



Etude du réalisable

Intégration de SAP Bo
dans Millésime CCAS version web

Millésime CCAS version web

Sommaire

1	Versions	3
2	Introduction.....	3
	a. Produit concerné	3
	b. Objet	3
	c. Notations	3
3	Paramétrages préalables.....	3
4	Connexion / Déconnexion : utilisation de <i>tokens</i>	4
	a. Connexion	4
	b. Déconnexion.....	5
5	Naviguer dans l'arborescence BO.....	6
6	Modifier les filtres d'un webi.....	9
7	Télécharger un webi	11
	a. pdf.....	11
	b. xls.....	11
	c. xlsx	11
8	Créer un nouveau dossier / webi	12
	a. Nouveau dossier	12
	b. Nouveau webi.....	12
9	Supprimer un webi	13
10	Dupliquer un webi	14
11	Création d'un utilisateur / gestion des droits.....	15
12	Personnalisation de l'interface.....	16
13	Accès direct aux éléments.....	17
	a. Ouvrir un document sans se connecter.....	17
	b. Accéder au Lauchpad (ancien InfoView) sans se connecter.....	17
	c. Accéder au CMC sans se connecter.....	17
14	Technologies envisagées	18
	a. Requêtes HTTP en WLangage.....	18
	b. Utilisation d'AJAX avec JQuery (JavaScript).....	18
	c. Appel à une dll externe avec la fonction API()	18
	d. Utilisation de cURL côté serveur via un fichier batch.....	18
15	Intégrations envisagées	19
	a. Parcours de l'arborescence des fichiers disponibles pour l'utilisateur	19
	b. Ouverture directe en édition d'un webi.....	19
	c. Téléchargement du webi.....	19
	d. Onglet "Infos" : derniers documents mis à jour	19

1 Versions

Date	Version	Evolutions	Auteur
05/02/2014	0.1	Création	Bastien Jorge
20/03/2014	0.2	Retour E. Vattat	Bastien Jorge

2 Introduction

a. Produit concerné

Cette étude concerne le produit Business Object XI 4.1 de la société SAP.

b. Objet

Ce document décrit les possibilités d'accès, d'édition, de création et de suppression offertes par l'API de BO en accès RESTful, sans connexion préalable via l'interface de l'outil.

c. Notations

Lors des requêtes ci-dessous, des variables sont utilisées : %var% dans le code, <nom> dans l'url. Il faudra les remplacer par les valeurs correspondances (6405 pour <port> lors des requêtes REST).

3 Paramétrages préalables

Pour pouvoir fonctionner correctement, les requêtes REST nécessitent des interventions :

- Ouvrir les ports 6400 – 6405 – 6410 (serveur et firewall) en entrant et sortant
- Autoriser REST (dans \Business Objects\Tomcat55\webapps\dswebobje\WEB-INF\conf, éditer axis2.xml, remplacer
`<parameter name="disableREST" locked="true">true</parameter>` par
`<parameter name="disableREST" locked="true">false</parameter>`, sauvegarder et redémarrer le serveur.)
- Vérifier que les fichiers *jquery.min.js* et *bo_filters.js* sont bien présents dans MOL_CCAS_WEB\, ainsi que le plugin *select2*
- Installer cURL sur la machine dans le dossier Exe\
- Déposer le batch Exe\curl\curl_exec.bat

4 Connexion / Déconnexion : utilisation de *tokens*

Avant toute utilisation de BO, il faut pouvoir s'identifier pour accéder aux diverses fonctionnalités. C'est là l'utilité du jeton de connexion.

a. Connexion

Pour générer un jeton de connexion, il faut effectuer la requête REST suivante :

Adresse : <http://<domaine>:<port>/biprws/logon/long>

Requête POST

Header

Key	Value
Accept	application/xml
Content-Type	application/xml

Accept contient le format de la réponse retournée par le serveur BO

Content-Type contient le format des données envoyées au serveur BO

Body

```
<attrs xmlns="http://www.sap.com/rws/bip">
  <attr name="userName" type="string">
    %id%
  </attr>
  <attr name="password" type="string">
    %password%
  </attr>
  <attr name="auth" type="string"
possibilities="secEnterprise,secLDAP,secWinAD">
    secEnterprise
  </attr>
</attrs>
```

Le premier attribut (userName) contient le nom de l'utilisateur (sensible à la casse), le deuxième (password) contient le mot de passe en clair, et le troisième (auth) est le type d'authentification (laisser secEnterprise normalement)



Retour du serveur de la forme de :

Header	
Key	Value
Server	Apache-Coyote/1.1
X-SAP-LogonToken	"webi4.implicit.local:6400@{3&2=6337,U3&2v=webi4.implicit.local:6400,UP&66=60,U3&68=secEnterprise:stagiaire,UP&S9=5820,U3&qe=100,U3&vz=Y61x6aZTvepnBjlcMLu_.sli diV_AeyF7UaPO4IcW2E,UP}"
Date	Wed, 05 Feb 2014 07:41:24 GMT
Content-Type	application/xml
Content-Length	588

Le seul élément important dans le header du retour est le contenu de X-SAP-LogonToken qui contient le jeton de connexion.

Body
<pre><entry xmlns="http://www.w3.org/2005/Atom"> <author> <name>@webi4.implicit.local:6400</name> </author> <id>tag:sap.com,2010:bip-rs/logon/long</id> <title type="text">Résultat de connexion</title> <updated>2014-02-05T07:41:24.733Z</updated> <content type="application/xml"> <attrs xmlns="http://www.sap.com/rws/bip"> <attr name="logonToken" type="string"> webi4.implicit.local:6400@{3&2=6337,U3&2v=webi4.implicit.local:6400,UP&66= 60,U3&68=secEnterprise:stagiaire,UP&S9=5820,U3&qe=100,U3&vz=Y61x6aZTvepnBjlcMLu_.sli diV_AeyF7UaPO4IcW2E,UP} </attr> </attrs> </content> </entry></pre>

On vient de récupérer le token (on n'a pas besoin du body pour l'exploiter).

b. Déconnexion

Une fois le jeton utilisé, il peut être utile de le détruire. Il faudra alors utiliser la requête REST suivante :

Adresse : <http://<domaine>:<port>/biprws/logoff>

Requête POST	
Header	
Key	Value
Accept	application/xml
Content-Type	application/xml
X-SAP-LogonToken	"webi4.implicit.local:6400@{3&2=6337,U3&2v=webi4.implicit.local:6400,UP&66=60,U3&68=secEnterprise:stagiaire,UP&S9=5820,U3&qe=100,U3&vz=Y61x6aZTvepnBjlcMLu_.sli diV_AeyF7UaPO4IcW2E,UP}"

X-SAP-LogonToken contient le jeton à détruire.

5 Naviguer dans l'arborescence BO

Il est possible de visualiser l'arborescence directe des fichiers et dossiers via une requête REST :

Adresse : <http://<domaine>:<port>/biprws/infostore/<file>/children>

Requête GET	
Header	
Key	Value
Accept	application/xml
Content-Type	application/xml
X-SAP-LogonToken	%token%

En fonction du paramétrage du serveur, <file> peut prendre plusieurs valeurs :

- Pour avoir les éléments directement inclus dans la racine :
 - 'Root%20Folder' si le serveur est en anglais
 - 'Dossier%20racine' si le serveur est en français

Si la valeur de <root> est incorrecte, le serveur retournera un statut 404 avec une erreur RWS00009.

Il est intéressant de se poser la question de savoir comment gérer le nom du dossier racine. La solution la plus simple à mettre en place reste probablement de stocker dans la base de données le nom du dossier racine, ainsi que celui du dossier utilisateur (*confer infra*), qui peuvent prendre des valeurs différentes selon les paramétrages de langue du serveur Business Object.

- Un id de dossier pour voir les éléments directement inclus dans ce dossier (ex. '552')
- 'Dossiers%20utilisateur' pour trouver le Dossier Utilisateur racine (nécessite d'être admin pour y accéder) dans un serveur paramétré en français¹.

¹ Il faut donc impérativement faire attention au paramétrage du serveur, car le dossier utilisateur est nécessaire pour pouvoir générer l'arborescence utilisateur.



Retour de la forme :

```

Body
<feed xmlns="http://www.w3.org/2005/Atom">
  <id>tag:sap.com,2010:bip-rs/ASHnC0S_Pw5LhKFbZ.iA_j4/children</id>
  <title type="text">Enfants de Dossier racine</title>
  <updated>2014-02-05T09:22:12.826Z</updated>
  <link href="http://webi4.solutions-implicit.fr:6405/biprws/infostore/Dossier%20racine/children?page=1&pageSize=50" rel="self"/>
  <link href="http://webi4.solutions-implicit.fr:6405/biprws/infostore/Dossier%20racine/children?page=1&pageSize=50" rel="first"/>
  <link href="http://webi4.solutions-implicit.fr:6405/biprws/infostore/Dossier%20racine/children?page=1&pageSize=50" rel="last"/>

  <entry>
    <title type="text">Mes Webis</title>
    <id>tag:sap.com,2010:bip-rs/AVwhi.xFgTdHmHza7wd7.7Q</id>
    <author>
      <name>System Account</name>
    </author>
    <link href="http://webi4.solutions-implicit.fr:6405/biprws/infostore/552" rel="alternate"/>
    <content type="application/xml">
      <attrs xmlns="http://www.sap.com/rws/bip">
        <attr name="id" type="int32">552</attr>
        <attr name="cuid" type="string">AVwhi.xFgTdHmHza7wd7.7Q</attr>
        <attr name="description" type="string"/>
        <attr name="name" type="string">Mes Webis</attr>
        <attr name="type" type="string">Folder</attr>
      </attrs>
    </content>
  </entry>
</feed>

```

Le formatage xml est le suivant :

- Une <entry> par élément contenu
- Pour chaque <entry> :
 - o <title> : Nom du fichier/dossier
 - o <author> : Informations sur l'auteur, principalement son nom (<name>)
 - o <link> : Lien d'accès (<http://<domaine>:<port>/biprws/infostore/<id>>)
 - o <content><attrs><attrs><content>, les différents attributs de l'<entry>, à savoir :
 - name="id" : id de l'<entry>, à utiliser pour parcourir l'arbo
 - name="cuid" : le CUID de l'<entry>, à utiliser pour ouvrir en lien direct (cf. infra)
 - name="description" : la description de l'<entry> (peut être vide)
 - name="name" : le nom de l'<entry>
 - name="type" : le type (Folder, Webi, Pdf...)



Il est possible de connaître les informations de l'élément seul. Pour cela, il suffit de solliciter le serveur BO avec la même requête, en omettant le 'children' :

Adresse : <http://<domaine>:<port>/biprws/infostore/<element>>

Requête GET	
Header	
Key	Value
Accept	application/xml
Content-Type	application/xml
X-SAP-LogonToken	%token%

On obtiendra alors le retour de la forme suivante :

Body
<pre> <entry> <title type="text">Mes Webis</title> <id>tag:sap.com,2010:bip-rs/AVwhi.xFgTdHmHza7wd7.7Q</id> <author> <name>System Account</name> </author> <link href="http://webi4.solutions-implicit.fr:6405/biprws/infostore/552" rel="alternate"/> <content type="application/xml"> <attrs xmlns="http://www.sap.com/rws/bip"> <attr name="id" type="int32">552</attr> <attr name="cuid" type="string">AVwhi.xFgTdHmHza7wd7.7Q</attr> <attr name="description" type="string"/> <attr name="name" type="string">Mes Webis</attr> <attr name="type" type="string">Folder</attr> </attrs> </content> </entry> </pre>

6 Modifier les filtres d'un webi

Avant de modifier un filtre, il faut d'abord s'assurer desquels sont soumis à une invite utilisateur².

Adresse :

<http://<domaine>:<port>/biprws/raylight/v1/documents/<id>/parameters>

Requête GET	
Header	
Key	Value
Accept	application/xml
Content-Type	application/xml
X-SAP-LogonToken	%token%

Retour de la forme :

Body
<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <parameters> <parameter dpId="DP0" type="prompt" optional="false"> <id>0</id> <technicalName>Saisir une ou plusieurs valeurs pour Year</technicalName> <name>Saisir une ou plusieurs valeurs pour Year</name> <answer type="Text" constrained="false"> <info cardinality="Multiple"> <lov refreshable="true" partial="false" hierarchical="false"> <id>UNIVERSELOV_DS0.DObc</id> <values> <value>2004</value> <value>2005</value> <value>2006</value> </values> <columns mappingId="0"> <column type="String" id="0">Year </column> </columns> </lov> <previous> <value>2005</value> <value>2006</value> </previous> </info> <values> <value>2005</value> <value>2006</value> </values> </answer> </parameter> </parameters></pre>

² Si le webi n'est pas soumis à filtres, on obtiendra ce retour :

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<parameters />
```



On peut récupérer :

- L'id de l'attribut à modifier : `<parameters><parameter><id />`
- Le nom de l'attribut à modifier : `<lov><columns><column id="0" />`
- Les valeurs acceptées : `<lov><values><value /><value />`
- Les valeurs actuelles : `<previous>`

De plus, il est intéressant de noter que l'attribut entre les balises `<column><column>` correspond au nom de l'objet dans l'Univers qui a servi de filtre. Cette observation permet, entre autres, de normer le passage de paramètres de manière automatique (consulter la fiche d'un usager en particulier depuis sa fiche personnelle par exemple).

On peut donc maintenant modifier les paramètres :

Adresse :

<http://<domaine>:<port>/biprws/raylight/v1/documents/<id>/parameters>

Requête PUT	
Header	
Key	Value
Accept	application/xml
Content-Type	application/xml
X-SAP-LogonToken	%token%

Body
<pre> <parameters> <parameter> <id>0</id> <answer> <values> <value id="0">2005</value> <value id="1">2006</value> </values> </answer> </parameter> </parameters> </pre>

On donne :

- L'id de l'attribut à modifier `<id>%attr_id%</id>`
- Une valeur de cet attribut `<value id="x">%val%</value>`
(si plusieurs valeurs, incrémenter l'id de value)

Retour de la forme :

Body
<pre> <success> <message>The resource of type 'Document' with identifier '5080' has been successfully updated.</message> <id>5080</id> </success> </pre>

Il faut impérativement penser à utiliser le même token pour voir les applications effectives si l'on cherche à télécharger le document avec les filtres mis à jour.

7 Télécharger un webi

Il est possible de télécharger un webi, sous plusieurs formats finaux, notamment .pdf, .xls et .xlsx

Adresse :

[http://<domaine>:<port>/biprws/raylight/v1/documents/<id>\[parameters\]](http://<domaine>:<port>/biprws/raylight/v1/documents/<id>[parameters])

Requête GET	
Header	
Key	Value
Accept	application/%format%
Content-Type	application/xml
X-SAP-LogonToken	%token%

a. pdf

- Pas de paramètres
- %format% = "pdf"

b. xls

- Paramètres : "optimized=true&dpi=150"
- %format% = "vnd.ms-excel"

c. xlsx

- Paramètres : "optimized=true"
- %format% = "vnd.openxmlformats-officedocument.spreadsheetml.sheet"

Dans tous les cas, **il faudra rediriger la sortie standard vers un fichier pdf, xls ou xlsx.**

8 Créer un nouveau dossier / webi

a. Nouveau dossier

Il est **impossible de créer un nouveau dossier** via la méthode REST, mais il faut se poser la question de la nécessité pour l'utilisateur final de pouvoir en créer un. En effet, l'utilisateur dispose déjà de catégories définies en amont (Aides extérieures, Logement...), il suffit, pour créer un webi, de le générer dans un dossier existant.

De plus, la problématique de la gestion des droits d'accès au dossier met en valeur la nécessité de passer par le CMC.

b. Nouveau webi

Il est possible d'ajouter un nouveau document vierge dans un dossier utilisateur. Cependant, il est auparavant nécessaire d'**activer les droits de création** correspondant à l'utilisateur effectuant la requête, et donc au token généré. Il peut s'agir d'une évolution future pour permettre à l'administrateur BO de créer dynamiquement un nouveau webi vide et de l'ouvrir, mais ce sera écarté à l'heure actuelle.

Adresse : <http://<domaine>:<port>/biprws/raylight/v1/documents>

Requête POST	
Header	
Key	Value
Accept	application/xml
Content-Type	application/xml
X-SAP-LogonToken	%token%

Body
<pre><document> <name>Nom du document</name> <folderId>%id_folder%</folderId> </document></pre>

On récupère ainsi l'id du document nouvellement créé :

Body
<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <success> <message>The resource of type 'Document' with identifier '7289' has been successfully created.</message> <id>7289</id> </success></pre>

9 Supprimer un webi

Pour supprimer un webi, il faut lancer la requête REST suivante :

Adresse : <http://<domaine>:<port>/biprws/raylight/v1/documents/<id>>

Requête DELETE	
Header	
Key	Value
Accept	application/xml
Content-Type	application/xml
X-SAP-LogonToken	%token%

Retour de la forme :

Body
<pre><success> <message>The resource of type 'Document' with identifier 7289 has been successfully removed.</message> <id>7289</id> </success></pre>



10 Dupliquer un webi

Pour dupliquer un document, il faut avoir les droit sur ce document et sur le dossier dans lequel on veut le copier. Il est également nécessaire d'activer les droits de création correspondant au token utilisé.

Adresse : <http://<domaine>:<port>/biprws/raylight/v1/documents?sourceId=<id>>

Requête POST	
Header	
Key	Value
Accept	application/xml
Content-Type	application/xml
X-SAP-LogonToken	%token%
Body	
<pre><document> <name>Nom du document</name> <folderId>%id_folder%</folderId> </document></pre>	

On récupère ainsi l'id de la copie :

Body
<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <success> <message>The resource of type 'Document' with identifier '12816' has been successfully created.</message> <id>12816</id> </success></pre>



11 Création d'un utilisateur / gestion des droits

Via l'API RESTful, il sera impossible de gérer les utilisateurs. Une tentative de solution a cependant été imaginée après navigation dans le dossier utilisateurs : en effet, les utilisateurs eux-mêmes sont considérés comme des fichiers. Ainsi, la possibilité de créer manuellement ce type de fichier (avec les dossiers favorites_folder, inbox, etc., associés, les lier à des groupes...) a été envisagée, mais deux problèmes se sont posés :

- Il n'est, à notre connaissance, pas possible de créer des dossiers via l'API RESTful
- Créer des utilisateurs manuellement peut s'avérer dangereux en cas de changement de système de fonctionnement par SAP.

Mais il serait cependant à priori possible d'effectuer une gestion des utilisateurs via l'API Java (<http://scn.sap.com/community/bi-platform/java-sdk/blog/2013/05/14/scripts-for-user-management>). Il faudra cependant se demander s'il est réellement utile de proposer cette gestion directement sur l'outil Millesime ou s'il vaut mieux gérer les autorisations via le CMC.

De plus, la prise en main de l'API Java est relativement complexe pour l'usage de cette seule fonctionnalité.



12 Personnalisation de l'interface

Changer par exemple le logo SAP pour en mettre un autre est faisable, et à priori simplement en CSS.

Voir ici :

<http://scn.sap.com/community/bi-platform/blog/2013/06/01/tutorial-branding-and-theming-of-bi-launch-pad>



13 Accès direct aux éléments

a. Ouvrir un document sans se connecter

Il est possible d'ouvrir un document sans avoir à se connecter via l'interface BO, grâce à un token. Cela permet notamment d'entrer en fenêtre édition du document (cas d'un webi) ou de visualiser le document via l'interface BO (cas d'un pdf par exemple).

Il suffit d'avoir un token disposant des droits nécessaires pour l'ouverture du webi.

L'url appelée sera la suivante :

[http://\(1\):\(2\)/BOE/OpenDocument/opensdoc/openDocument.jsp?iDocID=\(3\)&siDType=CUID&X-SAP-LogonToken=\(4\)](http://(1):(2)/BOE/OpenDocument/opensdoc/openDocument.jsp?iDocID=(3)&siDType=CUID&X-SAP-LogonToken=(4))

- (1) : Nom ou adresse IP du serveur BO (ex : localhost)
- (2) : Port du serveur de jsp utilisé par BO (ex : 8080)
- (3) : CUID du document à ouvrir (ex : AVOEBpPXOvpGhcrzzE_RAeY)
- (4) : Token (sans les doubles quotes) URLEncodé (en utf8)

b. Accéder au Lauchpad (ancien InfoView) sans se connecter

De la même manière, il faut appeler l'url suivante :

[http://\(1\):\(2\)/BOE/BI/logon/start.do?ivsLogonToken=\(3\)](http://(1):(2)/BOE/BI/logon/start.do?ivsLogonToken=(3))

- (1) : Nom ou adresse IP du serveur BO (ex : localhost)
- (2) : Port du serveur de jsp utilisé par BO (ex : 8080)
- (3) : Token (sans les doubles quotes) URLEncodé (en utf8)

c. Accéder au CMC sans se connecter

Il est impossible d'accéder directement au CMC via un token ou de bypasser le formulaire de login de quelque façon que ce soit sans intervenir lourdement sur le contenu des fichiers de configuration sur le système.



14 Technologies envisagées

a. Requêtes HTTP en WLanguage

La première techno envisagée a été d'utiliser les fonctions WLanguage natives : *HTTPRequête()* et *HTTPEnvoieFormulaire()*. Malheureusement elles ne sont pas suffisamment abouties pour effectuer des requêtes REST complexes : impossible de solliciter un serveur en PUT ou en DELETE ou d'y joindre un body.

b. Utilisation d'AJAX avec JQuery (JavaScript)

Un petit plugin a commencé à être réalisé pour solliciter le serveur BO avec des fonctions AJAX. Mais utiliser JavaScript nous a confronté à deux problèmes non négligeables :

- Il est impossible de générer dynamiquement un fichier à partir du contenu récupéré (PDF, XLS ou XLSX) ou de l'écrire sur le serveur en JavaScript
- Accéder au serveur BO par AJAX oblige à autoriser tous les clients qui vont le solliciter, alors que si le traitement est effectué par le serveur il suffit d'autoriser uniquement ce dernier.

c. Appel à une dll externe avec la fonction API()

Une librairie pour cURL, permettant d'effectuer des requêtes REST, existe (libcurl.dll), mais elle doit être utilisée avec un langage C. Il a été possible de contourner l'usage de constantes internes à la librairie en les recopiant dans la page WebDev, mais il est impossible de passer une fonction callback en WLanguage via la fonction API(), pourtant nécessaire à la récupération du retour de requête.

d. Utilisation de cURL côté serveur via un fichier batch

Cette solution est celle qui a été retenue, car elle a pu contourner toutes les difficultés évoquées précédemment. De plus, cURL est un logiciel Open Source et libre de droits, gratuit même pour une utilisation commerciale.

Pour cela, il faudra installer cURL dans le projet (à l'emplacement `\MOL_CCAS\Exe\curl\`), et déposer dans ce répertoire le fichier batch.

ATTENTION : Le serveur ne doit pas être saturé au niveau de la RAM : les tests en local ont parfois échoué lorsque la mémoire était trop sollicitée et ont causé une erreur 27 – **Out of Memory** (*"A memory allocation request failed. This is serious badness and things are severely screwed up if this ever occurs"*). A noter que la machine sur laquelle tournait le script n'avait que 2Go de mémoire.



15 Intégrations envisagées

a. Parcours de l'arborescence des fichiers disponibles pour l'utilisateur

Dans la mesure où une requête REST permet de connaître le contenu immédiat d'un dossier ainsi que la nature des éléments (fichiers ou dossiers) s'y trouvant, il est aisé, via récursivité, de connaître l'arborescence complète du Dossier racine.

En pratique, on n'utilisera pas la récursivité du fait du temps de réponse très long du serveur, on se contentera d'une recherche des fils immédiat dès que l'utilisateur cherchera à accéder au contenu d'un dossier.

b. Ouverture directe en édition d'un webi

Une fois le token récupéré, on peut générer un lien "OpenDocument" pour ouvrir directement le webi en édition. Ou un fichier (type pdf) en lecture via l'interface BO.

c. Téléchargement du webi

BO permet de récupérer le contenu d'un webi et de le rediriger vers un fichier. Ainsi, il est possible de permettre à l'utilisateur de télécharger le webi au format pdf, xls ou xlsx, après avoir mis à jour les différents filtres du webi si nécessaire.

d. Onglet "Infos" : derniers documents mis à jour

Les requêtes RESTful peuvent accéder à la boîte de réception BO d'un utilisateur. Il est donc tout à fait envisageable d'y récupérer les instances générées pendant la nuit par exemple, et de les afficher dans l'onglet Infos (pour l'instant inutilisé) afin de mettre en valeur les dernières mises à jour.

Il faudra donc paramétrer les planifications de manière rigoureuse, notamment en sélectionnant une notification à chaque utilisateur concerné.