

# Maven

- [Introduction](#)
- [mvn.esup-portail.org](#)
- [sonatype & entrepôt maven central](#)
- [Quand et pourquoi publier sur le maven central ?](#)
- [Contacts](#)

## Introduction

Les projets Java issus de la communauté EsupPortail utilisent actuellement principalement Maven ( <https://maven.apache.org/> ) comme 'outil de packaging'.

Cette page tente de tenir à jour les quelques spécificités de l'usage de Maven dans notre communauté EsupPortail.

Maven permet notamment une gestion des dépendances facilitées par la mise en commun des librairies sur des entrepôts Maven communs.

Si maven facilite la gestion des dépendances et donc l'usage et l'intégration de bibliothèques dans les développements Esup en Java, il est aussi parfois utilisé pour le déploiement/exploitation via le 'maven overlay'.

*On note par exemple un tel usage dans la mise en oeuvre des portlets au niveau du packaging du socle Esup/uPortal -voir "the Maven Overlay Strategy" et dans Apereo CAS : <https://github.com/apereo/cas-overlay-template>*

Aussi certains développements Esup sont poussés sur les entrepôts maven dans cet optique.

## mvn.esup-portail.org

Pour certains projets, notamment pour l'usage du framework esup-commons (v2) poussée et préconisée jusqu'aux environs de 2013, EsupPortail a mis en place un entrepôt maven (depuis 2009) spécifique disponible depuis ce lien :

<https://mvn.esup-portail.org>

Dans cet entrepôt, on peut également trouver des librairies modifiées / patchées pour certains développements Esup.

## sonatype & entrepôt maven central

Depuis 2016, EsupPortail peut également pousser ses développements sur le maven central.

Pour ce faire, nous passons par sonatype : <https://oss.sonatype.org>

Pour coller aux conventions de nommage des espaces de noms Java, nous poussons sur le groupId org.esupportail, mais nous poussons également indifféremment sur org.esup-portail. EsupPortail a en effet le contrôle sur l'espace de noms [esupportail.org](https://oss.sonatype.org) en plus de [esup-portail.org](https://mvn.esup-portail.org).

Le ticket ayant initié cette publication pour le groupId org.esupportail se trouve ici <https://issues.sonatype.org/browse/OSSRH-26136>

esup-smsu-api-lib et comme esup-otp-cas utilisent l'espace de nom org.esup-portail, cf <https://issues.sonatype.org/browse/OSSRH-10160>

Pour la mise en oeuvre de la publication sur maven central via sonatype, nous pouvons nous aider du guide donné ici : <http://central.sonatype.org/pages/apache-maven.html>

Comme exemple de projets configurés pour être poussé sur le maven central, vous pouvez regarder le pom.xml d'esup-filemanager : <https://repo1.maven.org/maven2/org/esupportail/portlet/filemanager/esup-filemanager/3.1.2/esup-filemanager-3.1.2.pom>

esup-portlet-sympa est également un bon exemple, d'autant que l'ensemble des modifications effectuées sur le pom.xml et nécessaires à la publication sur sonatype et l'entrepôt maven central ont été ici fait en un seul commit : <https://github.com/EsupPortail/esup-portlet-sympa/commit/35ca51ef90cdd98579ac1c9ebb434b078c8e4b73>

Les jar sont signés avec une clé GPG. Les droits d'upload sur sonatype sont fonction du compte <https://issues.sonatype.org> utilisé pour l'upload vers [oss.sonatype.org](https://oss.sonatype.org) . Le mot de passe ou plus les tokens peuvent être stockés dans settings.xml de maven.

Pour gradle, le fonctionnement est similaire ; voir à ce propos le projet esup-otp-cas et son fichier build.gradle : <https://github.com/EsupPortail/esup-otp-cas/blob/master/build.gradle>

## Quand et pourquoi publier sur le maven central ?

La publication sur le maven central constitue une étape supplémentaire dans la chaîne de distribution de logiciels, cela alourdit encore la procédure de mise à disposition de nouvelles versions (relases) d'une application.

Aussi, suivant l'application, cette étape n'est pas indispensable, ni même opportune : elle doit répondre à un besoin réel et identifié.

La distribution de librairies Java à utiliser dans diverses applications Java justifie cette mise en oeuvre.

esup-otp-cas est un très bon exemple d'une telle mise en oeuvre : l'architecture et la distribution du logiciel Apereo CAS se basent sur des modules Java chargés via gradle depuis les entrepôts Maven.

L'alternative d'utiliser <https://jitpack.io> relié directement à github.com est pratique mais pas sans inconvénient : instabilité des builds du notamment au fait que ceux-ci sont mis en oeuvre dynamiquement et à la demande ; le dépôt dans le maven central répliqué et stable, notamment des versions taguées (releases), est bien plus adapté à la distribution d'un tel logiciel.

## Contacts

Si vous souhaitez obtenir un compte sur l'entrepôt maven esup, pousser vos développements esup sur le maven central, vous pouvez nous contacter :

[coord-tech@esup-portail.org](mailto:coord-tech@esup-portail.org)