

Site AE

Réalisé dans le cadre de l'unité de valeur
TO52



Écrit par **Bartuccio Antoine**
Automne 2019

Gestionnaire de l'UV : Yassine RUICHEK

Professeur encadrant : Franck GECHTER



Sommaire

1	Présentation du projet	3
2	Technologies utilisées	4
3	Suivi d'incident	5
4	Travail réalisé	6
4.1	Travail de maintenir	6
4.2	Migration Django 2.2	6
4.3	Journal des opérations	7
4.4	Documentation	7
4.5	Amélioration des performances	8
4.5.1	Amélioration du SAS	8
4.5.2	Les ventes de clubs	9
4.5.3	En vrac	10
4.6	Résolution de bugs	10
4.6.1	Faible des permissions	10
4.6.2	En vrac	11
5	Améliorations pour le futur	12

1 Présentation du projet

Le but de ce projet est de fournir à l'Association des Étudiants de l'UTBM une plate-forme pratique et centralisée de ses services. Le Site de l'AE tient à jour le registre des cotisations à l'association, prend en charge la trésorerie, les ventes de produits et services, la diffusion d'événements, la gestion de la laverie et bien plus encore. Il est conçu de manière suffisamment générique pour être facilement adaptable à une autre association.

C'est un projet bénévole qui tire ses origines des années 2000. Il s'agit de la troisième version du site de l'association initiée en 2015. C'est une réécriture complète en rupture totale des deux versions qui l'ont précédée.

Au sein de ce projet, j'endosse à la fois le rôle de maintenir et de lead developer. C'est moi qui suis en charge de valider le travail des autres développeurs et de garantir l'intégrité du projet. En tant que lead developer, je suis à la fois celui qui contribue le plus sur le projet et celui qui décide de l'évolution du projet dans le futur.

2 Technologies utilisées

Pour comprendre un projet, il est toujours intéressant de regarder des technologies sur lesquelles celui-ci repose.

Le projet est écrit en Python et repose sur le framework Django. C'est un framework web pour Python apparu en 2005. Il fournit un grand nombre de fonctionnalités pour développer un site rapidement et simplement. Cela inclut entre autre un serveur Web de développement, un parseur d'URLs pour le routage des différentes URI du site, un ORM (Object-Relational Mapper) pour la gestion de la base de données ainsi qu'un moteur de templates pour le rendu HTML. Django propose une version LTS (Long Term Support) qui reste stable et est maintenu sur des cycles plus longs, ce sont ces versions qui sont utilisées.

Notre approche du problème est orientée backend first et ceci pour plusieurs raisons :

- Simplicité de maintenance
- Réduction de la barrière d'entrée par la limitation des technologies
- Vitesse de téléchargement et d'affichage côté utilisateur

Pour la liste complète de tout ce qui est utilisé, pour plus de détails sur comment tout ceci a été choisi ainsi que la philosophie du projet en lui-même, une section complète sur ce sujet est disponible sur la documentation officielle du projet <https://sith-ae.readthedocs.io/fr/latest/about/tech.html>.

3 Suivi d'incident

Dans l'optique d'améliorer le logiciel, il est nécessaire d'identifier rapidement et efficacement tous les incidents dus à une erreur logicielle lors de l'exécution. Une méthode souvent employée est d'utiliser un *bug tracker* (tel GLPI) où les utilisateurs font un rapport en cas de dysfonctionnement du logiciel afin d'en avertir les responsables. C'est une méthode qui selon moi manque d'efficacité puisque les utilisateurs ne s'en servent que très rarement. Les erreurs remontent donc très lentement voir jamais et sont difficiles à localiser puisque entièrement dépendantes de la description de l'utilisateur. La solution employée ici est une approche basée sur la remontée automatique d'incident en cas d'erreur via l'utilisation du logiciel Sentry.

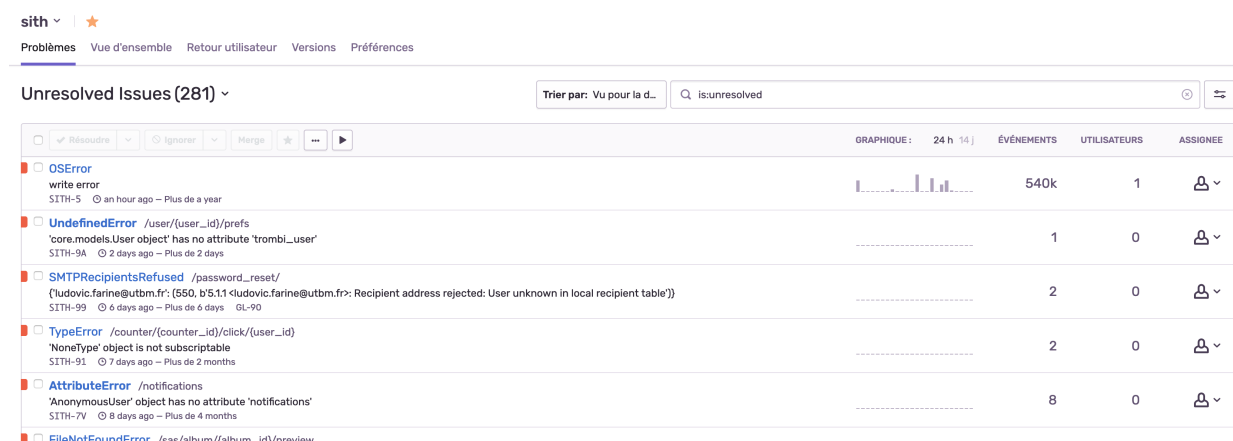


FIGURE 1 – Erreurs non résolues

Sentry est une plate-forme logicielle en ligne que j'ai eu l'occasion de découvrir en ST40 et qui permet de référencer toutes les erreurs d'exécution d'un logiciel. Il suffit d'inclure la librairie fournie par Sentry dans le projet, de la configurer correctement et toutes les erreurs sont automatiquement remontée au serveur. J'ai mis en place cette solution durant mon stage l'année dernière et elle s'est révélée cruciale depuis lors.

Au travers de l'interface en ligne, il est possible d'identifier la fréquence d'apparition d'une erreur et ainsi hiérarchiser la priorité de résolution. Il est également possible d'obtenir des informations sur l'utilisateur ayant rencontré le problème (navigateur utilisé, url appelée, date et heure de la visite) mais également et surtout la ligne précise où le programme a planté ainsi que toute la pile d'exécution qui précède l'incident ! L'utilisateur peut également laisser un petit commentaire au travers d'une boîte de dialogue directement sur la page d'erreur du site, simplifiant grandement la procédure pour lui.

4 Travail réalisé

Lors de ce semestre, j'étais chargé de mission au sein du bureau AE afin d'épauler l'équipe informatique. À ce titre, j'étais le référent principale sur le projet et participais activement à de nombreuses tâches variées.

La majorité des objectifs fixés étaient à très court terme et étaient surtout dictés par les problèmes utilisateurs et les demandes du bureau. Voici un petit résumé du travail qui a été effectué.

4.1 Travail de maintenir

Comme expliqué précédemment, j'endosse sur ce projet le rôle de maintenir. Cela signifie qu'une partie importante de mon travail est de discuter avec les contributeurs, de relire leur code, de les aider à le corriger et l'améliorer et de garantir que les déploiements en production se passent correctement.

4.2 Migration Django 2.2

Comme expliqué précédemment, le projet se base sur le framework Django pour fonctionner. Nous utilisons exclusivement les versions supportées à long terme (LTS). L'objectif principale de ce semestre et la priorité numéro une du projet était de migrer de la version 1.11 qui terminait sa vie début 2020 pour passer à la version 2.2 qui sera supportée jusqu'en 2022.

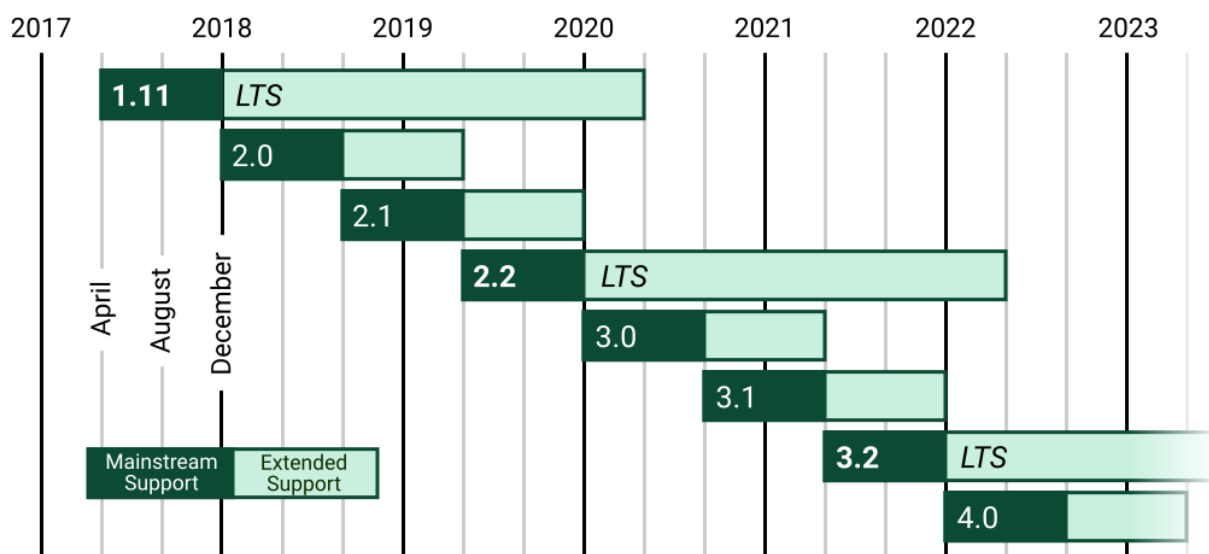


FIGURE 2 – Sorties et support des versions de Django

Le passage à une nouvelle version majeure de Django n'est pas sans risque. En effet, il est très facile d'introduire de nombreux bugs et comportements différents suite à des changements d'API

ou encore de suppression de fonctions dépréciées. Le Python étant un langage interprété, il est très difficile de détecter les erreurs provoquées par de potentielles ruptures d'API.

La migration s'est déroulée de la manière suivante :

- Écriture de tests unitaires et d'intégration
- Lecture des changements majeurs introduits dans la nouvelle version
- Mise à jour
- Recherche et correction des erreurs jusqu'à ce que le projet démarre enfin
- Utilisation d'outils d'analyse statiques offerts par le framework pour trouver des erreurs
- Détection d'erreurs au travers des tests unitaires et d'intégration
- Test sur la pré-production
- Test en production lors d'une période calme d'utilisation

4.3 Journal des opérations

Au cours du semestre, le BdF m'a rapporté certains abus de droits sur les suppressions de transactions. Pour éviter les dérives, il a été convenu de mettre en place un système de journal d'opérations pour référencer toute suppression de transaction ou rechargement.

Date	Type d'opération	Étiquette	Opérateur
2020-01-16 15:21:05.356663+00:00	Suppression de vente	Selling: 1 x Cheeseburger (1.000000) for	
2020-01-16 15:20:55.185031+00:00	Suppression de vente	Selling: 1 x Cheeseburger Frites (1.900000) for	
2020-01-15 16:49:28.403689+00:00	Suppression de vente	Selling: 1 x Formule Buns (1.500000) for	
2020-01-14 13:01:35.979668+00:00	Suppression de vente	Selling: 1 x Jus de fruits (0.700000) for	
2020-01-13 11:47:11.830999+00:00	Suppression de vente	Selling: 1 x Sandwich Plus MDE (2.850000)	
2020-01-11 10:31:10.931803+00:00	Suppression de vente	Selling: 1 x Café (0.350000) for	
2020-01-10 23:24:05.635626+00:00	Suppression de vente	Selling: 2 x Chimay (Blanche/rouge/bleue) (3.900000) for	
2020-01-10 22:37:37.694209+00:00	Suppression de vente	Selling: 1 x SODEBO - Triangle (2.700000) for	
2020-01-10 19:55:57.581437+00:00	Suppression de rechargement	Refilling: 10.00 for	
2020-01-10 10:17:16.450052+00:00	Suppression de vente	Selling: 1 x Soft (0.700000) for	

FIGURE 3 – Journal des opérations

J'ai opté pour une approche simple et générique au travers de signaux de suppression. Le principe est simple, chaque opération de base de donnée via l'ORM du framework peut être interceptée et il est possible d'y ajouter un comportement personnalisé. En se mettant sur le signal de pré suppression, j'enregistre la date et l'heure de celle-ci, le type d'opération effectué (ici suppression de vente ou de rechargement), l'utilisateur responsable de cette suppression ainsi qu'un label permettant de comprendre ce qui s'est passé.

J'espère ne pas avoir à étendre la liste des opérations suivis avec ce système puisque suivre les actions des utilisateurs n'est pas dans la philosophie du projet.

4.4 Documentation

Un point trop souvent négligé sur ce projet est l'aspect documentation. Celle-ci se limite généralement à commenter certains passage du code mais ce n'est clairement pas suffisant. La documentation d'installation était répartie sur deux fichiers, on y retrouvait beaucoup de répétitions, aucune explication ni sur la philosophie du projet ni sur sa prise en main.

Pour pérenniser le projet, il était nécessaire de passer par l'écriture d'une documentation simple, compréhensible et complète. Les duplications d'explications ont été supprimées, le style d'écriture uniformisé et presque tout a été traduit en français pour la compréhension des nouveaux utilisateurs. Cette nouvelle documentation contiens :

- Une introduction au projet et sa philosophie
- Un guide d'installation du projet et de l'environnement de développement
- Une présentation de la structure du projet
- Un guide de prise en main rapide du projet
- Des tutoriaux sur des tâches récurrentes
- Une présentation de notre sur-couche qui se greffe au dessus du framework
- Des informations diverses pour aider à prendre en main certains outils
- Des liens vers des tutoriaux à propos des technologies utilisées
- De quoi télécharger les anciens rapports de TO sur le projet

Le tout est hébergé sur le site d'un prestataire externe (readthedocs) et est automatiquement mis à jour lors de la fusion des modifications dans le master du projet.

4.5 Amélioration des performances

4.5.1 Amélioration du SAS

Au début du semestre, nous avons détecté d'énormes soucis de performances en provenance du SAS (la galerie de photos). Le calcul des permissions était extrêmement lent et la génération de la page était très lente. Il faut savoir que lorsqu'un navigateur récupère des images, il télécharge chacune d'entre elles indépendamment et de manière asynchrone. Cela permet d'améliorer généralement la vitesse d'affichage d'une page web. Cependant, dans notre cas précis, chaque image correspondait à une nouvelle requête et à un nouveau calcul de permission.

La première approche a été d'optimiser un maximum les requêtes effectuées pour la recherche. Cela m'a amené à ajouter un index sur la table des fichiers du site. En effet, une photo sur le SAS est enregistré dans la même table qu'un fichier et est différencié uniquement par sa colonne photo qui est à vrai. Ajouter un index sur cette colonne a considérablement amélioré les performances.

Appliquer un index ne suffisait pas, les performances étaient encore désastreuses. J'ai donc ajouté du cache sur les vérifications de permissions de fichier. Puisque toutes les images du SAS ont exactement la même visibilité, il suffit de vérifier les permissions d'une seule image pour garantir que l'utilisateur a le droit de voir la suite. La technique ici est donc de garder en cache le résultat des vérifications pour un utilisateur sur un temps court (4s) mais spécifiquement sur le SAS. Ce cache est partagé entre les requêtes et même les différents runners qui font tourner le site et cette technique a permis de gagner encore en performances.

Enfin, pour accélérer encore plus le SAS, j'ai rajouté une pagination sur les albums. En effet, il se trouve que le plus long dans la génération d'une page est la création du fichier HTML en lui

même à partir du template. Limiter le nombre d'éléments à y inscrire est la manière la plus efficace d'améliorer les performances.

4.5.2 Les ventes de clubs

Puisque toutes les ventes de produits sont effectués sur le site, il est nécessaire d'avoir une interface pour vérifier qu'un utilisateur a bel et bien acheté son produit. C'est le rôle de l'interface de vente des clubs. Elle est spécifique à chaque club et comprend un formulaire où il est possible de donner des contraintes sur la recherche (période des ventes, produit à suivre, comptoir sur lequel on souhaite voir les ventes). La page affiche ensuite les utilisateurs ayant acheté le produit, à quelle heure, en quelle quantité et sur quel comptoir. Il est également possible de télécharger cette liste au format CSV. Problème : l'interface ne répond pas sur un grand nombre de vente et c'est particulièrement problématique pour notre foyer étudiant où il est impossible de visualiser plus d'une semaine de ventes. De plus, le formulaire n'offre aucune distinction entre produit archivé et non archivé. Il n'est en plus possible de sélectionner qu'un seul comptoir et/ou produit à la fois, ce qui rend les recherches pénibles.

Dans un premier temps, j'ai retravaillé ce formulaire pour le rendre plus pratique à utiliser. La distinction entre produit archivé et non archivé a été faite. Il est désormais possible de sélectionner plusieurs produits ainsi que plusieurs comptoirs à la fois.

Dans un second temps, je me suis penché sur les problèmes de performances. Ayant déjà revu et optimisé les requêtes effectuées pendant la refonte du formulaire, j'ai gagné du temps sur la génération de la page en ajoutant une pagination. Tout comme pour le SAS, c'est encore une fois la génération du fichier HTML qui est la plus consommatrice de temps. Cette simple modification permet désormais d'afficher la quasi intégralité des ventes du foyer contre moins d'une semaine au départ.

Enfin, il a fallu intervenir sur la génération du CSV. Même si désormais il est possible d'afficher bien plus de ventes, le problème de génération de fichier est également présent sur le CSV et trop d'entrées à écrire résulte purement et simplement en un time out du serveur. Ici, impossible d'utiliser la technique de la pagination. Je me suis donc tourné vers le streaming de fichier. Pour

FIGURE 4 – Le nouveau formulaire de ventes

parvenir à rendre l'intégralité des ventes, l'utilisateur télécharge le fichier pendant sa génération par le serveur. Les téléchargements sont lents, il est impossible pour le navigateur d'estimer le temps de téléchargement mais désormais il est possible de récupérer les ventes. Le compromis vaut donc le coup.

4.5.3 En vrac

- Mise en cache du status des comptoirs et des anniversaires
- Amélioration de l'ergonomie et de la vitesse d'affichage des comptoirs de vente
- Amélioration des performances du forum
- Amélioration des performances des notifications

4.6 Résolution de bugs

4.6.1 Faille des permissions

Cette faille a été publiquement annoncée et expliquée. Les explications sont disponibles sur le forum du site à cette adresse : <https://ae.utbm.fr/forum/topic/17943?page=1msg2277272>.

Suite à l'ajout d'un lien direct vers les billets électroniques vendus au Gala dans le mail de confirmation d'achat, nous avons observé une charge très élevée sur nos serveurs. L'infrastructure était saturée et il était nécessaire de relancer le site très souvent pour ne pas remplir toute la mémoire vive du serveur.

Il se trouve que cela venait d'une erreur dans notre système de gestion des permissions. Pour faire simple, il y a deux types de vérifications de permissions sur le site : la vérification sur un élément unique et la vérification sur une liste d'éléments. Pour distinguer les deux cas d'utilisation, nous avons opté pour un try/catch qui permettait de tester à quel type de test nous avions à faire. C'était une grave erreur à l'origine de nombreux dysfonctionnements au travers du site.

Lors de la génération d'une page, si une erreur non gérée a lieu, le système considère dans tous les cas que c'est un cas de vérification multiple. Il peut alors se retrouver à tester l'intégralité des éléments du même type que celui testé présents dans la base de données. Dans le cas de notre billet électronique, lorsqu'un utilisateur n'avait pas les droits d'accès après avoir cliqué (si par exemple il n'est pas encore connecté), la génération du billet plante et l'intégralité des ventes du site sont alors testées une à une. C'est indétectable en développement local puisque les jeux de données sont trop petits mais sur le site réel, cela représente plus de 910 000 éléments !

Pire encore, si après la vérification individuelle des données il existe au moins un élément visible par l'utilisateur, le système de permission considère que la page peut être vue ! Il suffit alors que la deuxième génération de page ne plante pas cette fois-ci pour qu'elle soit accessible à l'utilisateur. C'est le cas des billets électroniques ici.

J'ai donc réécrit entièrement cette portion cruciale du code pour le rendre plus sécurisé. En résulte un site vraiment plus stable et plus rapide.

4.6.2 En vrac

- Corrections d'erreurs et amélioration de la finition de la section trombinoscope
- Correction d'erreurs sur l'administration des écrans de communication
- Correction d'erreurs sur l'édition des profils utilisateurs
- Meilleure gestion des erreurs avec le serveur mail de l'UTBM
- Amélioration du système de mot de passe oublié lors de l'expiration des liens
- Correction d'erreurs lors de l'abonnement à une section du forum
- Correction d'erreurs sur l'identification des utilisateurs sur les photos

5 Améliorations pour le futur

Il reste toujours énormément de travail à effectuer sur ce projet, même après plus de 3 ans de travail dessus. C'est un projet à long terme avec des thématiques récurrentes d'améliorations. Ces améliorations se présentent sous quatre grands axes :

L'interface L'interface utilisateur, et plus généralement le design du site en lui-même, est un problème récurrent qui ressort à chaque semestre. Le site a visuellement l'air vieux et il n'est pas toujours ergonomique, surtout sur mobile. Durant ce semestre, Mr Émile Kirnberger a eu l'occasion de travailler sur ce sujet au cours de sa PA00. Il a réalisé des maquettes pour mobile du site et incorporer ces idées dans le site sera le sujet d'un travail ultérieur.

La documentation On n'a jamais terminé un projet, on ne fait qu'arrêter de travailler dessus. C'est le cas également pour la documentation de celui-ci qui doit constamment être mise à jour. Il manque encore beaucoup d'informations sur de nombreux modules du projet dans la nouvelle documentation. Il manque également de nombreux tutoriels pour les développeurs.

La stabilité Même si elle a considérablement été améliorée ce semestre, ce n'est toujours pas suffisant et il reste toujours du travail à faire. Outre la résolution de bugs et l'amélioration des performances, il manque beaucoup de tests unitaires et d'intégration pour s'assurer d'éviter toute régression possible.

Les fonctionnalités Le nouveau site n'incorpore toujours pas l'intégralité des fonctionnalités de l'ancien, même 3 ans plus tard. Il manque entre autres les outils pour les responsables sites où les utilisateurs peuvent réserver les salles et le matériel. Certains utilisateurs réclament également le retour du Galaxy, sorte d'outil qui permet de visualiser au moyen d'un graph la proximité entre deux utilisateurs avec seulement les informations du site (clubs en commun, photos en commun...). C'est quelque chose qui n'est pas cruciale et ce sera très certainement la dernière fonctionnalité à implémenter mais il faut la garder dans un coin de l'esprit.