

# ESUP-DAYS (34)

## Un nouvel ENT pour AMU

---

Dominique LALOT / Grégory TRUCY  
Pôle Environnement Numérique-Direction du Numérique  
Aix Marseille Université

## HISTORIQUE

- Ancien ENT basé sur uPortal
  - Développé par le consortium JASIG devenu APEREO
  - uPortal proposé en France vers 2002 suite appel d'offre du MEN dans le cadre du projet [ESUP](#)
- On a connu différentes versions depuis 2005
  - 2.3
  - 3.2
  - 4.2

## HISTORIQUE

- Mises à jour: entre 3 et 6 mois
- Complexe: beaucoup de fichiers XML à éditer
- Long à démarrer (parfois 2 minutes)
- Beaucoup de lignes de codes et de librairies ajoutées
- 4 serveurs avec pas mal de RAM
- Des fonctionnalités non utilisées
- Une interface admin buguée (v4.2): fichiers xml et commandes ant (2 minutes)

## POURQUOI DU NOUVEAU

On a juste besoin finalement d'un gestionnaire de bookmark en fonction d'un profil.  
Pourquoi ne pas consacrer le temps d'une migration (quelques mois) à coder une application

- ✓ Nouvelle technologie
- ✓ Plus robuste
- ✓ Plus adaptée à nos besoins
- ✓ Qu'on maîtrise
- ✓ Et qui marche quelque soit le nombre d'utilisateurs!

## POURQUOI DU NOUVEAU ?

Fonctionnalités souhaitées :

- ✓ Gérer des favoris sur les applications
- ✓ Mettre en évidence les applications les plus utilisées
- ✓ Forcer des applications présentées en dehors des menus
- ✓ Faire une recherche dans les menus
- ✓ Avoir des compteurs sur une application (mail, calendrier)
- ✓ Faire un changement d'identité sur un autre utilisateur
- ✓ Relier les applications à SIAMU pour les alertes applicatives
- ✓ Tout administrer via l'interface
- ✓ Afficher un message d'accueil en fonction d'un filtre LDAP (via application externe)

## POURQUOI DU NOUVEAU ?

Des années que ça nous trottait dans la tête.

Notre collègue David nous quitte mais sur son temps libre, il nous code le nouvel ENT sur ces directives:

- ✓ Du Single Page (Le navigateur gère un maximum de choses)
- ✓ Des JSON Web Token (JWT) pour communiquer avec les backend (web services). Transmis par cookie ou depuis session storage
- ✓ Pas de sessions côté serveur (stateless)
- ✓ Ne pas trop changer l'interface existante
- ✓ Développement maquette fini en décembre 2019

En frontal, codé en Angular 6 et JAVA Spring Boot côté backend. Peu de documentation et aucune formation ( COVID ) pour reprendre le projet de David.

## SINGLE PAGE APPLICATION (SPA): Comment ça marche

Première interaction: chargement du code JavaScript dans le navigateur

- Une grosse page statique avec du code compact compressé et peu lisible, chargement des CSS, polices etc..

Les clics suivants sur la page ne provoquent plus de chargement de pages.

- Le code appelle des web services sur le serveur (backend), examine les contenus JSON et gère son affichage. Les accès aux web services sont protégés par des JWT.



## MIGRER QUAND ?

- Trouver le temps
- Prendre un risque aussi ...
- Bonne nouvelle, on migre d'HARPEGE à SIHAM, avec la fin de ESUP-agent, une portlet de moins
- Il faut trouver une alternative à la portlet ESUP-filemanager
  - 3 mois de dur labeur pour trouver une application (Pydio), cassifier Pydio, épurer l'interface utilisateur.
- Développement de scripts de récupération des données (applications / menus / esup-bookmark vers Nextcloud)
- Début septembre, panique, l'ENT ne répond plus
  - problème infra mais on ne le sait pas.
  - dans l'urgence, on travaille sur la nouvelle version pour de la pré-prod, et on poursuit nos tests de performance (Umeter, mais compliqué pour du SPA)

## MIGRATION

- Octobre 2020: uPortal plante régulièrement dès qu'on dépasse 50000 utilisateurs
- Notre version de test du nouvel ENT semble correcte
- Allez: on y va.

# Chiche

## MISE EN PRODUCTION

- Démarrage: pic de charge rapidement sur la BDD (*Problème sur le mapping Objet Relationnel*)
  - augmentation du nombre maximum de connections du serveur SQL
  - Pool SQL dans le serveur Tomcat trop faible
- +24h
  - Les logs LDAP débordent..
  - Analyse du code: corrections
    - On met plus de choses dans le JWT pour minimiser les requêtes LDAP (nom, prénom, mail, admin). Angular peut lire le JWT et a toutes les informations hors menus.
- +72h
  - Les logs du load balancer HAPROXY débordent
    - Oups: beaucoup trop de requêtes (bizarre)

## MISE EN PRODUCTION: Mapping objet défectueux

Lorsqu'on clique sur une application pour une première fois il faut insérer un enregistrement en BDD pour l'utilisateur qui servira à stocker le nombre de clics sur l'application.

On a constaté le fonctionnement suivant:

- Lecture de la table complète (plus de 100000 entrées) pour mise en mémoire
- Vidage de la table
- Remplissage avec la ligne supplémentaire

Le problème => la table est vide au moment de la mise en production, donc beaucoup d'insertions !

- Des milliers d'utilisateurs se connectent et le nombre de requêtes augmente de plus en plus au fil des clics.
- Un mécanisme de « table lock » empêche les requêtes de s'exécuter rapidement jusqu'à remplir le pool de connexion jdbc (800)

## MISE EN PRODUCTION: Les compteurs s'affolent

### Analyse du problème

- Les compteurs de mails ou calendrier s'affolent. Ils devaient être lus toutes les 10 minutes.
- Et on en voit beaucoup plus que prévu dans le navigateur: des dizaines en même temps.
- Au bout de quelques heures c'est quelques centaines de requêtes qu'un navigateur génère!

C'est forcément côté Navigateur et dans le code JavaScript (Angular) que le problème se situe

```
const source = interval(applicationFavorite.timer * 60000);
```

Ceci ajoutait toutes les 10 minutes un event pour aller chercher les compteurs ...

Il a fallu isoler l'instruction pour ne la jouer qu'une seule fois.

## CORRECTION DE BUGS: Mises à jour ?

Comment fait-on côté navigateur ?

Single Page = le serveur ne fait plus rien que transmettre des contenus JSON (JavaScript Object Notation, en gros clef=valeur)

Mince, on fait comment ?

Test: on génère des redirects HTTP depuis le backend (serveur)

Le navigateur ignore car c'est Angular qui gère et si le code ne gère pas, aucune action n'est faite

**Aïe ...**

## CORRECTION DE BUGS

- On peut espérer que les personnes ferment leur navigateur
- Que les mises à jour des machines fassent redémarrer.
- 3 mois après, 2 machines avaient encore notre première version d'octobre
  - un Windows 7 à la faculté de Pharmacie  
(non arrêté pendant 4 mois, mais mis en veille de temps en temps)
  - un Linux dans un labo  
(non arrêté, donc pas de mises à jour de sécurité)

# 1 million de requêtes/jour

## MIEUX GERER LES SESSIONS ET LES MISES A JOUR

Un JWT est une information signée que le backend fournit au navigateur contenant des informations. Cette information est renvoyée par le navigateur à chaque échange avec le backend

- Il a une durée de vie. Au départ 10 jours! Descendu de suite à une journée.
- Côté Backend: si JWT expiré, on renvoie un HTTP DENY
- Côté Angular: si HTTP DENY, on recharge l'application (mise à jour enfin possible)

## ON A DÛ « MAITRISER » CECI

Notre pôle fait de l'intégration mais pas du développement pur et dur à part sur GEFORP.

On travaillé un mois pour résoudre les problèmes et acquérir quelques petites compétences autour de :

- IDE (Eclipse)
- JAVA / Spring
- Hibernate (Mapping Objet Relationnel)
- Angular
- Git

<https://github.com/amu-dosi-polen/ent-frontend>

<https://github.com/amu-dosi-polen/ent-backend>

## CONCLUSION

- Plus aucun plantage
- 1 seul serveur (enfin un autre pour la redondance) au lieu de 4
- Le rush des résultats de la faculté de médecine est passé sans problème. 4000 étudiants en quelques minutes ...
- On peut redémarrer le backend sans que les utilisateurs le remarquent (20 secondes contre deux minutes)
- On peut facilement déléguer l'administration

**L'ENT est le point d'entrée de nos utilisateurs.  
Il doit être le maillon fort et non pas le maillon faible**

# DEMONSTRATION