

Esup-papercut

Présentation

Esup-papercut est une (petite) servlet affichant les informations du compte **Papercut** de l'utilisateur.

Papercut est un outil payant de gestion d'impressions. Il propose notamment une gestion de quotas d'impressions.

En plus d'afficher le nombre d'impressions et le quota restant, esup-papercut peut permettre à l'utilisateur de recrediter son compte Papercut, ce via un paiement **Paybox** ou/et **IzlyPay**.

Paybox est un service de paiement en ligne (commercial) permettant de se décharger de la procédure de paiement en ligne via cartes bancaires, e-carte bleues, comptes paypal et autres.

IzlyPay est la nouvelle solution poussée par les CROUS permettant aux établissements qui le souhaitent d'intégrer le paiement en ligne dans leurs applications institutionnelles depuis l'espace Izly et donc le crédit Izly de l'utilisateur.

- [Présentation](#)
- [Éléments requis](#)
- [Téléchargement / code source](#)
- [Changelog](#)
 - [Avril 2022 : 2.0.4](#)
 - [Mai 2020 : 2.0.3](#)
 - [Septembre 2019 : 2.0.2](#)
 - [Septembre 2019 : 2.0.1](#)
 - [Août 2019 : 2.0.0](#)
 - [Juillet 2016 : 1.1.2](#)
 - [Février 2016 : 1.1.1](#)
 - [Novembre 2015 : 1.1.0](#)
 - [Janvier 2015 : 1.0.0](#)
- [Configurations](#)
 - [Configurations générales](#)
 - [Base de données](#)
 - [Graphiques / courbes de statistiques](#)
 - [CAS](#)
 - [Configurations du contexte](#)
 - [Configurations papercut](#)
 - [Configurations paybox](#)
 - [Configurations izlypay](#)
 - [Configurations de l'anonymisation](#)
- [Copies d'écran](#)
- [Migration v1 v2](#)
- [Documentations papercut - paybox - izlypay](#)
 - [Paybox](#)
 - [Papercut](#)
 - [IzlyPay](#)
- [Présentation](#)
 - [Présentation EsupDay 28 : 18 septembre 2019](#)
 - [Ancienne présentation](#)
- [Contacts](#)
 - [Crédits & Licence](#)

Éléments requis

Pour fonctionner, esup-papercut requiert :

- **un serveur CAS pour l'authentification / identification**, celui-ci doit permettre à esup-papercut de récupérer en tant qu'**attributs utilisateur CAS** :
 - l'identifiant utilisateur tel qu'utilisé dans papercut
 - l'email de l'utilisateur
 - éventuellement un attribut supplémentaire permettant d'identifier si l'utilisateur doit avoir le rôle ADMIN ou le rôle USER :
Pour que la configuration "esup.contexts.test.esupPapercutCasAttributeRuleUser.uid=." fonctionne par exemple, il faut donc que l'attribut uid soit passé comme attribut utilisateur par CAS*
Si vous avez un doute sur les attributs remontés par CAS à votre service, en mettant org.jasig.cas.client.validation.Cas20ServiceTicketValidator en debug, vous obtiendrez l'assertion complète CAS, celle-ci devra être de la forme suivante, et donc contenir la balise <cas:uid>

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
  <cas:authenticationSuccess>
    <cas:user>toto</cas:user>
    <cas:attributes>
      ....
      <cas:uid>toto</cas:uid>
      ...
      ...
    </cas:attributes>
  </cas:authenticationSuccess>
</cas:serviceResponse>
```

Si il vous manque des attributs, vous devez (re)configurer votre CAS ou tout du moins la configuration du service liée à votre esup-papercut côté CAS pour que celui-ci puisse récupérer les attributs souhaités.

- **la solution Papercut**
 - que celle-ci soit installée et en production dans l'établissement,
 - que celle-ci utilise des comptes dont les identifiants sont connus par CAS,
 - que son Web Service soit accessible depuis esup-papercut - IP déclaré dans le back-end papercut et token d'authentification connu de esup-papercut
- **un compte paybox ou Paylzy est requis si vous souhaitez proposer la fonctionnalité de paiement en ligne**
 - des paramètres d'identification de ce compte doivent être renseignés dans esup-papercut
 - pour paybox, esup-papercut utilise l'intégration paybox **version hmac** (sans module cgi) - il faut donc un compte paybox avec un hmac de configuré.
Actuellement (contrainte paybox, cf documentation paybox [ManuelIntégrationPayboxSystem_V6.1_FR.pdf](#) fin de la page 18) si vous avez un compte paybox utilisé par une application faisant de l'intégration paybox version "module cgi" (ancienne méthode d'intégration), vous ne pouvez pas utiliser celui-ci (il faudra ouvrir un compte paybox supplémentaire).
- En version 2.0, esup-papercut n'est plus portlet mais servlet : esup-papercut est donc **indépendant de l'ENT type EsupPortail/uPortal** - il doit être installé en tant qu'**application web servlet Java** de manière indépendante et autonome.

Téléchargement / code source

<https://github.com/EsupPortail/esup-papercut>

La version **2.0.4** est disponible depuis avril 2022 : <https://github.com/EsupPortail/esup-papercut/>

Changelog

Avril 2022 : 2.0.4

- Mise à jour Spring
- Mise à jour chartjs
- Cache Control et cache busting

Mai 2020 : 2.0.3

- Anonymisation des logs plus vieux de x jours

Septembre 2019 : 2.0.2

- Liens admin/statistiques pour les manager

Septembre 2019 : 2.0.1

- Fixe de sécurité

Août 2019 : 2.0.0

- Passage de portlet à servlet
- Support du paiement via IzlyPay

Juillet 2016 : 1.1.2

- Optimisation de l'export CSV (streaming)
- Mise à jour de la librairie javascript chart.js
- Fixe : statistiques/archivage/droits avec publication multiple de la portlet

Février 2016 : 1.1.1

- Ajout d'un nouveau paramètre (optionnel) forwardUrl en plus de responseServerUrl

- mise à jour des adresses IP des serveurs Paybox (conf par défaut) suite à l'annonce de modification de ces dernières par Verifone (annonce/mail intitulé "**Important Ajout Adresses IP / Nouveaux Datacenters**")

Novembre 2015 : 1.1.0

- Ajout de statistiques sous forme de courbes (images) pour les admins/managers : nombre de transactions et montants par mois - requêtes par défaut adaptées pour PostgreSQL.

Janvier 2015 : 1.0.0

- Consultation du nombre d'impressions et quota restant.
- Formulaire de paiements avec montants prédéfinis (bornés par l'administrateur au travers des préférences portlet) redirigeant sur un formulaire paybox.
- Textes personnalisables via des préférences portlet.
- Support de la publication portlet multiple dans un portail avec des préférences portlets différentes.
- Vue mobile
- Historique des paiements pour un utilisateur.
- Historique des paiements pour l'ensemble des utilisateurs avec système d'archivage (l'admin peut maximiser le montant cumulé par utilisateur via une préférence portlet).
- Export des paiements en CSV

Configurations

Le fichier `src/main/resources/esup-papercut.properties` contient l'ensemble des configurations applicatives : base de données, mail, configurations nécessaires à Papercut, Paybox et IzlyPay.

Le fichier `src/main/resources/logback.xml` quant à lui permet de configurer les logs.

Ce sont ces 2 fichiers que vous devrez modifier pour installer esup-papercut.

esup-papercut peut proposée au travers d'une même instance des contextes d'usages différentes avec des configurations différentes (un contexte de test et un de production par exemple).

Un contexte test est proposé par défaut ; il est défini dans `src/main/resources/esup-papercut.properties` comme l'ensemble des autres configurations (mises à part les logs donc).

Concernant le logo, le plus simple est d'écraser directement le logo esup-portail donné dans le fichier `src/main/resources/static/images/logo.png` en y mettant le logo (png) de votre établissement.

Configurations générales

Base de données

esup-papercut utilise une base de données pour stocker (historiser) l'ensemble des transactions effectuées par l'utilisateur.

Les paramètres de la base sont à donner dans `src/main/resources/esup-papercut.properties` :

```
spring.datasource.driver-class-name=org.postgresql.Driver
spring.datasource.url=jdbc:postgresql://localhost:5432/esuppapercut
spring.datasource.username=esuppapercut
spring.datasource.password=esup
```

La base créée, la construction de la table peut se faire en démarrant une première fois esup-papercut avec dans `src/main/resources/esup-papercut.properties` le paramètre **hibernate.hbm2ddl.auto** à **create**.

Une fois ce premier démarrage effectué et donc la table créée, remettez **hibernate.hbm2ddl.auto** à **update** afin que la base ne soit pas écrasée à chaque redémarrage.

Une base **postgresql** est conseillée - ne serait-ce que par rapport aux requêtes utilisées pour réaliser les graphiques / courbes de statistiques ; par défaut ces requêtes sont propres à postgresql, cf ci-dessous.

Graphiques / courbes de statistiques

Attention, les graphiques (courbes) des statistiques sont construits grâce à des requêtes SQL qui sont liées au SGBD : nous n'avons pas pu développer de requêtes SQL indépendant du SGBD utilisé.

Ces requêtes peuvent être paramétrées lors de la publication de la portlet via les préférences portlet. Si vous utilisez PostgreSQL, vous pouvez laisser ces préférences par défaut, sinon il faut les adapter en regardant <https://github.com/EsupPortail/esup-papercut/blob/master/src/main/java/org/esupportail/papercut/domain/PayboxPapercutTransactionLog.java> pour voir les requêtes postgresql à adapter.

Ainsi requeteNbTransactions correspond en postgresql à :

```
SELECT date_part('year',transaction_date) as year, date_part('month',transaction_date) as month, count(*) as count FROM pay_papercut_transaction_log WHERE papercut_context=:papercutContext GROUP BY year, month ORDER BY year,month
```



Appel à contribution ...

Si vous adaptez ces requêtes à d'autres SGBD comme MySQL par exemple, n'hésitez pas à nous les partager.

CAS

Dans src/main/resources/esup-papercut.properties :

```
cas.key=EsupPapercutCASUnivVille
cas.url=https://cas.univ-ville.fr
cas.service=https://esup-papercut.univ-ville.fr
```

Configurations par contexte

Mis à part esup.default-context qui permet de définir le contexte par défaut à afficher à l'utilisateur (sachant qu'un contexte n'est proposée que si l'utilisateur y a le rôle de USER, MANAGER ou ADMIN), toutes les autres configurations sont données par contexte.

esup.contexts.test.title correspond ainsi au titre à afficher par le contexte test de votre instance esup-papercut.

Configurations du contexte

- **papercutContext** : nom du contexte : doit correspondre au contexte - on a ainsi "esup.contexts.test.papercutContext=test"
- **papercutUserIdAttr** : attribut correspondant à l'identifiant de l'utilisateur dans papercut
- **userEmail** : attribut correspondant à l'email de l'utilisateur ; celui-ci est donné à paybox (si paiement par paybox) pour que paybox puisse envoyer un ticket de paiement à l'utilisateur
- **validatePayboxJustWithRedirection** : permet de valider le paiement paybox via l'interaction navigateur client -> ENT (en plus de paybox -> ENT) : nous conseillons de le laisser à false en production.
- **esupPapercutCasAttributeRuleAdmin.xy** : règle permettant d'affecter le rôle ADMIN à l'utilisateur connecté. xy doit être un attribut utilisateur remonté par CAS, la valeur de cette configuration étant une regexp jouée sur la (ou les) valeur(s) de l'attribut (xy) remonté.
- **esupPapercutCasAttributeRuleManager.xy** : comme esupPapercutCasAttributeRuleAdmin mais pour le rôle MANAGER
- **esupPapercutCasAttributeRuleUser.xy** : comme esupPapercutCasAttributeRuleAdmin mais pour le rôle USER ; si tous les utilisateurs doivent pouvoir accéder au contexte on peut ainsi simplement mettre : "esup.contexts.test.esupPapercutCasAttributeRuleUser.uid=.*"
- **title** : titre de la page
- **htmlFooter** : texte html en "pied de page" de l'application
- **htmlUserHeader** : texte html en entête du contenu de la page
- **htmlUserFooter** : texte html en "pied de page" du contenu de la page
- **montantMin** : montant minimum proposé pour paiement
- **montantMax** : montant maximum proposé pour paiement
- **montantStep** : intervalle de la barre de sélection du montant
- **montantDefaut** : montant proposé par défaut pour paiement
- **papercutSheetCost** : coût d'une page (N&B) en centimes ; permet de donner une indication du nombre de pages par montant ; si -1 aucune indication n'est donnée.
- **papercutColorSheetCost** : coût d'une page couleur en centimes ; permet de donner une indication du nombre de pages par montant ; si -1 aucune indication n'est donnée.
- **transactionNbMax** : nombre de transactions maximal autorisé (les transactions 'archivées' ne sont pas comptées - l'archivage sert à cela ici) : si dépassé, le formulaire de paiement n'est plus accessible ; -1 correspond à ne pas poser de telle contrainte.
- **transactionMontantMax** : montant cumulé des transactions maximal autorisé (les transactions 'archivées' ne sont pas comptées - l'archivage sert à cela ici) : si dépassé, le formulaire de paiement n'est plus accessible ; -1 correspond à ne pas poser de telle contrainte.

Configurations papercut

- **server** : serveur papercut présentant un accès HTTP(S) pour appel du Web Service
- **port** : port du serveur papercut présentant un accès HTTP(S) pour appel du Web Service
- **authToken** : token (secret) partagé avec papercut (configuration disponible dans papercut : Options > Actions > Éditeur de configuration (avancé) > auth.webservices.auth-token).
- **accountName** : préférence portlet accountName - à définir lors de l'utilisation de Papercut avec l'option "comptes multiples" activée.

Configurations paybox

Sous-configurations de esup.contexts.test.paybox (avec test un contexte) ; si vous ne souhaitez pas proposer de formulaires de paiement via paybox, omettez simplement de configurer (commentez les ou supprimez les du fichier de configuration) l'ensemble de ces paramètres esup.contexts.test.paybox.*

- **legende** : légende du formulaire paybox présentant les différents montants disponibles

Si vous utilisiez esup-papercut version 1 en portlet, vous pouvez reprendre les données de l'ancienne version et les convertir pour qu'ils conviennent à la nouvelle version : renommage de table et colonnes.

Le script [src/etc/migrate-v2-v3.sql](#) permet cette migration de données.

Documentations papercut - paybox - izlypay

Paybox

http://www1.paybox.com/telechargement_focus.aspx?cat=3

Papercut

Web Service API : <http://www.papercut.com/products/ng/manual/apdx-tools-web-services.html>

A noter que le logiciel papercut embarque une documentation plus usuelle avec code d'exemple (php et java notamment) à l'appui dans

`[app-path]/server/examples/webservices/`

IzlyPay

Esup-Papercut est un des premiers services à proposer l'intégration de IzlyPay, Esup-Papercut fait ainsi figure d'exemple d'intégration d'IzlyPay dans un service institutionnel.

La documentation d'intégration d'IzlyPay est disponible ici : [Documentation IzlyPay.docx](#)

Pour tout questions à propos d'IzlyPay directement, merci de prendre contact avec departement-vem@cnous.fr

Présentation

Présentation EsupDay 28 : 18 septembre 2019

[EsupPapercut v2 et IzlyPay - diaporama et vidéo de présentation.](#)

Ancienne présentation

Voici une présentation interne d'esup-papercut à l'Université de Rouen Normandie - elle correspond à la version 1 d'esup-papercut, quand esup-papercut était portlet et donc lié à l'ENT EsupPortail/uPortal.

[esup-papercut-presentation.pdf](#)

Contacts

Merci de vous abonner et d'utiliser la liste mail "esup-utilisateurs" pour nous contacter (problèmes techniques notamment) :

<https://listes.esup-portail.org/sympa/info/esup-utilisateurs>

Crédits & Licence

Copyright (C) 2020 Esup Portail <http://www.esup-portail.org>

@Author (C) 2020 Vincent Bonamy <Vincent.Bonamy@univ-rouen.fr>

@Contributor (C) 2020 Jean-Pierre Tran <Jean-Pierre.Tran@univ-rouen.fr>

@Contributor (C) 2020 Valentin Hagnéré <valentin.hagnere1@univ-rouen.fr>

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.