

# ESUP-DAYS (22)

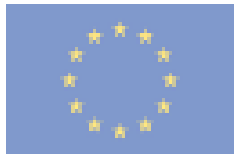
---

## ESUP-PUBLISHER, UNE PUBLICATION SIMPLIFIÉE

Julien Gribonvald -- GIP RECIA

## PLAN

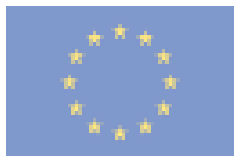
- ♦ **Pourquoi ESUP-Publisher ?**
- ♦ **Choix des technologies**
- ♦ **Retour sur les technologies utilisées**
- ♦ **Un aperçu de l'application**



## POURQUOI ESUP-PUBLISHER ?

### Portlet Esup-News vieillissant

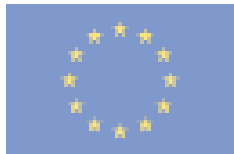
- ♦ Compatibilité mobile
- ♦ Nécessité de mise à jour
- ♦ Améliorer la gestion des permissions



## POURQUOI ESUP-PUBLISHER ?

### Retour des utilisateurs

- ♦ Ergonomie et besoin d'une publication rapide
- ♦ Gestion des permissions sur des groupes
- ♦ Définition du public ciblé lors de la rédaction
- ♦ Publier des contenus différents (actualités, ressources, événements, etc.)



## CHOIX DES TECHNOLOGIES

### Application servlet !

#### ◆ Backend Java en REST

- Spring-boot
- Spring-security
- Spring-data
- Spring-ldap
- Liquibase
- Jackson (json)

## CHOIX DES TECHNOLOGIES

### Application servlet !

#### ♦ Frontend

- AngularJS
- Bootstrap
  
- Outils : npm, Yeoman, Bower, Grunt, Karma

Référence : JHipster

## RETOUR SUR LES TECHNOLOGIES UTILISÉES

### ♦ Spring-boot

- Prise en main pas évidente
- Définition de profiles et des bean programmatiquement (fini le xml !)
- Tomcat embarqué pour l'autonomie

### ♦ Spring-data

- Utilisation de QueryDslPredicateExecutor via les objets Query type très pratique

=> Plus du tout nécessaire de définir des requêtes SQL dans des DAO

- Pagination aisée et quasiment en direct entre l'UI et la base de données

### ♦ Spring-security

- Robuste et adaptable
- Long à mettre en place la première fois avec CAS + mapping de l'utilisateur LDAP + base de données

## RETOUR SUR LES TECHNOLOGIES UTILISÉES

### ♦ AngularJS & Co.

- Agréable pour l'utilisateur mais c'est presque tout !
- À éviter en portlet car peut vite poser des problèmes dans le portail
- Lourd et long à mettre en place, ex : avec l'authentification CAS
- Ne convient pas à un service avec une petite équipe de développeurs
- On trouve trop de « plugins », et à adapter/déboguer selon les besoins
- Complexité des cycles (compile, pré-post link) avec communication entre « plugin » (transclusion)
- Problèmes des versions (problème général aux outils google)
- Utilisations de nombreux outils pour gérer les « plugins », pour la compression, pour le déploiement, pour les tests, etc.



## UN APERÇU DE L'APPLICATION

