

# **BPMN Camp - Saison 3**

## **Exercices**

---

## **BPMN Camp - Saison 3: Exercices**

Copyright © 2015 Bonitasoft

---

# Table des matières

Préface .....	iv
1. Exercice : Modélisation basique de processus .....	1
Objectif .....	1
Instructions résumées .....	1
Instructions pas à pas .....	1
2. Exercice : Ajout de données et spécification de contrats .....	3
Objectif .....	3
Instructions résumées .....	3
Instructions pas à pas .....	4
3. Exercice : Contrôle dynamique du flux d'exécution grâce aux données .....	7
Objectif .....	7
Instructions résumées .....	7
Instructions pas à pas .....	7
4. Exercice : Création de formulaires .....	10
Objectif .....	10
Instructions résumées .....	10
Instructions pas à pas .....	10
5. Exercice : Gestion d'acteurs .....	16
Objectif .....	16
Instructions résumées .....	16
Instructions pas à pas .....	16
6. Exercice : Notification par le biais d'un connecteur .....	18
Objectif .....	18
Instructions résumées .....	18
Instructions pas à pas .....	18

---

# Préface

L'objectif des exercices est de parvenir à produire un processus basique de demande de congés.

Ce dernier se déroulera de la façon suivante :

1. Un employé dépose une demande de congés
2. Le responsable de l'employé décide ou non d'approuver la demande

Afin de réaliser cela, nous allons emprunter une approche itérative alternant contenu théorique et exercices pratiques. Chaque exercice permettra d'obtenir une version exécutable du processus de plus en plus complète.

Chaque exercice est découpé en 3 sections :

- **Objectif** - Présente les concepts clés visés par l'exercice.
- **Instructions résumées** - Contient les grandes étapes de la réalisation de l'exercice.
- **Instructions pas à pas** - Fournit une description détaillée pas à pas des étapes permettant de réaliser l'exercice.

---

# Chapitre 1. Exercice : Modélisation basique de processus

## Objectif

L'objectif de cet exercice est de créer une première version "basique" du diagramme de processus de demande de congés.

A ce stade, le processus est exécutable mais ne présente pas encore d'intérêt métier car il ne dispose pas de formulaires ni de données. Nous le compléterons dans les exercices suivants.

## Instructions résumées

Afin de réaliser l'exercice, créer un diagramme "Diagramme demande de congés" contenant un processus "Demande de congés" en version 1.0.

Le pool du processus devra contenir les éléments BPMN suivants :

- Un événement de début "Saisir demande"
- Une tâche humaine "Valider demande"
- Un événement de terminaison "Fin - Demande traitée"

## Instructions pas à pas

1. **Démarrer le Studio Bonita.**
2. **Créer un nouveau diagramme de processus.**
3. **Spécifier le nom du diagramme et celui du processus.**

Cliquer sur le menu "Diagramme / Renommer" en haut du Studio.

Entrer "Diagramme demande de congés" pour le nom du diagramme et "Demande de congés" pour celui du pool.

4. **Renommer l'événement de démarrage en "Saisir demande".**

Sélectionner l'événement de démarrage présent dans le diagramme.

Naviguer dans l'onglet "Général / Général" puis entrer le nouveau nom.

5. **Renommer la tâche humaine en "Valider demande"**

6. **Ajouter un événement de terminaison nommé "Fin - Demande traitée"**

Sélectionner la tâche humaine, faire un glisser-déplacer à partir de l'icône contextuel d'événement (A) et Sélectionner l'événement de terminaison (B).

Une fois l'évènement créé, le renommer.

7. **Vérifier que le diagramme ressemble à ceci :**

**8. Enregistrer le processus.**

Cliquer sur le bouton "Enregistrer" de la barre de menu supérieur du Studio.

**9. Exécuter le processus pour le tester.**

Cliquer sur le bouton de la barre de menu supérieur du Studio.

Votre navigateur devrait s'ouvrir et présenter un formulaire d'instanciation automatiquement généré. Soumettre le formulaire.

Vous devriez être redirigé vers la boîte de réception de tâches du Portail Bonita. Sélectionner la tâche "Valider demande" et cliquer sur "Faire" en haut à droite de l'écran.

Soumettre le formulaire de l'étape "Valider demande" pour terminer le cas.

---

# Chapitre 2. Exercice : Ajout de données et spécification de contrats

## Objectif

L'objectif de cet exercice est de continuer l'implémentation du processus de demande de congés en :

1. Configurant un modèle de données métier communément appelé BDM (Business Data Model)
2. Spécifiant des contrats pour l'instanciation du processus et l'exécution des étapes humaines

Une fois complété, le processus sera exécutable avec des formulaires générés automatiquement.

### Avertissement

Il est important de strictement respecter la casse et la syntaxe des noms techniques fournis dans les instructions sous peine d'erreurs.

## Instructions résumées

Dupliquer le diagramme de processus de l'exercice précédent pour créer une version 2.0.

Créer un objet BDM "DemandeConges" (il s'agit d'un nom technique, il faut donc omettre les espaces, accents et autres caractères spéciaux) avec les attributs suivants :

Nom	Type	Obligatoire
idDemandeur	Long	Non
dateDebut	Date	Oui
nombreJours	Entier	Oui
estApprouvee	Booléen	Non

Déclarer une variable métier "demande" de type "DemandeConges" sur le pool.

Générer un contrat d'instanciation en utilisant la variable métier "demande" et en incluant donc les éléments suivants :

- dateDebut
- nombreJours

Ajouter les deux contraintes suivantes sur le contrat d'instanciation :

- "dateDebut" doit être dans le futur
- "nombreJours" doit être strictement supérieur à zéro

Initialiser le BDM "demande" avec le code suivant:

```
import org.bonitasoft.engine.api.ProcessAPI;
import org.bonitasoft.engine.bpm.process.ProcessInstance;
import com.company.model.DemandeConges;

// Récupérer l'instance de processus en cours
ProcessAPI processAPI = apiAccessor.getProcessAPI();
ProcessInstance processInstance = processAPI.getProcessInstance(processInstanceName);
```

```
// Initialiser nouvelle demande
DemandeConges nouvelleDemande = new DemandeConges();
nouvelleDemande.setIdDemandeur(processInstance.getStartedBy());
nouvelleDemande.setDateDebut(contratDemande.get("dateDebut"));
nouvelleDemande.setNombreJours(contratDemande.get("nombreJours"));
return nouvelleDemande;
```

Ajouter un contrat sur l'étape "Valider demande" avec les éléments suivants:

- estApprouvee (Booléen)

Ajouter une opération sur l'étape "Valider demande" pour affecter la valeur retournée par le contrat à l'attribut "estApprouvee" du BDM "demande".

## Instructions pas à pas

### 1. Dupliquer le diagramme de processus existant pour créer une version 2.0.

Dans le menu supérieur du Studio, cliquer "Diagramme / Dupliquer".

Mettre à jour les numéros de version du diagramme ET du processus.

### 2. Créer le BDM "DemandeConges".

Naviguer dans le menu "Développement / Modèle de Données Métier / Gérer".

Cliquer sur "Ajouter" (A), dans la liste d'objets métiers et nommer l'objet "DemandeConges" (B) (il s'agit d'un nom technique, il faut donc omettre les espaces, accents et autres caractères spéciaux).

Avec l'objet "DemandeConges" sélectionné, ajouter les attributs suivants (C):

Nom	Type	Obligatoire
idDemandeur	Long	Non
dateDebut	Date	Oui
nombreJours	Entier	Oui
estApprouvee	Booléen	Non

Cliquer sur "Terminer".

### 3. Déclarer une instance du BDM "DemandeConges" dans le processus.

Sélectionner le pool du processus et naviguer dans l'onglet "Données / Variables du pool".

Cliquer sur "Ajouter" dans la section "Variables métier".

Nommer la variable "demande" et sélectionner l'objet métier "DemandeConges".

Cliquer sur "Terminer".

### 4. Configurer le contrat d'instanciation

Sélectionner le pool de processus et naviguer dans l'onglet "Exécution / Contrat / Entrées".

Cliquer sur "Ajouter à partir données".

Sélectionner le BDM "demande" et cliquer sur "Suivant".



Décocher les attributs "idDemandeur" et "estApprouvee".

Click sur "Terminer" et ignorer le message d'avertissement concernant l'initialisation incomplète de la variable métier.

Toujours dans l'onglet "Exécution / Contrat", basculer dans l'onglet "Contraintes".

Ajouter la contrainte suivante :

Propriété	Valeur
Nom	dateDebutDansFutur
Expression	return contratDemande.dateDebut.compareTo(new Date()) > 0;
Message d'erreur	La date de début doit être dans le futur

Ajouter une seconde contrainte:

Propriété	Valeur
Nom	auMoinsUnJour
Expression	return contratDemande.nombreJours > 0;
Message d'erreur	Le nombre de jours doit être supérieur à zéro

##### 5. Initialiser le BDM de la demande à partir du contrat d'instanciation.

Sélectionner le pool du processus et naviguer dans l'onglet "Données / Variables du pool".

Sélectionner la variable "demande" et cliquer sur "Modifier".

Cliquer sur l'icône "crayon" associé au champ "Valeur par défaut" pour ouvrir l'éditeur d'expression.

S'assurer que le type d'expression sélectionné est "Script".

Nommer le script "initDemande".

Entrer le code suivant:

```
import org.bonitasoft.engine.api.ProcessAPI;
import org.bonitasoft.engine.bpm.process.ProcessInstance;
import com.company.model.DemandeConges;

// Récupérer l'instance de processus en cours
ProcessAPI processAPI = apiAccessor.getProcessAPI();
ProcessInstance processInstance = processAPI.getProcessInstance(processInstance.getId());

// Initialiser nouvelle demande
DemandeConges nouvelleDemande = new DemandeConges();
nouvelleDemande.setIdDemandeur(processInstance.getStartedBy());
nouvelleDemande.setDateDebut(contratDemande.get("dateDebut"));
nouvelleDemande.setNombreJours(contratDemande.get("nombreJours"));
return nouvelleDemande;
```

---

##### 6. Configurer le contrat de l'étape "Valider demande".

Sélectionner l'étape "Valider demande" et naviguer dans l'onglet "Exécution / Contrat / Entrées".

Ajouter une entrée "estApprouvee" de type "BOOLEAN".

**7. Ajouter une opération sur la tâche "Valider demande" pour mettre à jour la demande.**

Sélectionner l'étape "Valider demande" et naviguer dans l'onglet "Exécution / Opérations".

Cliquer sur "Ajouter".

Sélectionner "demande" comme cible.

Cliquer sur le lien "Prend la valeur de" et changer l'opérateur en "Utiliser une méthode Java".

Sélectionner la méthode "setEstApprouvee" et cliquer sur "OK".

Sélectionner l'entrée de contrat "estApprouvee" dans le menu déroulant de droite.

**8. Enregistrer le processus.**

Cliquer sur le bouton "Enregistrer" de la barre de menu supérieur du Studio.

**9. Exécuter le processus et tester les contraintes des contrats.**

Lancer le processus et remplir les formulaires automatiquement générés.

**Astuces:**

- pour le champ "dateDebut", le format de date attendu est AAAA-MM-JJ.
- pour le champ "estApprouvee", une valeur booléenne est attendue: soit "true" ou "false" (tout autre valeur sera évaluée comme "false").

---

# Chapitre 3. Exercice : Contrôle dynamique du flux d'exécution grâce aux données

## Objectif

L'objectif de cet exercice est d'enrichir le processus existant en ajoutant la gestion dynamique de la validation de la demande de congés. Ceci sera réalisé à l'aide d'une porte logique exclusive donnant accès à deux flux d'exécutions permettant de gérer le refus ou l'approbation de la demande.

## Instructions résumées

Dupliquer le diagramme de processus de l'exercice précédent pour créer une version 3.0.

Modifier le diagramme pour placer les éléments suivants à la suite de l'étape "Valider demande" :

- une porte exclusive "Demande approuvée ?"
- une tâche automatique "Notifier approbation"
- un événement de terminaison "Fin - Demande approuvée"
- une tâche automatique "Notifier refus"
- un événement de terminaison "Fin - Demande refusée"

Configurer les transitions en sortie de la porte exclusive pour diriger dynamiquement le flux d'exécution du processus en fonction des données du processus.

## Instructions pas à pas

**1. Dupliquer le diagramme de processus de l'exercice précédent pour créer une version 3.0.**

**2. Supprimer l'événement de terminaison.**

Sélectionner l'évènement dans le diagramme et appuyer sur la touche "Suppr" du clavier.

**3. Ajouter une porte exclusive "Demande approuvée ?" à la suite de la tâche "Valider demande".**

Sélectionner la tâche "Valider demande", effectuer un glisser-déplacer à partir de l'icône contextuel de porte logique.

Une fois la porte créée, utiliser l'icône contextuel "boîte à outils" (1) pour changer son type en "Exclusif" (2).

Renommer la porte en "Demande approuvée ?".

**4. Créer une tâche automatique "Notifier approbation" à partir de la porte logique.**

5. **Ajouter un événement de terminaison "Fin - Demande approuvée" à la suite de la tâche automatique "Notifier approbation".**

6. **Créer une tâche automatique "Notifier refus" à partir de la porte logique.**

7. **Ajouter un événement de terminaison "Fin - Demande refusée" à la suite de la tâche automatique "Notifier refus".**

8. **Nommer les 2 transitions entre la porte logique et les étapes automatiques**

La transition menant à l'étape "Notifier approbation" se nommera "Oui" et l'autre "Non".

9. **Configurer la condition sur la transition "Oui" à partir de l'attribut "estApprouvee" du BDM "demande".**

Sélectionner la transition et cliquer sur l'icône crayon à côté du champ "Condition" pour ouvrir l'éditeur d'expression.

Dans l'éditeur d'expression, sélectionner "Java" comme type d'expression, sélectionner la variable "demande" et la méthode "isEstApprouvee".

10. **Configurer la transition "Non" en tant que flux par défaut.**

Sélectionner la transition "Non" et cocher la case "Flux par défaut".

11. **Déclencher la validation du diagramme.**

Maintenant que les conditions sont correctement configurées sur les transitions il est possible de relancer la validation. Naviguer dans l'onglet "Statut de validation" et cliquer sur "Rafraîchir".

12. **Vérifier que le diagramme ressemble à ceci :**

13. **Exécuter deux fois le processus pour tester les différents chemins.**

Attention : ne pas cliquer une deuxième fois sur le bouton "Lancer", cela aurait pour effet de redéployer le processus et donc de supprimer les instances existantes. Pour créer la seconde instance, il faut naviguer dans le portail.

Ouvrir le portail à l'aide de l'icône dans la barre d'outils supérieure du Studio ou à l'aide du lien dans l'interface web :

Ensuite, naviguer dans le menu "Processus" puis sélectionner la définition de processus "Demande de congés" en version 3.0 et cliquez sur "démarrer".

14. **Vérifier le bon déroulement du processus dans l'interface d'administration du portail**

Une fois dans le portail, changer de profil dans le coin supérieur droit pour passer en "Administrateur".

Naviguer dans le menu "BPM / Cas" (A) puis sélectionner l'onglet "Cas archivés" (B) et cliquer sur le lien "..." dans la colonne action (C) pour visualiser le détail d'un cas.

Vérifier les "Tâches terminées" pour valider le bon déroulement du processus :



---

# Chapitre 4. Exercice : Création de formulaires

## Objectif

L'objectif de cet exercice est de fournir des formulaires conviviaux et adaptés pour l'exécution du processus.

## Instructions résumées

Dupliquer le diagramme de processus de l'exercice précédent pour créer une version 4.0.

Créer les formulaires suivants :

- Un formulaire d'instanciation de processus au niveau du pool permettant d'alimenter les données "dateDebut" et "nombreJours".
- Un formulaire d'étape permettant d'afficher les données "dateDebut" et "nombreJours" en lecture seule puis d'alimenter la donnée "estApprouvee".

## Instructions pas à pas

1. **Dupliquer le diagramme de processus de l'exercice précédent pour créer une version 4.0.**

2. **Créer un formulaire d'instanciation.**

Sélectionner le pool du processus et naviguer dans l'onglet "Exécution / Formulaire d'instanciation".

Utiliser le menu déroulant "Formulaire cible" pour créer un nouveau formulaire.

Ceci aura pour effet d'ouvrir le UI Designer dans votre navigateur avec formulaire généré automatiquement à partir du contrat d'instanciation.

3. **Renommer le formulaire**

Utiliser le champ texte en haut de l'écran pour renommer le formulaire en "remplirDemandeConges" (il s'agit d'un nom technique, il faut donc omettre les espaces, accents et autres caractères spéciaux).

Cliquer sur enregistrer.

4. **Ajouter un conteneur de formulaire**

Depuis la palette de widget située sur la gauche, faire glisser un widget de type "Form container" dans une nouvelle ligne en dessous du widget "Contrat demande".

**Astuce :** Une zone en pointillé permettant d'insérer une nouvelle ligne apparaît lorsque l'on survole la séparation entre deux widget.

Le conteneur de formulaire permet de fournir une aide à la validation du contenu du formulaire (champs obligatoires et valeur des champs).

**5. Déplacer les widgets existants dans le conteneur de formulaire**

Sélectionner le widget "Date Debut" et le faire glisser dans le conteneur de formulaire.

Répéter l'opération pour les widgets "Nombre Jours" et "Submit" en créant de nouvelles lignes.

Une fois les widgets regroupés sous le conteneur de formulaire, sélectionner et supprimer les lignes vides.

**Astuce :** Survoler la barre grise se situant à gauche d'une ligne (A) pour faire apparaître les actions dont la suppression (B).

**6. Changer le titre du formulaire**

Sélectionner le widget "Contrat demande" en haut du formulaire.

Dans la section à droite de l'écran, éditer les propriétés suivantes:

Propriété	Valeur
Texte	Saisir une demande de congés
Niveau du titre	Niveau 2
Alignement	au centre

**7. Changer le widget "Date Debut"**

Sélectionner le widget "Date debut" et éditer les propriétés suivantes:

Propriété	Valeur
Obligatoire	oui
Libellé	Date de début
Placeholder	Date de début des congés
Format de la date	yyyy-MM-dd

**8. Changer le widget "Nombre Jours"**

Sélectionner le widget "Nombre Jours" et éditer les propriétés suivantes:

Propriété	Valeur
Obligatoire	oui
Libellé	Nombre de jours
Placeholder	Nombre de jours de congés
Valeur minimum *	1

\* à ne pas confondre avec la propriété "Longueur minimum de la valeur".

**9. Supprimer les valeurs initiales des champs (cela permettra d'afficher les placeholders)**

Dans la partie inférieure de l'écran, sous l'onglet "variables", cliquer sur l'icône crayon situé à droite de la ligne contenant la variable "formInput".

Remplacer la valeur du JSON par ceci:

```
{
```

```
"contratDemande" : {  
  "dateDebut" : null,  
  "nombreJours" : null  
}
```

#### 10. Déclarer une variable pour gérer les potentielles erreurs de soumission du formulaire

Dans la partie inférieure de l'écran, sous l'onglet "variables", cliquer sur "Créer une nouvelle variable".

Nommer la variable "erreur", laisser son type à "String" et sa valeur vide puis enregistrer.

#### 11. Changer le widget "Submit"

Sélectionner le widget "Submit" et éditer les propriétés suivantes:

Propriété	Valeur
Libellé	Soumettre
Valeur de réponse en cas d'échec	erreur

Le fait d'avoir affecté la variable "erreur" à la réponse en cas d'échec nous permettra de récupérer le détail de l'erreur.

#### 12. Afficher dynamiquement l'erreur

Depuis la palette de widget située sur la gauche, faire glisser un widget de type "Text" en dessous du bouton "Soumettre".

Cliquer sur l'icône "binding" associé à la propriété "Hidden".

Entrer ce contenu dans le champ texte qui est apparu (ceci aura pour effet d'afficher le widget uniquement en cas d'erreur):

```
erreur == null
```

Remplacer la propriété "Texte" du widget par ce contenu:

```
<b>Erreur :</b> {{erreur | json}}
```

#### 13. Vérifier le formulaire

Vérifier que le formulaire ressemble à ceci dans l'éditeur:

Cliquer sur le bouton "Aperçu" en haut de l'écran.

Vérifier que l'aperçu ressemble à ceci:

On notera les points suivants dans l'aperçu:

- le widget présentant l'erreur n'est pas visible
- le bouton "Soumettre" est désactivé par défaut (ceci est dû à la validation fournie par le conteneur de formulaire)

Utiliser l'aperçu pour tester la validation des champs.



Un fois le formulaire valide, le bouton "Soumettre" sera activé.

**Note :** il n'est toutefois pas possible de soumettre le formulaire depuis l'aperçu même lorsque ce dernier est valide.

#### 14. Enregistrer le formulaire

Utiliser le bouton "Enregistrer" en haut de l'écran pour enregistrer le formulaire.

#### 15. Créer un formulaire pour l'étape "Valider demande".

Dans le Studio, sélectionner l'étape "Valider demande" et naviguer dans l'onglet "Exécution / Formulaire".

Utiliser le menu déroulant "Formulaire cible" pour créer un nouveau formulaire.

#### 16. Renommer le formulaire

Depuis le UI Designer, renommer le formulaire en "validerDemandeConges" et l'enregistrer.

#### 17. Récupérer les données métier de la demande en cours

Dans la section "variables" en bas de l'écran, cliquer sur le bouton "Créer une nouvelle variable".

Configurer la variable avec les propriétés suivantes:

Propriété	Valeur
Nom	demande
Type	External API
URL d'API	../{ context.demande_ref.link }

**Note :** nous utilisons ici, un appel aux APIs REST de Bonita BPM pour récupérer notre BDM "demande". Nous bénéficions d'une variable "context" qui expose des liens vers les données du processus dont notre demande sous la forme "demande\_ref". Nous pouvons ainsi charger la demande grâce à l'attribut "link" qui donne l'URL à appeler pour récupérer l'objet.

#### 18. Ajouter une variable pour la gestion d'erreur

Créer une nouvelle variable, la nommer "erreur", laisser son type à "String" et sa valeur vide puis enregistrer.

#### 19. Supprimer les variables "formInput" et "formOutput"

Utiliser l'icône "poubelle" pour supprimer les variables "formInput" et "formOutput".

#### 20. Supprimer la case à cocher "Est Approuvée"

Sélectionner la case à cocher "Est Approuvée" et utiliser le contrôle de suppression qui apparaît lorsque l'on survole le widget avec le curseur.

#### 21. Supprimer le bouton "Submit"

#### 22. Ajouter un titre au formulaire

Faire glisser un widget de type "Titre" depuis la palette vers l'espace inoccupé en haut du formulaire.

Configurer le widget de la manière suivante:

Propriété	Valeur
Texte	Valider une demande de congés
Niveau du titre	Niveau 2
Alignement	au centre

**23. Ajouter un widget pour afficher la date de début**

Faire glisser un widget de type "Input" dans une nouvelle ligne sous le titre du formulaire.

Configurer le widget comme ceci:

Propriété	Valeur
Lecture seule	oui
Libellé	Date de début
Valeur	demande.dateDebut   date

**Note :** nous utilisons ici un filtre "| date" dans la valeur pour formater la date de façon lisible.

**24. Ajouter un widget pour afficher le nombre de jours**

Faire glisser un widget de type "Input" dans une nouvelle ligne sous le widget "Date de début".

Configurer le widget comme ceci:

Propriété	Valeur
Lecture seule	oui
Libellé	Nombre de jours
Valeur	demande.nombreJours

**25. Ajouter un widget pour refuser la demande**

Faire glisser un widget de type "Button" en bas du formulaire.

Configurer le widget comme ceci:

Propriété	Valeur
Width	6
Libellé	Refuser
Alignement	à droite
Style	danger
Données envoyées au clic	{"estApprouvee" : false}
Valeur de réponse en cas d'échec	erreur
URL cible en cas de succès	/bonita

**26. Ajouter un widget pour approuver la demande**

Faire glisser un widget de type "Button" dans la zone à droite du widget "Refuser".

Configurer le widget comme ceci:

Propriété	Valeur
Libellé	Approuver

Propriété	Valeur
Style	success
Données envoyées au clic	{"estApprouvee" : true}
Valeur de réponse en cas d'échec	erreur
URL cible en cas de succès	/bonita

**27. Afficher dynamiquement les potentielles erreurs de soumission**

Ajouter un widget de type "Text" en bas du formulaire.

Cliquer sur l'icône "binding" associé à la propriété "Hidden".

Entrer ce contenu dans le champ texte qui est apparu:

```
erreur == null
```

Remplacer la propriété "Texte" du widget par ce contenu:

```
<b>Erreur :</b> {{erreur | json}}
```

**28. Vérifier le formulaire**

S'assurer que le formulaire ressemble à ceci:

**29. Enregistrer le formulaire**

Utiliser le bouton "Enregistrer" en haut au centre de l'écran pour enregistrer le formulaire.

**30. Tester l'exécution du processus**

Lancer l'exécution du processus par le biais du Studio.

Valider à partir de l'historique des cas que le bon chemin a bien été emprunté jusqu'à la fin du processus.

---

# Chapitre 5. Exercice : Gestion d'acteurs

## Objectif

L'objectif de cet exercice est d'apporter une notion de collaboration au processus existant en distribuant les formulaires entre deux acteurs : le demandeur et le validateur.

## Instructions résumées

Dupliquer le diagramme de processus de l'exercice précédent pour créer une version 5.0.

Ajouter une lane "Validateur" au diagramme et y déplacer la tâche "Valider demande".

Configurer un filtre d'acteur de type "Manager de l'initiateur du processus" sur la lane "Validateur".

## Instructions pas à pas

1. **Dupliquer le diagramme de processus de l'exercice précédent pour créer une version 5.0.**

2. **Ajouter une lane "Validateur" au diagramme**

Sélectionner une lane à partir de la palette d'éléments BPMN présente sur le côté gauche du Studio puis cliquer dans le pool du processus.

Sélectionner la lane, naviguer dans l'onglet "Général / Lane" puis modifier le nom en "Validateur".

3. **Renommer l'autre lane en "Demandeur".**

4. **Faire glisser la tâche "Valider demande" dans la lane "Validateur".**

5. **Configurer un filtre d'acteur de type "Manager de l'initiateur du processus" sur la lane "Validateur".**

Sélectionner la lane "Validateur", naviguer dans l'onglet "Général / Acteur".

Sélectionner l'acteur "Employee actor" à partir du menu déroulant.

Cliquer sur le bouton "Définir" associé au filtre.

Sélectionner un filtre de type "Manager de l'initiateur du processus" puis cliquer sur "Suivant".

Nommer le filtre "managerInitiateur" et cliquer sur "Terminer".

6. **Vérifier que le diagramme ressemble à ceci :**

7. **Exécuter le processus avec les deux acteurs**

Lancer le processus à partir du Studio (l'utilisateur Walter Bates sera utilisé).

Soumettre le formulaire de "Saisie demande congés". Si les acteurs sont correctement configurés, la tâche "Valider demande de congés" ne devrait pas être proposée.

Se déconnecter du portail en naviguant sur le nom d'utilisateur dans le coin supérieur droit puis "Déconnexion".

Se connecter avec l'utilisateur "helen.kelly" et le mot de passe "bpm".

Si le filtre d'acteur s'est bien exécuté, la tâche "Valider demande de congés" devrait être disponible dans la liste des tâches à faire.

---

# Chapitre 6. Exercice : Notification par le biais d'un connecteur

## Objectif

L'objectif de cet exercice est de traiter une interaction entre le processus et un système externe par le biais d'un connecteur. Nous nous intéresserons ici à la notification du résultat de la demande de congés par le biais d'un connecteur d'envoi d'email.

## Instructions résumées

Dupliquer le diagramme de processus de l'exercice précédent pour créer une version 6.0.

Ajouter un connecteur d'envoi d'email sur les tâches automatiques "Notifier approbation" et "Notifier refus". Ceux-ci enverront un email au demandeur avec le statut de validation de sa demande.

Le code ci-dessous sera utilisé pour récupérer l'email du demandeur dans le connecteur:

```
import org.bonitasoft.engine.api.IdentityAPI;
import org.bonitasoft.engine.api.ProcessAPI;
import org.bonitasoft.engine.bpm.process.ProcessInstance;
import org.bonitasoft.engine.identity.ContactData;

// Récupération APIs
ProcessAPI processAPI = apiAccessor.getProcessAPI();
IdentityAPI identityAPI = apiAccessor.getIdentityAPI();
// Récupération initiateur du processus
ProcessInstance processInstance = processAPI.getProcessInstance(processInstanceName);
long initiatorUserId = processInstance.getStartedBy();
// Récupération email de l'initiateur
ContactData contactData = identityAPI.getUserContactData(initiatorUserId, false);
return contactData.getEmail();
```

## Instructions pas à pas

1. **Dupliquer le diagramme de processus de l'exercice précédent pour créer une version 6.0.**

2. **Tester le connecteur d'envoi d'email pour obtenir la bonne configuration SMTP.**

Naviguer dans le menu "Développement / Connecteurs / Tester un connecteur" de la barre supérieure du Studio.

Sélectionner un connecteur de type "Courriel" à partir du filtre ou depuis la catégorie "Communications".

Remplir les paramètres de connexion de votre compte email et passer à la page suivante.

Entrer votre email en tant que destinataire et expéditeur et passer à la page suivante.

Entrer "Bonita test" en sujet, cliquer le bouton "Test" et confirmer sans cocher de dépendance.

Un message similaire à celui ci-dessous devrait s'afficher:

S'assurer que l'email est bien reçu en allant voir votre boîte réception (cela peut prendre quelques minutes). Si ce n'est pas le cas, mettre à jour la configuration puis re-tester.

Une fois la configuration validée, cliquer sur .

Nommer la configuration "configEmail" et la sauvegarder.

Fermer l'interface de test du connecteur.

### 3. Ajouter un connecteur d'envoi d'email sur la tâche "Notifier approbation".

Sélectionner la tâche "Notifier approbation", naviguer dans l'onglet "Exécution / Connecteurs en entrée" puis cliquer sur "Ajouter".

Sélectionner un connecteur de type "Courriel".

Spécifier "envoiEmailApprobation" comme nom et passer à la page suivante.

Ne pas remplir les paramètres de la page mais cliquer sur .

Sélectionner la configuration "configEmail" et passer à la page de configuration du destinataire.

Entrer votre adresse email dans le champ expéditeur "De".

Utiliser l'icône "crayon" pour éditer l'expression sur le champ destinataire "A".

Changer le type d'expression à "Script", nommer le script "recupEmailDemandeur" et coller le code ci-dessous dans la zone d'édition du script:

```
import org.bonitasoft.engine.api.IdentityAPI;
import org.bonitasoft.engine.api.ProcessAPI;
import org.bonitasoft.engine.bpm.process.ProcessInstance;
import org.bonitasoft.engine.identity.ContactData;

// Récupération APIs
ProcessAPI processAPI = apiAccessor.getProcessAPI();
IdentityAPI identityAPI = apiAccessor.getIdentityAPI();
// Récupération initiateur du processus
ProcessInstance processInstance = processAPI.getProcessInstance(processInstance.getId());
long initiatorUserId = processInstance.getStartedBy();
// Récupération email de l'initiateur
ContactData contactData = identityAPI.getUserContactData(initiatorUserId, false);
return contactData.getEmail();
```

Passer à la page suivante et spécifier "Demande de congés approuvée" comme sujet.

Cliquer sur "Terminer".

### 4. Ajouter un connecteur d'envoi d'email sur la tâche "Notifier refus".

Répéter l'étape précédente en nommant le connecteur "envoiEmailRefus" et en spécifiant "Demande de congés refusée" comme sujet.

### 5. Mettre à jour l'email de l'utilisateur de test "Walter Bates"

Ouvrir le Portail depuis le Studio grâce au bouton de la barre d'outils.

Passer en profil "Administrateur" puis naviguer dans le menu "Organisation / Utilisateurs".

Sélectionner l'utilisateur "Walter Bates" et cliquer sur le bouton "Détails".

Cliquer sur le bouton "Mise à jour de l'utilisateur".

Sélectionner l'onglet "Carte de visite"

Entrer votre email dans le champ "Email".

Sauvegarder le changement et fermer le Portail.

6. **Exécuter deux fois le processus pour tester les différents chemins et s'assurer que les emails sont bien envoyés.**