cockpit
IT Service Manager

Tickets - Synchroniser Cockpit ITSM avec Jira Software

Document FAQ

Table des matières

Introduction 3
I.Objectif 3
II.Principes 3
Paramétrage 4
I.Types de tickets 4
II.Statuts 4
A.Cockpit ITSM 4
B.Jira Software 5
III.Priorités 6
A.Cockpit ITSM 6
B.Jira Software 6
IV.Utilisateur 6
A.Gestion de l'utilisateur 6
B.Jeton API 7
V.Équipes 7
A.Cockpit ITSM 7
B.Jira Software 7
Source fichier XML 8
I.Balise <connector> 8
II.Processus synchronisation 10
A.Type de ticket 10
B.Mappage des IDs des tickets 10
C.Création d'un ticket 11
D.Mise à jour d'un ticket 13
E.Clôture des tickets 15

Introduction

I. Objectif

Le but de ce document est de présenter un exemple fonctionnel de synchronisation entre un portail Cockpit ITSM et une instance Jira Software.

II. Principes

- Se reporter au document « FAQ - Synchronisation Guide d'utilisation » pour le fonctionnement et l'utilisation du menu de Cockpit ITSM.
- En consultant ce document, se reporter simultanément au document « FAQ - Synchronisation Jira Source Code XML », c'est ce document source qui est expliqué dans le document présent.
- Une bonne maîtrise des solutions Jira Software et Cockpit IT Service Manager est nécessaire.

Paramétrage

Objectifs : Avant de rédiger le code XML destiné à synchroniser un portail Cockpit ITSM avec une instance Jira Software, les quelques éléments de paramétrage suivants sont à prendre en compte.

I. Types de tickets

Cockpit ITSM peut synchroniser 3 types de tickets correspondant aux 3 types de processus ITIL (Demande, Changement, Incident).

Dans Jira Software il existe plusieurs types de tickets (Task, Bug, etc.) et il est possible de créer des types de tickets personnalisés.
Les différents types de tickets peuvent être synchronisés avec Cockpit ITSM, pour identifier les types et termes à utiliser dans le code XML :

- Dans Jira aller dans le menu « Paramètres Jira > Tickets > Types de ticket ».
- Pour synchroniser un type de ticket, relever la valeur de son champ « Nom » qui sera à utiliser dans le code « XML » (ne pas utiliser une traduction, cliquer sur « Modifier » pour avoir le terme exact).

> Note : Dans Jira, les types de ticket appartiennent à des systèmes de types de ticket et un système de types de ticket est utilisé par un projet.
> Ainsi quand on créé un ticket, les types disponibles dépendent du projet du ticket.

Le tableau ci-dessous propose une correspondance entre les types de tickets pouvant être synchronisés :

Cockpit ITSM	Jira
REQUEST	TASK
INCIDENT	BUG
CHANGE	Autre (Story, etc.) ou personnalisé

II. Statuts

Principe :
Les statuts des tickets vont être mappés entre Cockpit ITSM et Jira.
Exemple ci-dessous, nous mappons le statut Jira « Open » avec le statut Cockpit ITSM « New » :

```
<valueMap externalValue="Open" cockpitValue="New"/>
```

A. Cockpit ITSM

Menu : Tickets > Configuration > Incidents / Demandes / Changements > Statuts
Principes :
Utiliser la valeur du champ « Reference » des statuts pour les désigner dans le code XML.

B. Jira Software

1. États

Menu : Paramètres Jira > Tickets > Caractéristiques du ticket > États

Principes :

Cliquer sur l'action « Modifier » de l'état que vous souhaitez désigner dans le fichier XML, utiliser la valeur du champ « Nom ».

Les états de Jira correspondent aux statuts de Cockpit ITSM.

2. Workflows

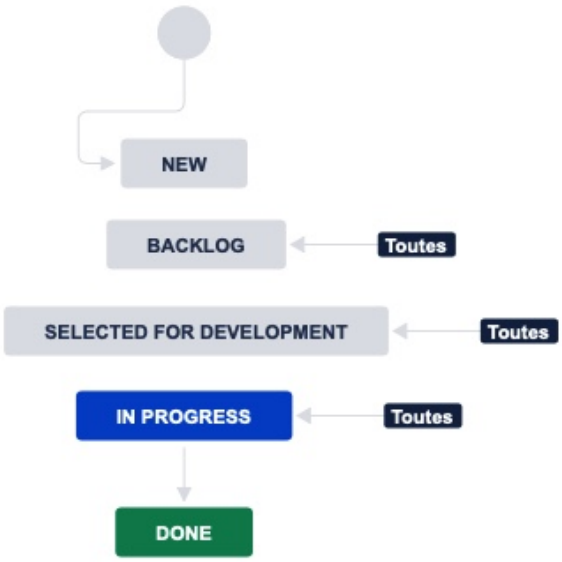
Menu : Paramètres Jira > Tickets > Flux de travail > Flux de travail

Principes :

Les différents statuts que peut avoir un ticket sont liés au flux de travail (workflow) du ticket.

Dans un workflow certains statuts peuvent être attribués directement aux tickets, d'autres statuts ne peuvent être attribués que par une ou plusieurs transitions, c'est-à-dire des statuts « précédents ». Le but étant d'imposer un cycle de vie aux tickets.

Dans l'exemple ci-dessous, le statut « In progress » peut être attribué à partir de n'importe quel statut mais le statut « Done » ne peut l'être qu'après le statut « In progress ».



C'est un élément important à prendre en compte lors du paramétrage du fichier XML, dans cet exemple si un ticket Cockpit est pris en compte puis clôturé (statut « Done ») sans passer par un statut « In progress », le statut du ticket dans Jira ne pourra être être synchronisé avec celui de Cockpit.

III. Priorités

Principes :

Les priorités des tickets vont être mappés entre Cockpit ITSM et Jira Software.

Exemple ci-dessous, nous mappons la priorité Jira « Highest » avec la priorité Cockpit ITSM « High » :

```
<valueMap externalValue="Highest" cockpitValue="High"/>
```

A. Cockpit ITSM

Menu : Tickets > Configuration > Incidents / Demandes / Changements > Priorités

Principes :

- Utiliser le champ « Reference » des priorités pour les désigner dans le code XML lors de l'association avec les priorités de Jira Software.
- Vérifier si le champ « Priorité » est sélectionné dans le modèle de ticket utilisé pour la création des tickets lors de la synchronisation.
- Vérifier les priorités qui peuvent être utilisées dans le catalogue de l'organisation.

B. Jira Software

Menu : Paramètres Jira > Tickets > Caractéristiques du ticket > Priorités

Principes :

Cliquer sur l'action « Modifier » de la priorité que vous souhaitez désigner dans le fichier XML, utiliser la valeur du champ « Nom ».

IV. Utilisateur

Principe : Un utilisateur Jira sera nécessaire pour la connexion au portail Jira Software ainsi que pour les actions sur les tickets. Il est fortement conseillé de dédier un utilisateur pour les actions de synchronisation.

A. Gestion de l'utilisateur

Menu : Aller dans « User Administration > Users » et créer un nouvel utilisateur.

Paramétrage :

- Dans la balise <connector /> il faudra indiquer l'email de l'utilisateur dans le paramètre « username ».
- Le nom de l'utilisateur (termes utilisés dans le champ « rapporteur » d'un ticket par exemple) sera à utiliser dans les filtres des balises <create />, <update /> et <close />.
- Rôles : l'utilisateur doit avoir suffisamment de droits pour manipuler les tickets.
- Time zone : Afin d'éviter des problèmes de synchronisation, l'utilisateur final Cockpit ITSM et l'utilisateur Jira doivent utiliser la même TimeZone qui est déclarée dans le paramètre « defaultTimeZone ».

Il est conseillé de paramétrer ces 2 utilisateurs en TimeZone UTC.

Dans les paramètres personnels de l'utilisateur Jira, indiquer « GMT Offset » dans le champ de la région et « (GMT+00:00) GMT » dans le champ du fuseau horaire. Dans le paramètre « defaultTimeZone » de la balise connector il faudra alors indiquer « UTC ».

> Note : Si vous utilisez une autre TimeZone que UTC, le principe sera le même, indiquer « Région/Fuseau », « Europe/Paris » par exemple.

B. Jeton API

Menu : Depuis l'utilisateur Jira aller dans « Paramètre de compte > Sécurité > Jeton d'API > Créer et gréer des jetons d'API » et créer un nouveau jeton d'API.

Paramétrage :

Dans le paramètre « password » de la balise <connector /> il faudra indiquer le jeton API précédemment créé.

V. Équipes

Principe : Synchroniser les équipes propriétaires des tickets Cockpit et Jira.

> Note : Dans Cockpit ITSM un ticket appartient obligatoirement à une équipe.

A. Cockpit ITSM

Menu : Administration > Équipes

Principes :

- Éditer une équipe et relever son nom, utiliser le nom de l'équipe dans le fichier XML.

B. Jira Software

Menu : Paramètres Jira > Personnes

Principes :

- Éditer une équipe, relever l'ID apparaissant dans l'Url, exemple : https://xxxxx.atlassian.net/jira/people/team/**ad77ab65-9522-497f-a6a0-01fb61476bb6**
  Utiliser l'ID pour désigner l'équipe dans le fichier XML.

```
<lookup targetType="team" externalField="assignment_group" cockpitField="assignedTeam" direction="IN">
    <valueMap externalValue="ad77ab65-9522-497f-a6a0-01fb61476bb6" cockpitValue="SUPPORT"/>
</lookup>
```

Source fichier XML

Objectifs :

- Présenter un cas concret de fichier XML permettant de synchroniser une instance Cockpit ITSM avec une instance Jira Software.
- L'exemple peut servir de base pour des synchronisations avec d'autres instances Jira, vous devriez être capable d'adapter le fichier source à votre contexte.

I. Balise <connector>

```
<?xml version="1.0" encoding="UTF-8"?>
<ticketSync xmlns="http://www.cockpit-itsm.com/TicketSyncConfiguration" description="Jira"
syncFrequencyInMinutes="5">
```

- Ne pas modifier la valeur de l'attribut « **xmlns** ».
- Indiquer dans l'attribut « **syncFrequencyInMinutes** » la fréquence de la synchronisation en minutes.

```
<connector id="jira" username="xxxx@yyyy.com" password="xxxxxxxx" url="https://xxxxxxxx.atlassian.net"
defaultTimeZone="Europe/Paris" dateFormat="yyyy/MM/dd HH:mm">
```

- **Début de la balise <connector>.**
- « **id** » indique quel type de connecteur vous utilisez, ici le type est « jira ».
- « **username** » et « **password** » sont respectivement l'email de l'utilisateur et le jeton API créés dans la partie précédente « Utilisateurs ».
- « **url** » est l'url d'accès à l'instance Jira Software.
- « **defaultTimeZone** » : Time Zone utilisé pour interpréter les champs dates de l'instance Jira. Indiquer « UTC » si le TimeZone des paramètres de l'utilisateur Jira a été paramétré en GMT+00:00 comme vu dans la partie précédente « Utilisateurs ».
- « **dateFormat** » : Format de date de l'utilisateur, le format « yyyy/MM/dd HH:mm » est le format par défaut. Il peut être modifié côté Jira dans les paramètres avancés.

```
  <parameters>
   <entry>
      <string>traceDirectory</string>
      <string>/tmp/Jira</string>
   </entry>
```

- Le répertoire des logs se trouve sur la machine hébergeant le portail, si vous êtes en mode Saas vous n'accédez pas à cet élément, vous devez donc enlever cette balise <entry />.
- Si vous êtes en mode Premium, remplacer le chemin « /tmp/Jira » par le chemin adéquat.

```
        <entry>
```

```
    <string>maxResults</string>
    <int>50</int>
</entry>
```

- Le nombre d'enregistrement remontés à chaque synchronisation est limité à 100 (limitation liée à l'API Jira). Ajuster le cette limite selon l'activité des tickets et la fréquence de synchronisation ou mettez directement le maximum (100).

> Note : Si la limite d'enregistrements est dépassée lors d'une synchronisation, les données excédentaires – ex : création d'un ticket – ne seront pas prises en compte.

```
<entry>
    <string>dateFields</string>
    <string>
        created,
        updated,
    </string>
</entry>
```

- Déclaration des champs dates (création et mise à jour) des tickets Jira.

```
<entry>
    <entry>
    <string>fields</string>
    <string>
        attachment,
        comment,
        created
        description,
        id,
        issuetype,
        priority,
        project,
        reporter,
        resolution,
        resolutiondate,
        status,
        summary,
        updated
    </string>
</entry>
```

- Liste des champs pris en compte lors de la synchronisation. Si vous souhaitez que d'autres champs soient pris en compte, il faut les ajouter aux requêtes API, contacter alors le support Cockpit ITSM pour effectuer cette demande.
- Les champs ajoutés pourront être synchronisés avec des propriétés spécifiques Cockpit.

```
    <entry>
      <string>journalFields</string>
      <string>comments,work_notes</string>
```

```
    </entry>
```

- Permet d'inclure les champs du journal au moment de la création d'un ticket.
- La deuxième balise <string> contient la liste des champs du journal à synchroniser. Dans l'exemple ci-dessus les champs « comments » (pour les commentaires) et « work_notes » (pour les notes de travail). Pour ne pas synchroniser ces champs ne rien indiquer dans cette balise.
- Ne pas modifier la balise « journalFields ».

```
    <entry>
      <string>readOnlyFields</string>
      <string>
        __type__,
        created,
        id,
        reporter,
        updated
      </string>
    </entry>
```

- Déclaration des champs qui ne seront pas modifiés lors de la synchronisation (lecture seule).

```
<lastUpdateField>updated</lastUpdateField>
```

- Associe la date des opérations sur les tickets externes afin de maintenir l'ordre chronologique des opérations des 2 côtés (Cockpit ITSM et Jira).

- **Fin de la balise <connector>.**

II. Processus synchronisation

- Dans l'exemple suivant nous décrivons un processus de synchronisation entre les tickets de type « Incident » côté Cockpit ITSM et « Bug » côté Jira.
- La synchronisation se fera dans les 2 directions (Cockpit ITSM => Jira et inversement).
- Le processus traite séparément les créations, mises à jour et clôtures.

A. Type de ticket

```
<process ticketType="INCIDENT" discriminatorField="__type__" discriminatorValue="Bug"
ticketTemplate="JIRA">
```

- Le processus de synchronisation d'un type de ticket commence avec la balise <process />
- Attribut « **discriminatorField** » : Correspond au type du ticket
- Attribut « **discriminatorValue** » : Correspond au processus INCIDENT
- Attribut « **ticketTemplate** » : Correspond au champ « Reference » du modèle qui sera utilisé lors de la création d'un ticket côté Cockpit ITSM provenant de Jira.

B. Mappage des IDs des tickets

```
<ticketIdMap externalField="id" cockpitField="externalReference"/>
```

- La balise <ticketIdMap /> mappe les tickets Jira avec les tickets Cockpit ITSM :
  - Attribut « **externalField** » : contient la valeur du champ « id » qui correspond au champ ID des tickets Jira.
  - Attribut « **cockpitField** » : contient l'ID du ticket côté Cockpit ITSM.
- Si un ID de ticket n'est pas associé à un autre ID, alors nous considérons qu'il s'agit d'une création.
- Quand un ID est associé à un autre ID, selon le statut nous considérons une mise à jour ou une clôture.

C. Création d'un ticket

```
<create direction="BOTH">
```

- La balise <create /> traite de la création des tickets.
- La valeur « **BOTH** » de l'attribut « direction » indique que les tickets créés dans Jira seront créés dans Cockpit et inversement. Il est donc possible avec les valeurs « IN » (vers Cockpit) et « OUT » (vers Jira) de synchroniser les 2 systèmes uniquement dans un sens.

```
<filters>
  <filter direction="IN">
    <filterRule field="__operation__" value="CREATE"/>
    <filterRule field="reporter" value="API cockpit" inverted="true"/>
  </filter>
</filters>
```

- En premier lieu on applique un filtre sur les tickets entrants (le « IN » dans l'attribut « direction »), puis on ignore les tickets Jira créés par l'utilisateur Jira déclaré précédemment dans la balise <connector /> utilisé pour se connecter à Jira et pour les actions sur les tickets.

> Important : Dans l'attribut « value="API cockpit" » il ne faut pas indiquer l'email de l'utilisateur comme dans l'attribut « username » de la balise <connector />, mais le nom / prénom de cet utilisateur. Pour être sûr de la valeur à renseigner, créez un ticket Jira avec cet utilisateur et reprenez la valeur du champ « Rapporteur ».

- Sans ce filtre nous pourrions entrer dans une boucle infinie.
- Si la synchronisation ne va que dans un sens, ce filtre n'est pas utile.

1. Mappage des statuts et priorités : Jira => Cockpit ITSM

```
<mappers>
<copy externalField="id" cockpitField="externalReference" direction="IN"/>
<copy externalField="summary" cockpitField="title" stripHtml="true" direction="IN"/>
<copy externalField="description" cockpitField="description" stripHtml="true" direction="IN"/>
  <copy externalField="created" cockpitField="creationDate" direction="IN"/>
```

- La balise <mappers> indique que l'on va mapper les différentes valeurs d'un champ.
- La balise <copy> sert à mapper les champs sans modifier le contenu, c'est le cas des champs texte (exemple : titre des tickets).

- L'attribut « stripHtml » sert à conserver ou non la mise à en forme Html du texte :
  - True : le format Html n'est pas conservé
  - False : le format Html est conservé, c'est également la valeur par défaut quand l'attribut « stripHtml » n'est pas renseigné.
- Dans cette partie on copie des informations comme la description, l'ID du ticket, etc.

```
<lookup targetType="status" externalField="status" cockpitField="status" direction="IN">
  <valueMap externalValue="Open" cockpitValue="New"/>
  <valueMap externalValue="Reopened" cockpitValue="New"/>
  <valueMap externalValue="To Do" cockpitValue="To do"/>
  <valueMap externalValue="In Progress" cockpitValue="In progress"/>
  <valueMap externalValue="Building" cockpitValue="Waiting"/>
  <valueMap externalValue="*" cockpitValue="New"/>
</lookup>
```

- Tous les champs des tickets qui ne seront pas mappés seront ignorés.
- L'attribut **direction** avec la valeur « IN » dans la balise <lookup /> s'applique à tous les attributs <valueMap /> suivants.
- On utilise la balise <lookup> pour le mappage des champs qui ont des valeurs prédéfinies (ici les statuts).
- On commence par mapper les ID des champs avec les attributs **externalField** et **cockpitField**.
- Puis on mappe les valeurs des champs avec **externalValue** et **cockpitValue**.

> Note : Pensez à avoir une balise <valueMap> avec la valeur « * », au cas où un statut Jira n'est pas prévu dans le mappage, un ticket est tout de même créé.

```
<lookup targetType="priority" externalField="priority" cockpitField="priority" direction="IN">
  <valueMap externalValue="Highest" cockpitValue="High"/>
  <valueMap externalValue="High" cockpitValue="High"/>
  <valueMap externalValue="Medium" cockpitValue="Medium"/>
  <valueMap externalValue="Low" cockpitValue="Low"/>
  <valueMap externalValue="Lowest" cockpitValue="Low"/>
</lookup>
```

- Dans cette partie <lookup> les priorités des tickets Jira sont mappées avec les priorités Cockpit ITSM. Il faut prendre en compte toutes les priorités pour synchroniser tous les tickets.

```
<lookup targetType="team" externalField="assignment_group" cockpitField="assignedTeam"
direction="IN">
    <valueMap externalValue="ad77ab65-9522-497f-a6a0-01fb61476bb6" cockpitValue="NETWORK"/>
    <valueMap externalValue="*" cockpitValue="SUPPORT"/>
</lookup>
```

- Synchronisation des équipes, dans cet exemple les tickets de l'équipe Jira « ad77... » sont associés à l'équipe Network de Cockpit. Les tickets de toutes les autres équipes Jira sont associés à l'équipe Support Cockpit.

- Penser à prévoir une valeur « * » pour prendre en compte tous les tickets. Au contraire il est possible de ne synchroniser que les tickets de certaines équipes.

2. Mappage : Cockpit ITSM => Jira

```
<set externalField="__type__" cockpitField="N/A" value="Bug" direction="OUT"/>
<copy externalField="project" cockpitField="N/A" value="SUPPORT_COCKPIT" direction="OUT"/>
<copy externalField="summary" cockpitField="title" stripHtml="true" direction="OUT"/>
<copy externalField="description" cockpitField="description" wikiMarkup="true" direction="OUT"/>
```

- Dans Jira un ticket appartient à un projet, ici le projet dans Jira est « SUPPORT_COCKPIT ».
- L'attribut « wikiMarkup » est l'équivalent de « stripHtml » mais côté Jira. Dans la synchronisation « OUT » (Cockpit => Jira) il faut donc utiliser cet attribut pour conserver ou non le texte riche :
  - True : Le texte riche est conservé
  - False : Le texte riche n'est pas conservé

```
<lookup targetType="priority" externalField="priority" cockpitField="priority" direction="OUT">
    <valueMap externalValue="Highest" cockpitValue="High"/>
    <valueMap externalValue="High" cockpitValue="High"/>
    <valueMap externalValue="Medium" cockpitValue="Medium"/>
    <valueMap externalField="Low" cockpitValue="Low"/>
    <valueMap externalField="Lowest" cockpitValue="Low"/>
</lookup>
</mappers>
</create>
```

- Cette partie correspond au mappage des créations d'incidents de Cockpit ITSM vers Jira Software, le principe est le même que dans la partie précédente (Jira Software => Cockpit ITSM).
- Tous les attributs « direction » ont la valeur « **OUT** ».
- Les priorités Cockpit ITSM sont associées aux propriétés Jira.
- Les statuts ne sont pas mappés car quand on créé un ticket dans Jira, le statut dépend du système de flux de travail associé au projet où le ticket est créé.
- Fin du processus de création d'un ticket.

D. Mise à jour d'un ticket

```
<update discriminatorField="__operation__" discriminatorValues="UPDATE" attachmentPrivacy="PUBLIC"
direction="BOTH">
```

- La balise <update /> traite des mises à jour des tickets.
- Attribut « **discriminatorField** » : la valeur « __operation__ » indique que l'on s'intéresse aux actions sur les tickets.
- Attribut « **discriminatorValue** » : la valeur « UPDATE » indique que l'on prend en compte uniquement les opérations de mise à jour.
- Attribut « **direction** » : La mise à jour des tickets se fera dans les 2 sens, de Jira Software vers Cockpit ITSM et inversement.

```
<filters>
   <filter direction="IN">
      <filterRule field="updateAuthor" value="API Cockpit" inverted="true"/>
    <filterRule field="reporter" value="API Cockpit" inverted="true"/>
   </filter>
   <filter direction="OUT">
      <filterRule field="privacy" value="PUBLIC"/>
   </filter>
   <filter direction="OUT">
      <filterRule field="type" value="EDIT_PROPERTIES"/>
   </filter>
   <filter direction="OUT">
      <filterRule field="type" value="EDIT_TREATMENT"/>
   </filter>
   <filter direction="OUT">
      <filterRule field="type" value="ATTACHMENT_ADD"/>
   </filter>
</filters>
```

- Comme pour la création des incidents, un filtre est appliqué sur les tickets entrants pour ignorer les tickets Jira créés par l'utilisateur Jira « API Cockpit », afin d'éviter une boucle infinie.
- Un filtre sur les tickets sortants est appliqués pour ne prendre en compte que les mises à jour de ticket Cockpit ITSM effectuées avec un niveau niveau de confidentialité « Public ».
- Deux filtres sur les tickets sortants sont appliqués pour mettre à jour les tickets Jira quand les propriétés du ticket Cockpit ITSM sont modifiées :
  - « EDIT_PROPERTIES » : Les changements de priorité sont synchronisés
  - « EDIT_TREATMENT » : Les changements de statuts sont synchronisés

  Le principe est le suivant :
  - Les modifications des priorité et statut des tickets Cockpit ITSM se font en mode privé, si le filtre ne prenant en compte que les échanges publics est paramétré, les modifications des priorités et statuts ne seraient pas prises en compte lors de la synchronisation.

    C'est pourquoi il est nécessaire d'ajouter les 2 filtres « EDIT_PROPERTIES » et « EDIT_TREATMENT ».
  - Si le filtre « PUBLIC » n'est pas mis, toutes les modifications de tickets Cockpit ITSM, publiques ou privées, sont prises en compte, il n'est alors pas utile d'ajouter les filtres « EDIT_PROPERTIES » et « EDIT_TREATMENT ».

1. Mappage des statuts, priorités et équipes : Jira => Cockpit

```
<mappers>
<copy externalField="description" cockpitField="transientMessage" stripHtml="true" direction="IN"/>
<template externalField="body" cockpitField="transientMessage" direction="IN"><![CDATA[$data["body"]
<br />
$data["updateAuthor"]]]></template>
<copy externalField="updated" cockpitField="lastUpdate" direction="IN"/>
```

```
<lookup targetType="status" externalField="status" cockpitField="status" direction="IN">
	<valueMap externalValue="Open" cockpitValue="New"/>
	<valueMap externalValue="Reopened" cockpitValue="New"/>
	<valueMap externalValue="To Do" cockpitValue="To do"/>
	<valueMap externalValue="In Progress" cockpitValue="In progress"/>
	<valueMap externalValue="Building" cockpitValue="Waiting"/>
</lookup>

<lookup targetType="priority" externalField="priority" cockpitField="priority" direction="IN">
	<valueMap externalValue="Highest" cockpitValue="High"/>
	<valueMap externalValue="High" cockpitValue="High"/>
	<valueMap externalValue="Medium" cockpitValue="Medium"/>
	<valueMap externalValue="Low" cockpitValue="Low"/>
	<valueMap externalValue="Lowest" cockpitValue="Low"/>
</lookup>
```

- Même fonctionnement que pour la création de tickets, le statut des tickets pouvant être synchronisé.
- La balise template <template /> permet d'ajouter du texte et des variables. Dans cet exemple on ajoute à la description du ticket (« body ») un saut de ligne (
) puis le nom de l'utilisateur Jira qui a ajouté l'échange (variable « updateAuthor »). Il aurait été également possible d'ajouter du texte. Pour voir la liste des variables disponibles, contacter le support Cockpit ITSM.

2. Mappage : Cockpit => Jira

```
<copy externalField="id" cockpitField="externalReference" direction="OUT"/>
<copy externalField="body" cockpitField="transientMessage" wikiMarkup="true" direction="OUT"/>

<lookup targetType="status" externalField="status" cockpitField="status" direction="OUT">
	<valueMap cockpitValue="To Do" externalValue="New"/>
	<valueMap cockpitValue="To Do" externalValue="To do"/>
	<valueMap cockpitValue="In Progress" externalValue="In progress"/>
	<valueMap cockpitValue="Building" externalValue="Waiting"/>
	<valueMap cockpitValue="Done" externalValue="Solved"/>
</lookup>

<lookup targetType="priority" externalField="priority" cockpitField="priority" direction="OUT">
	<valueMap externalValue="Highest" cockpitValue="High"/>
	<valueMap externalValue="High" cockpitValue="High"/>
	<valueMap externalValue="Medium" cockpitValue="Medium"/>
	<valueMap externalValue="Low" cockpitValue="Low"/>
	<valueMap externalValue="Lowest" cockpitValue="Low"/>
</lookup>
</mappers>
</update>
```

- Mêmes principes que pour la création de tickets.

E. Clôture des tickets

```
<close discriminatorField="status" discriminatorValues="Done" attachmentPrivacy="PUBLIC"
direction="BOTH">
        <filters>
           <filter direction="IN">
                <filterRule field="updateAuthor" value="API Cockpit" inverted="true"/>
             <filterRule field="reporter" value="API Cockpit" inverted="true"/>
            </filter>
         </filters>
```

- La balise <close /> traite de la clôture des tickets.
- Nous prenons en compte que l'état de ticket Jira « Done » qui correspond à une clôture de ticket.
- Les clôtures sont prises en compte depuis Cockpit ITSM et Jira (paramètre « BOTH »).
- Un filtre entrant « IN » permet de ne pas prendre en compte les clôtures effectuées par l'utilisateur Jira « API Cockpit » et évite d'entrer dans une boucle infinie.

1. Mappage des champs Jira => Cockpit ITSM

```
<mappers>
<copy externalField="body" cockpitField="transientMessage" direction="IN"/>
<copy externalField="updated" cockpitField="lastUpdate" direction="IN"/>
<copy externalField="updated" cockpitField="realSolutionDate" direction="IN"/>

<lookup targetType="status" externalField="status" cockpitField="status" direction="IN">
        <valueMap externalValue="Done" cockpitValue="Solved"/>
        <valueMap externalValue="*" cockpitValue="Solved"/>
</lookup>

<lookup targetType="priority" externalField="priority" cockpitField="priority" direction="IN">
        <valueMap externalValue="Highest" cockpitValue="High"/>
        <valueMap externalValue="High" cockpitValue="High"/>
        <valueMap externalValue="Medium" cockpitValue="Medium"/>
        <valueMap externalValue="Low" cockpitValue="Low"/>
        <valueMap externalValue="Lowest" cockpitValue="Low"/>
</lookup>
```

- Les champs Cockpit « lastUpdate » et « realSolutionDate » sont obligatoires, ils contiendront la même valeur provenant du champ Jira « updated ».
- Le champ Cockpit « realSolutionDate » correspond à la clôture définitive du ticket effectuée par les utilisateurs finaux, le ticket ne peut être réouvert.

2. Mappage des champs Cockpit ITSM => Jira

```
<copy externalField="id" cockpitField="externalReference" direction="OUT"/>
<copy externalField="body" cockpitField="response" wikiMarkup="true" direction="OUT"/>

<lookup targetType="status" externalField="status" cockpitField="status" direction="OUT">
        <valueMap externalValue="Done" cockpitValue="Solved"/>
        <valueMap externalValue="Done" cockpitValue="*"/>
</lookup>
</mappers>
```

</close>

Fin du document