

Mise en place d'un système de tâches asynchrones pour le site Internet de ZESTE DE SAVOIR

Philippe SWARTVAGHER
`philippe.swartvagher@enseirb-matmeca.fr`

Contexte

[ZESTE DE SAVOIR](#) est un site web communautaire dédié au partage de la connaissance pour tous et sans pépins, créé en 2014. Plusieurs centaines de publications rédigées par des membres du site sont disponibles. Avant leur publication, une partie de celles-ci (articles et tutoriels) passent par une étape de validation sur le fond et la forme, étape durant laquelle un membre de l'équipe de validation bénévole accompagne les auteurs dans l'amélioration de leur publication. L'autre partie des publications, ce sont les billets, un moyen d'expression libre car ils ne passent pas par cette étape de validation avant d'être disponibles publiquement. Toutes les publications sont rédigées au format Markdown, puis [ZMARKDOWN](#) est utilisé pour la conversion vers les formats HTML, PDF et ePUB. Le site web est développé par équipe de bénévoles, principalement en Python avec [Django](#).

Plusieurs mécanismes du site sont exécutés en asynchrone, ou pourraient l'être :

- la conversion d'une publication au format Markdown vers un autre format (HTML, PDF ou ePUB). Cette tâche peut être relativement longue ($\mathcal{O}(\text{minute})$), peut échouer, et l'auteur d'une publication peut avoir besoin de la relancer ;
- l'indexation par le moteur de recherche interne de nouveau contenu (publication, commentaire, message sur le forum, ...);
- la prédiction qu'un commentaire ou un message sur le forum soit du spam ;
- l'envoi des statistiques de consultation du site à Matomo ;
- l'envoi des erreurs Python au système de surveillance Sentry ;
- certaines tâches de maintenance lancées quotidiennement.

Les techniques actuellement utilisées pour les exécutions asynchrones sont diverses (cron, thread dédié, processus qui poll en attente de travail, ...) et ne permettent pas forcément une intégration fine avec le site web, notamment pour assurer le suivi de l'exécution des tâches.

Objectifs

Il existe plusieurs systèmes d'exécution de tâches, on peut notamment citer [Celery](#) et [Huey](#). Faire exécuter les actions citées ci-dessus par un système de tâches permettrait une meilleure manipulation des tâches et de soulager le serveur web en enlevant les tâches du chemin du traitement des requêtes HTTP.

L'objectif de ce projet est de modifier l'architecture du site Internet de ZESTE DE SAVOIR pour utiliser un tel système de tâches.

Tâches techniques

- Choisir un système de tâches ;

- Modifier le code du site de ZESTE DE SAVOIR pour transmettre les tâches à exécuter en asynchrone à un système de tâches ;
- Migrer plusieurs mécanismes pertinents du site vers ce système de tâches ;
- Compléter [les scripts de déploiements](#) pour déployer également le système de tâches.

Ressources

- le dépôt Git du code source du site de ZESTE DE SAVOIR :
<https://github.com/zestedesavoir/zds-site/>
- les notes de version de Django 6 qui introduisent un support minimal pour tâches :
<https://docs.djangoproject.com/en/6.0/releases/6.0/#background-tasks>
- un article de blog qui explore comment Django 6 peut utiliser un système de tâches :
<https://roam.be/notes/2025/a-first-look-at-djangos-new-background-tasks/>