

Développement de nouveaux modules sur le projet Sith

Florent Jacquet
Guillaume Renaud
TO52 - A16

Sommaire

Les nouvelles applications

Eboutic

Le SAS

Les élections

La laverie

La communication

La gestion des stocks

Fonctionnement

Améliorations et difficultés

Le rôle de mainteneur

Réviser les merge requests

Gestion des bugs, des tickets, de la mise en production...

Conclusion

Les nouvelles applications

- Fournir une boutique
- Paiement en ligne en lien avec l'API du Credit Agricole
- Gestion des cotisations et rechargements
- Attention aux accès concurrentiels : pas visibles pendant le développement, car mono-thread, mais problèmes à la mise en production

- Galerie de photos
- Upload simple pour tout le monde, même pour plusieurs dizaines de photos
- Modération et gestion des droits basée sur la gestion des fichiers, ce qui a permis d'améliorer ces derniers
- Problèmes d'optimisation de certaines pages qui mettaient plus de 9 secondes à générer (plus que 2s maintenant)

- Grosse partie "gestion" : c'est Sli qui a principalement développé l'application
- Revue des merges request et choix de design
- Problématique de législation vite ignorées puisque validation officielle en AG

- Gestion d'un planning de reservation en prenant bien en compte les différents états (hors-service, ...) de chaque machine
- Génération de formulaires dynamiques en fonction des réservations (factory design pattern)

- Dynamise le site avec tous les textes paramétrables
- Fourni un système de news
- Fourni une newsletter

- Envoie de mails en masse
- Beaucoup de templates

La gestion des stocks

- Création automatique des listes de courses
- Approvisionnement des stocks
- Prise d'éléments dans le stock

Améliorations et difficultés

- Mise à jour quantité liste de courses
- Mise à jour automatique du stock selon les ventes
- Ajout au système de notifications

Difficultés

- Découverte du design pattern "factory" pour les formulaires dynamiques
- Apprentissage de Python, en plus du framework

Le rôle de mainteneur

Réviser les merge requests

- Long et fastidieux
- Nécessaire pour maintenir une base de code cohérente
- Permet de retrouver les bugs des nouveaux contributeurs
- Oriente les contributeurs sur la bonne voie et la marche à suivre avec Django/Jinja2/etc...

- Ouverture/fermeture des tickets
- Mailing list/IRC
- Mise en production, gestion des migrations
- Restauration de la base de tests régulièrement

Organisation de la passation

Conclusion

- Apprentissage Django/Git
- Nouvelle mise en pratique des concepts de base de données relationnelles
- Utilisation poussée de Gitlab
- Formation de nouveaux contributeurs

Merci de votre attention

Questions ?

Remarques ?