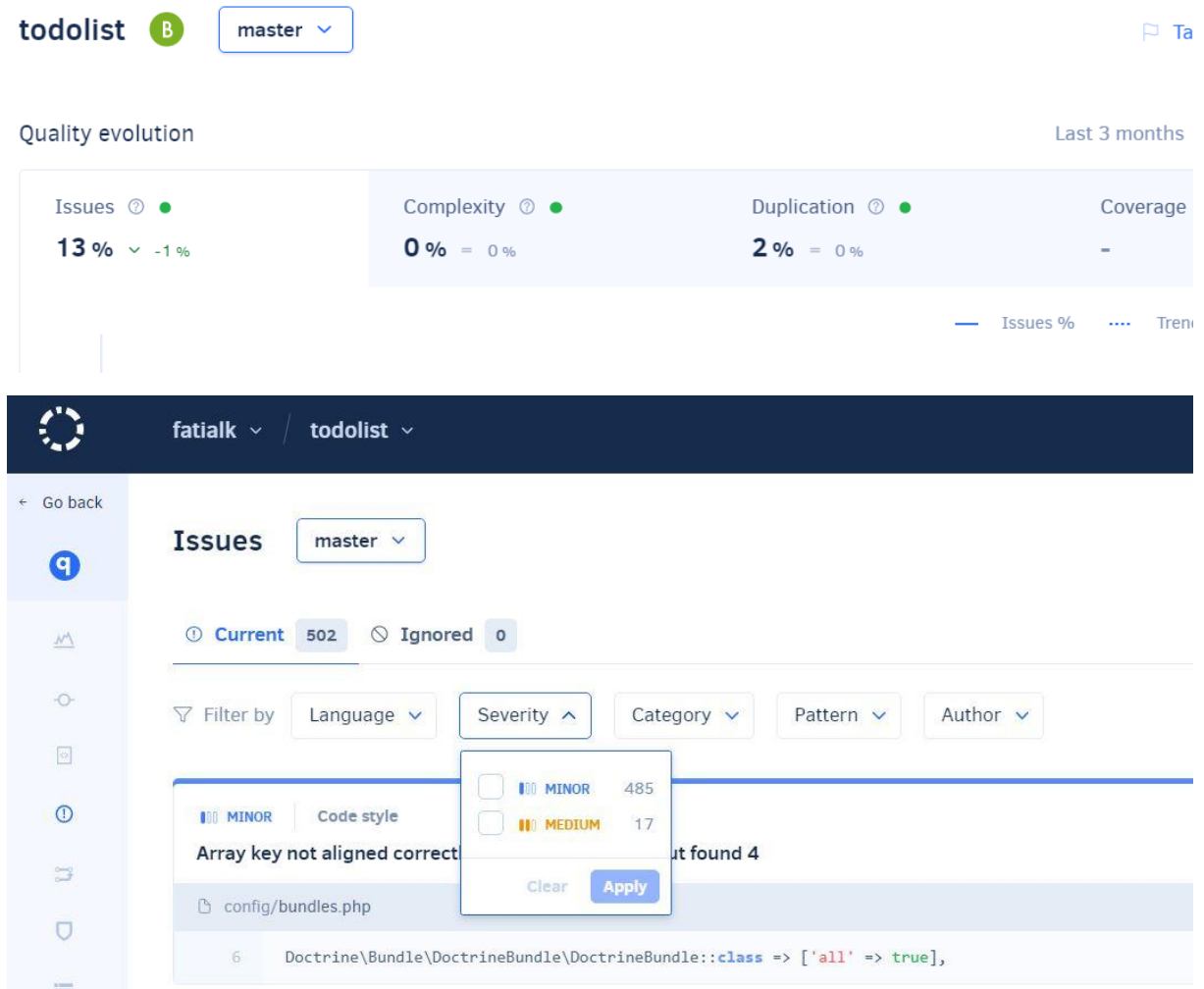
        $em->remove($task);
        $em->flush();

        $this->addFlash('success', 'La tâche a bien été supprimée.');
```

```
    } else {
        $this->addFlash('error', "Vous n'avez pas les droits pour
supprimer cette tâche");
    }
    return $this->redirectToRoute('task_list');
}
```

### III. Résultats de l'audit final

#### 1. audit qualité du code Codacy

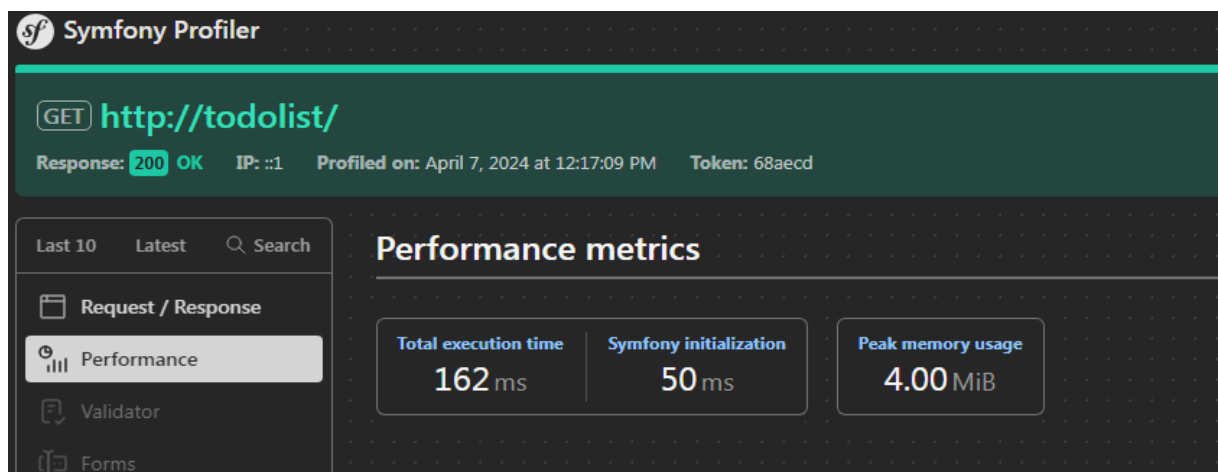


A l'issue des corrections et améliorations apportées, l'audit Codacy final montre un niveau de qualité équivalent à la note **B** et un **pourcentage d'issues de 13%** avec un **total de 502 issues** dont **0 critiques**.

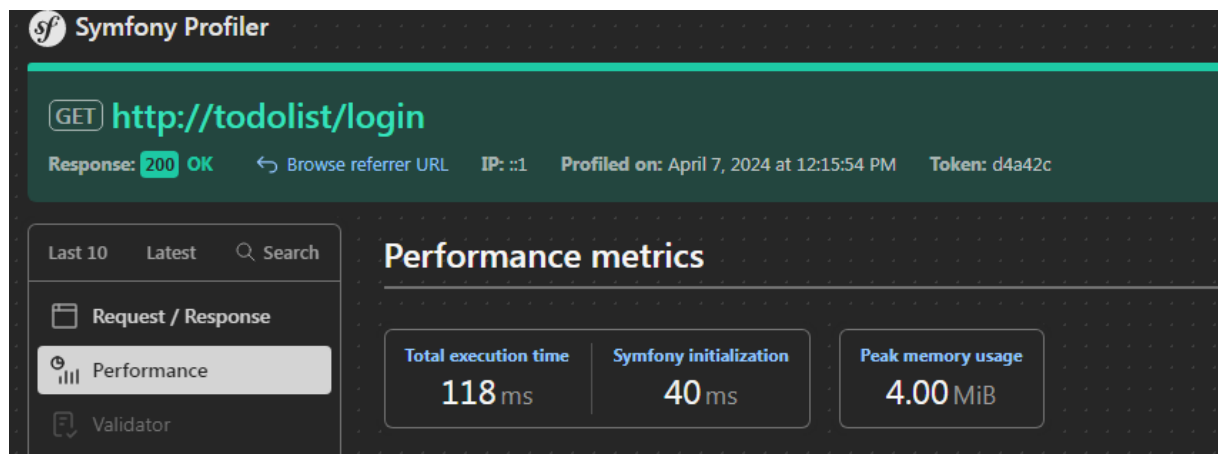


## IV. audit de performance Symfony profiler

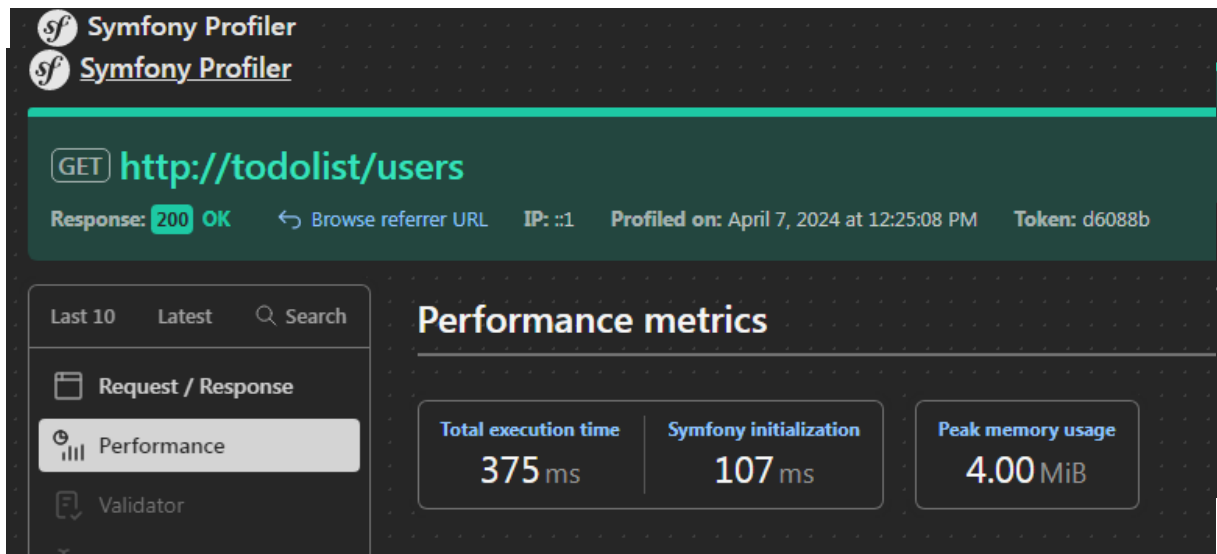
1. Page d'accueil : <http://todolist/>



2. page de connexion: <http://todolist/login>



3. page de création d'un utilisateur: <http://todolist/register>

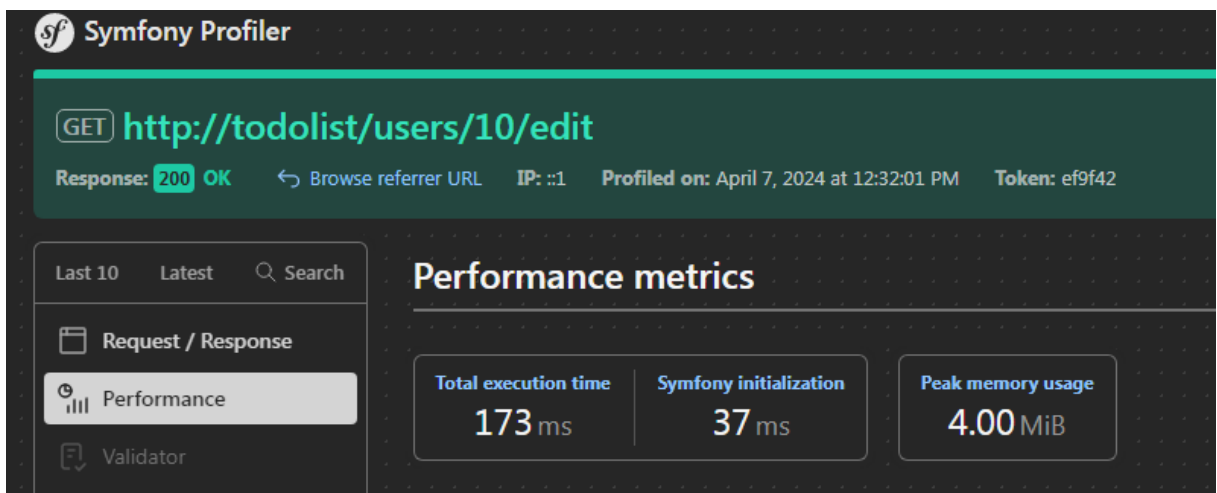


The screenshot shows the Symfony Profiler interface for the endpoint `http://todolist/users`. The status is `200 OK`. The left sidebar contains links for 'Last 10', 'Latest', 'Search', 'Request / Response', 'Performance' (selected), and 'Validator'. The main section displays 'Performance metrics' with three boxes: 'Total execution time' (375 ms), 'Symfony initialization' (107 ms), and 'Peak memory usage' (4.00 MiB).

Metric	Value
Total execution time	375 ms
Symfony initialization	107 ms
Peak memory usage	4.00 MiB

4. page qui retourne la liste de tous les utilisateurs : <http://todolist/users>

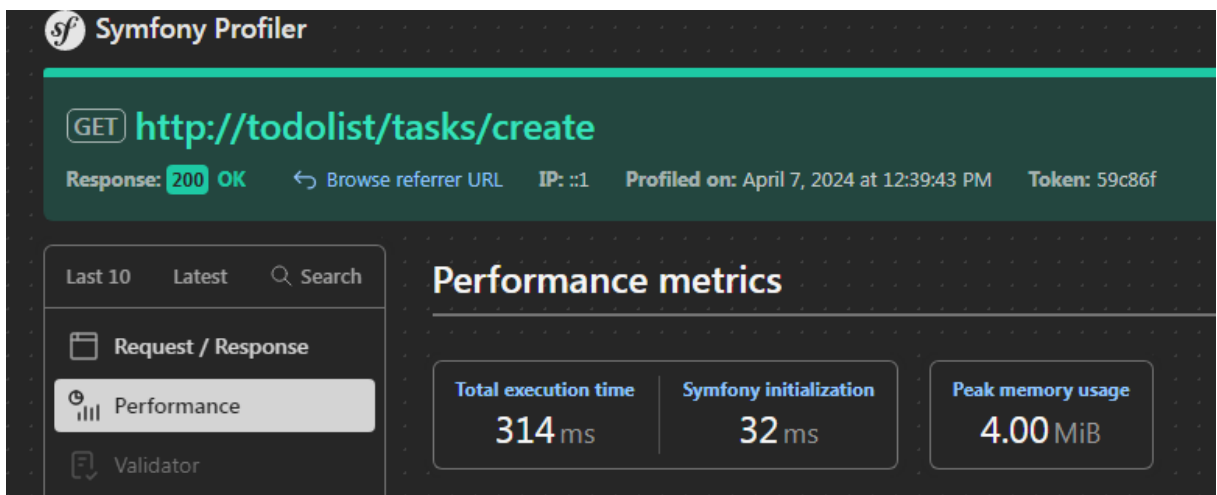
5. page de modification d'un utilisateur : <http://todolist/users>



The screenshot shows the Symfony Profiler interface for the endpoint `http://todolist/users/10/edit`. The status is `200 OK`. The left sidebar contains links for 'Last 10', 'Latest', 'Search', 'Request / Response', 'Performance' (selected), and 'Validator'. The main section displays 'Performance metrics' with three boxes: 'Total execution time' (173 ms), 'Symfony initialization' (37 ms), and 'Peak memory usage' (4.00 MiB).

Metric	Value
Total execution time	173 ms
Symfony initialization	37 ms
Peak memory usage	4.00 MiB

6. page de création d'une tâche : <http://todolist/tasks/create>



The screenshot shows the Symfony Profiler interface for the endpoint `http://todolist/tasks/create`. The status is `200 OK`. The left sidebar contains links for 'Last 10', 'Latest', 'Search', 'Request / Response', 'Performance' (selected), and 'Validator'. The main section displays 'Performance metrics' with three boxes: 'Total execution time' (314 ms), 'Symfony initialization' (32 ms), and 'Peak memory usage' (4.00 MiB).

Metric	Value
Total execution time	314 ms
Symfony initialization	32 ms
Peak memory usage	4.00 MiB

7. page qui retourne la liste des tâches : <http://todolist/tasks/create>

