

# ARCHITECTURE ORIENTE SERVICES

Exercices d'applications

Dr Salma SASSI  
Maitre-assistant  
FSJEG Jendouba

salma.sassi@fsjegj.rnu.tn  
sassisalma@yahoo.fr

# Chapitre 2 : Cartographie et architecture

**Objectif** : Ces exercices ont pour objectif de vous familiariser avec les techniques de modélisation d'architecture.

**La démarche proposée est la suivante :**

1. Cartographie applicative
2. Architecture n-tiers
3. Qualification des flux

## Cartographie applicative

On s'intéresse au SI de l'usine TartEnPion, basée en Normandie, qui fabrique des tartes surgelées pour le compte d'une grande entreprise de l'industrie agroalimentaire appelée Miam! dont le siège se trouve à Paris.

*Remarque : pour des raisons pédagogiques, cette description est volontairement très simplifiée et très incomplète, en particulier beaucoup d'éléments importants nécessaires au fonctionnement d'une usine classique n'y figurent pas (gestion des coûts, gestion des ressources humaines, suivi financier, suivi des processus, maintenance des lignes de production...).*

### Travail demandé

- Identifier les grands blocs applicatifs du SI de cette usine
- Représenter chaque application par un rectangle.
- Répertorier les informations échangées par ces grands blocs applicatifs  
Représenter chaque information échangée par une flèche allant de l'application qui produit l'information à l'application qui consomme l'information, et indiquer sur cette flèche l'information échangée.

## Architectures n-tiers

Cet exercice concerne uniquement l'application VENTES (centralisée au siège de Miam!) et l'application GSTOCKS (localisée au sein de l'usine TartEnPion). L'application VENTES est une application 4-tiers qui est utilisée par l'ensemble du pôle commercial via une interface web accessible uniquement dans l'enceinte du siège social. Les commerciaux en déplacement auprès des distributeurs accèdent à l'application via un client Android. L'application GSTOCKS s'appuie sur une base de données Oracle pour gérer les stocks et est accédée par les employés de l'usine via un client VB.NET. A partir de l'application VENTES, il est possible d'interroger l'application GSTOCKS pour connaître la date de disponibilité d'un produit donné (en fonction du stock ou de la production).

### Travail demandé

- Faire un schéma d'architecture montrant uniquement ces deux applications en faisant apparaître leurs différents tiers.

## Qualification des flux

L'entreprise X a fait réaliser une cartographie applicative de son SI. Une première étude des flux a permis de recenser partiellement direction, format, protocole, nature, fréquence et volumétrie de ces flux.

## Travail demandé

Identifier pour les différents flux recensés s'ils sont :

- intra-application ou inter-applications ?
- AtoA ou BtoB ?
- unitaire ou de masse ?
- fil de l'eau ou cadencé ?
- synchrone ou asynchrone ?
- requête-réponse ou publication-abonnement ?

Remarque : pour certains flux, les informations disponibles peuvent ne pas permettre de déterminer certaines caractéristiques...

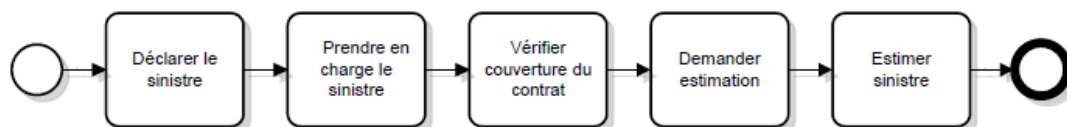
# Chapitre 3 : Le modèle SOA

**Objectif** : ces exercices ont pour objectif de vous familiariser avec les processus métier et les services

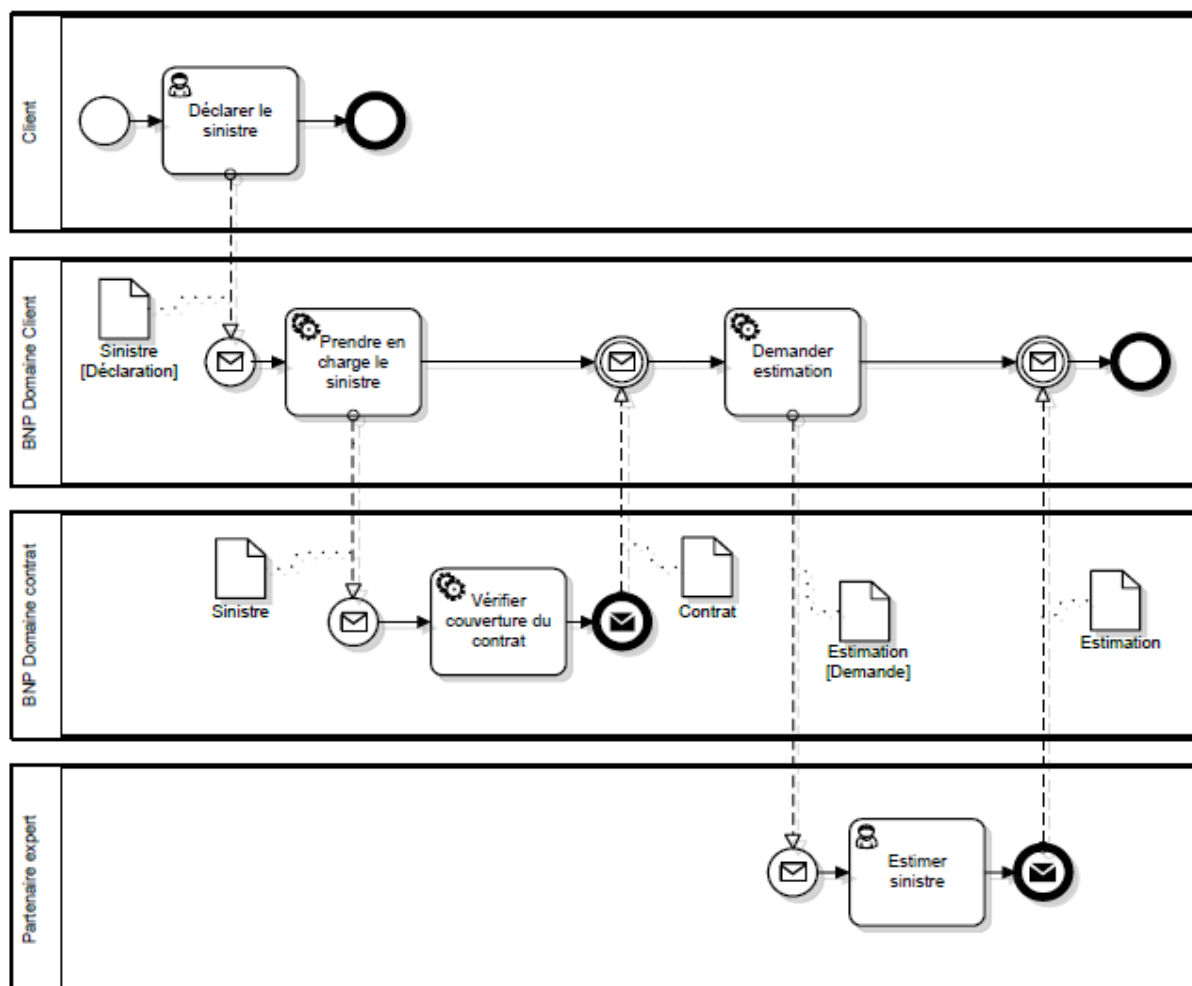
La démarche proposée est la suivante :

1. Formalisation des processus fonctionnels
2. Déclinaison du processus sur les domaines fonctionnels
3. Identification des services

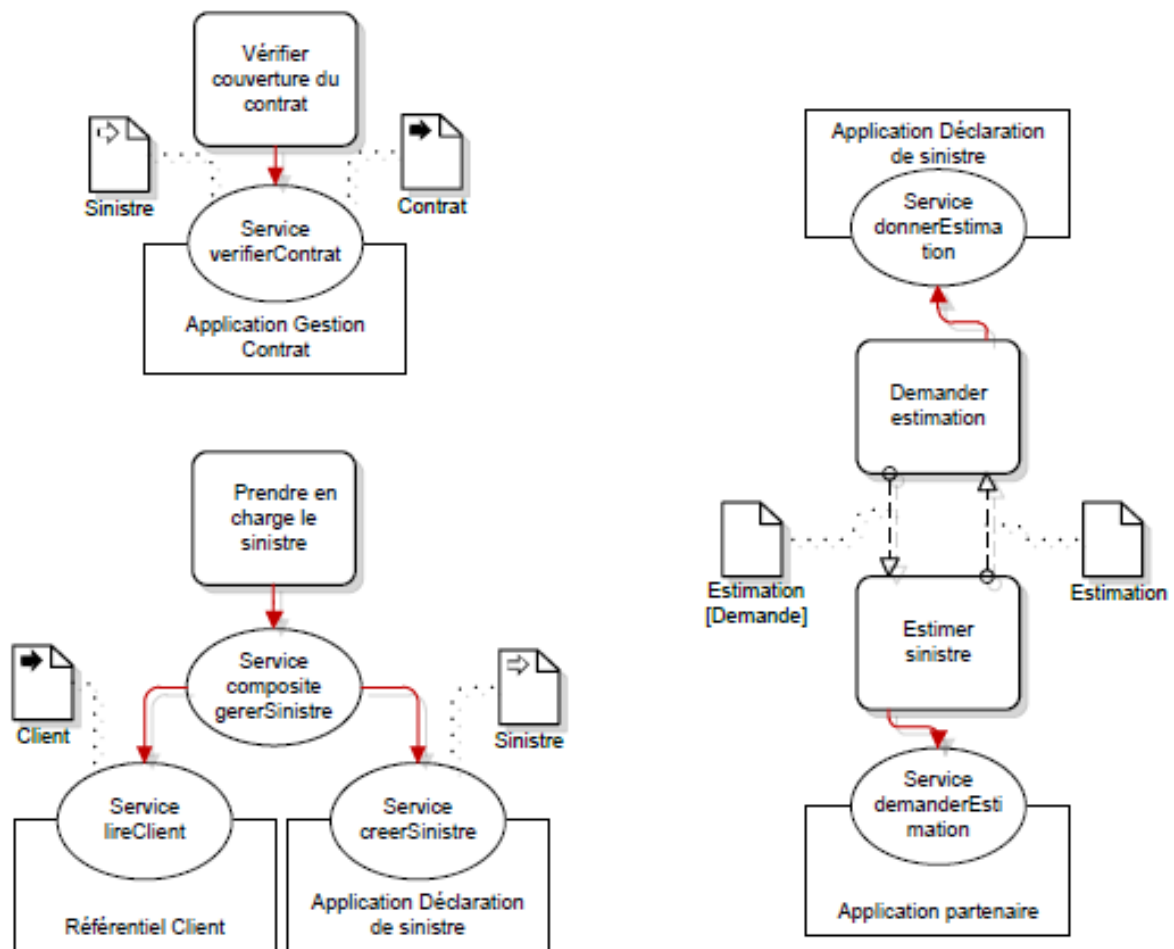
## Étape 0 : Formalisation des processus fonctionnels



## Étape 1 : Décliner le processus sur les domaines fonctionnels du SI



## Étape 2 : Identification des services



## Travail demandé

- Expliquer le processus métier et les services métiers

# Chapitre 4 : Modélisation des processus

**Objectif** : ces exercices ont pour objectif de vous familiariser avec la modélisation de processus (BPM - Business Process Modeling) de manière concrète à travers une série d'exercices progressifs.

La démarche proposée est la suivante :

1. Prise en main de l'environnement de modélisation
2. Gestion des congés
3. Gestion des notes de frais

Pour tous les exercices, nous utiliserons la notation BPMN 2.0.

## Prise en main de l'environnement de modélisation

### Outil de modélisation

Nous allons utiliser la version d'essai d'un outil de modélisation collaboratif en ligne appelé **Signavio**.

Lancer un navigateur web et aller à l'URL suivante : <https://www.signavio.com/fr/>

Remplir le formulaire d'inscription en indiquant une adresse email, vous recevrez un email avec un lien vous permettant de vous connecter.

### Configuration de l'espace de travail

Une fois logués sur Signavio, aller dans le dossier "Shared Documents" et créer un répertoire à votre nom. Par la suite, vous enregistrerez tous vos modèles dans ce répertoire.

### Documentation

Vous trouvez avec ce manuel quelques documents essentiels pour bien utiliser la notation BPMN 2.0 :

- Un **extrait de la spécification de BPMN 2.0** expliquant et illustrant la signification des différents éléments syntaxiques.
- Un **poster synthétique** de présentation de l'ensemble de la notation en *anglais* et en *français*

## Gestion des congés

Considérons une entreprise qui souhaite informatiser la gestion des demandes de congés. La procédure existante est entièrement manuelle et basée sur un formulaire papier, qui doit circuler entre les différentes personnes impliquées. Dans le cadre d'une démarche SOA, l'objectif de l'entreprise est de :

1. Appuyer ce processus sur un moteur de workflow qui permettra une transmission informatique des demandes entre les différentes personnes impliquées ;
2. Automatiser les tâches du processus qui peuvent l'être en s'appuyant sur des services exposés au niveau du SI.

## **Description du processus**

Le processus de gestion des congés est le suivant :

1. Le salarié soumet une demande de congés : il choisit la date de début, le nombre de jours ouvrés à poser et propose une répartition des jours à poser en CP (congés payés) et RTT (réduction du temps de travail).
2. La secrétaire de l'entité est chargée de vérifier que les soldes courants de CP et RTT du salarié lui permettent de poser les congés souhaités :
  - a. Si les soldes du salarié permettent de poser les congés, la secrétaire transmet la demande au responsable de l'entité.
  - b. Sinon, la demande est retournée au salarié, qui doit la modifier.
3. Le responsable d'entité est chargé de valider les congés du salarié en fonction des contraintes opérationnelles :
  - a. Il peut arriver que les congés soient refusés par le responsable pour des raisons opérationnelles. Un refus clôt la demande de congés, sans pour autant empêcher le salarié de soumettre une nouvelle demande s'il le souhaite.
  - b. Si le responsable valide les congés, alors la demande est de nouveau transmise à la secrétaire, qui est chargée de modifier les soldes de congés du salarié.

En fin de processus, le salarié reçoit un récapitulatif des informations concernant sa demande.

## **Travail demandé : Modéliser le processus**

1. Faire un premier modèle « macro » des grandes étapes du processus.
2. Faire apparaître les différents chemins et issues possibles du processus.
3. Déterminer les différents acteurs impliqués dans le processus et les étapes qu'ils sont chargés de réaliser.
4. Expliciter les données qui doivent être échangées au cours du processus.

## **Travail demandé : Analyser le processus**

On s'intéresse maintenant à l'informatisation du processus.

1. Créer un document texte simple qui vous permettra de sauvegarder vos réponses aux questions au fur et à mesure des questions. Pour cela, vous pouvez utiliser NotePad par exemple (attention, pas de fichier Word !!).
2. Quelles tâches pourraient être informatisées ? Distinguer les tâches qui peuvent être réalisées par un humain en utilisant une application, des tâches qui peuvent être totalement automatisées (réalisées par un programme sans intervention humaine).
3. Sur quelles fonctionnalités applicatives exposées au niveau du SI faudrait-il appuyer le processus pour permettre cette informatisation ? Lister les fonctions avec les données en entrée et les données en sortie.
4. Conserver votre fichier de réponses pour l'exercice suivant.

## Gestion des notes de frais

Dans l'espace de travail de Signavio, accéder à la partie "Shared Documents" puis créer une copie du modèle "NoteDeFrais-Exercice" dans votre répertoire : dans le dossier "Shared Documents", sélectionner le fichier, puis aller dans le menu "Edit > Copy" et sélectionner le répertoire cible où copier le fichier.

### Travail demandé : Analyser le modèle

Répondre aux questions suivantes :

- a. Quels sont les différents participants intervenant dans ce processus ?
- b. Quand ce processus est-il déclenché ?
- c. Quelles sont les issues possibles de ce processus ?
- d. Quelle est la signification de l'icône suivante :

Dans ces exercices, vous avez pu découvrir le langage de modélisation de processus BPMN. Dans les différents exercices vous avez mis en œuvre les constructions les plus importantes de ce langage, qui permet de modéliser des processus très complexes. Ce langage est très largement utilisé dans le monde professionnel pour spécifier et documenter des processus métier. Dans le cours suivant, nous verrons comment obtenir un processus exécutable à partir d'un modèle.



# Chapitre 5 : Web Services WS-\* et RESTful

**Objectif** : ces exercices ont pour objectif de vous faire découvrir les Web Services de manière concrète en utilisant les technologies Java et Java Enterprise Edition.

**La démarche proposée est la suivante :**

1. Prise en main de l'environnement de développement
2. Développement d'un service WS-\*
3. Développement d'un service RESTful

## Prise en main de l'environnement de développement

L'environnement de développement utilisé dans ces exercices est composé de :

- NetBeans : un environnement de développement (IDE - Integrated Development Environment), qui permet notamment d'éditer et compiler du code ;
- GlassFish : un serveur d'application Java Enterprise Edition qui permet de déployer les applications développées.

Le but de ce premier exercice est de vous familiariser avec l'IDE NetBeans et avec le serveur d'applications GlassFish afin de vous permettre ensuite de développer et déployer des web services.

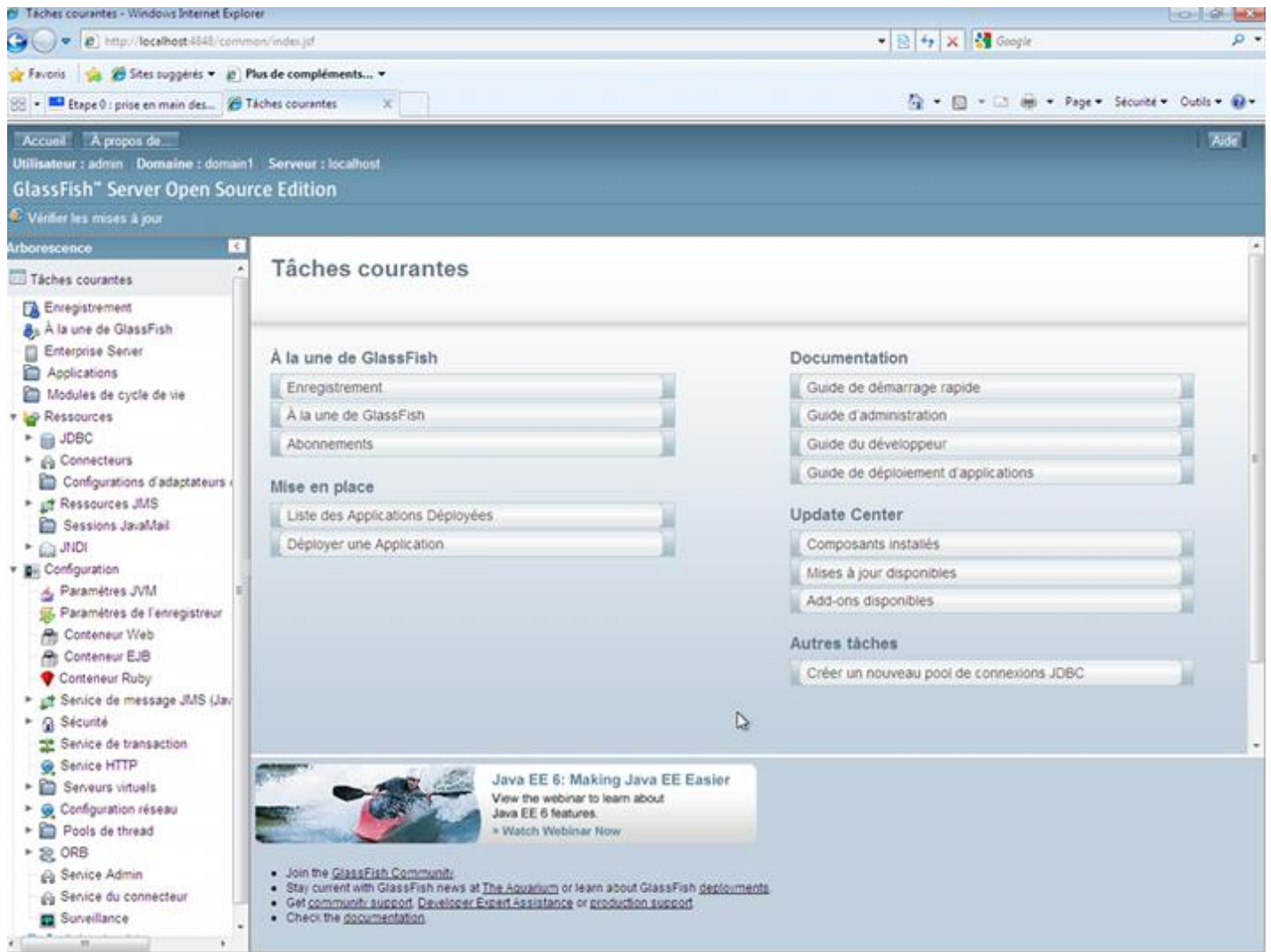
## Prise en main du serveur d'applications GlassFish

NetBeans dispose nativement d'une interface d'interaction avec le serveur d'application GlassFish. C'est cette interface que nous allons utiliser le plus souvent.

- ✓ Lancer NetBeans.
- ✓ Dans l'onglet Services, aller à Servers > GlassFish Server 3.1.2 puis choisir Start dans le menu contextuel (clic droit avec la souris).
- ✓ Regarder l'onglet GlassFish Server qui est apparu en bas de la fenêtre de Netbeans.  
A quoi sert-il ?
- ✓ Toujours sur Servers > GlassFish Server 3.1.2, choisir View Admin Console dans le menu contextuel.

Dans le navigateur web qui est lancé automatiquement, vous pouvez alors accéder à l'interface web d'administration du serveur GlassFish. Le chargement de la page peut prendre un peu de temps mais vous devez obtenir quelque-chose qui ressemble à la figure ci-dessous.

Si un login et un mot de passe vous sont demandés, utilisez `admin` pour le login et `adminadmin` pour le mot de passe.



- ✓ Quelles applications sont déjà déployées sur le serveur ?

## Création d'un projet Web Application avec NetBeans

NetBeans est basé sur la notion de projet. Un projet correspond à tout ou partie d'une application que l'on souhaite développer.

- ✓ Aller dans le menu File > New Project. Choisir Web Application dans la catégorie Java Web.
- ✓ Dans l'écran Name and Location, nommer le projet "FactorialWebService" et l'enregistrer dans C:\Utilisateurs\[votre login]\Documents\NetBeansProjects. ATTENTION, cette étape est très importante sans quoi vous ne pourrez pas déployer votre module sur le serveur !

**Si le répertoire "NetBeansProjects" n'existe pas** dans "C:\Utilisateurs\[votre login]\Documents\", créez le.

- ✓ Dans l'écran Server and settings, laisser les valeurs par défaut.
- ✓ Dans l'écran Frameworks, laisser les valeurs par défaut. Puis, cliquer sur Terminer.

Le projet que vous venez de créer contient tous les éléments de base d'une application web. L'étape suivante consiste à déployer cette application sur le serveur et à tester son fonctionnement.

## Déploiement de l'application web créée

Par défaut, lors de la création d'un projet de type Web Application, NetBeans crée une page web appelée `index.jsp` dans le répertoire Web Pages. Cette page sert de page d'accueil par défaut de l'application web ainsi créée.

- ✓ Ouvrir le fichier `index.jsp`.

Cette page d'accueil générée par défaut par NetBeans contient le code suivant :

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

Sans rentrer dans les détails, cette page web contient du code HTML et également du code JSP (Java Server Pages) entre `<%` et `%>`. JSP est le langage permettant d'inclure du code Java dans des pages web afin de créer une interface web pour une application Java. Nous n'allons pas nous servir de la technologie JSP par la suite, mais cette page va simplement nous permettre de tester le bon déploiement de notre application : une fois déployée, lors de son exécution cette application doit afficher le texte "Hello World!" dans un navigateur web.

- ✓ Sélectionner le projet de votre Web Application puis choisir **Clean and Build** dans le menu contextuel (clic droit).

Cette action déclenche la compilation du code de l'ensemble du projet et la génération d'une archive `.war` qui peut être déployée sur le serveur d'applications.

- ✓ Sélectionner le projet de votre Web Application puis choisir **Deploy** dans le menu contextuel (clic droit).

Cette action déclenche le déploiement de l'application sur le serveur.

- ✓ Avec un navigateur web, aller à l'URL <http://localhost:8080/FactorialWebService/>. Vous devez obtenir une page telle que celle-ci :



- ✓ Vérifier dans la console d'administration de GlassFish que l'application apparaît bien dans la liste des applications déployées (éventuellement rafraîchir la page). Tester le lancement de l'application depuis la console en utilisant le lien Launch.

**Vous êtes maintenant prêt à utiliser votre serveur d'applications pour déployer des Web Services !**

## Développement d'un service WS-\*

L'objectif de cet exercice est de **développer un service WS-\*** qui reçoit en paramètre un entier  $n$ , calcule  $n!$  (factorielle) et retourne le résultat.

Le code de cet exercice est à placer dans le projet "FactorialWebService" créé lors de l'exercice précédent.

- ✓ Dans le dossier "Source Packages", créer un package Java. Dans ce package, créer une classe Java avec une méthode qui pour tout entier  $n$  retourne  $n!$ .
- ✓ Annoter la classe de manière à l'exposer en tant que Web Service.
- ✓ Déployer votre application sur le serveur (clic droit puis Deploy).
- ✓ Regarder l'onglet GlassFish Server. A quelle URL votre service peut-il être interrogé ? Ou peut-on télécharger le contrat WSDL du service ? Et le XML Schema des données échangées par votre service ?
- ✓ Dans un navigateur web, taper l'URL de votre service, suivie de ?Tester. Qu'avez-vous sous les yeux ?

**Dans cet exercice, vous avez vu comment développer un Web Service à partir d'une classe Java.**

## Développement d'un service RESTful

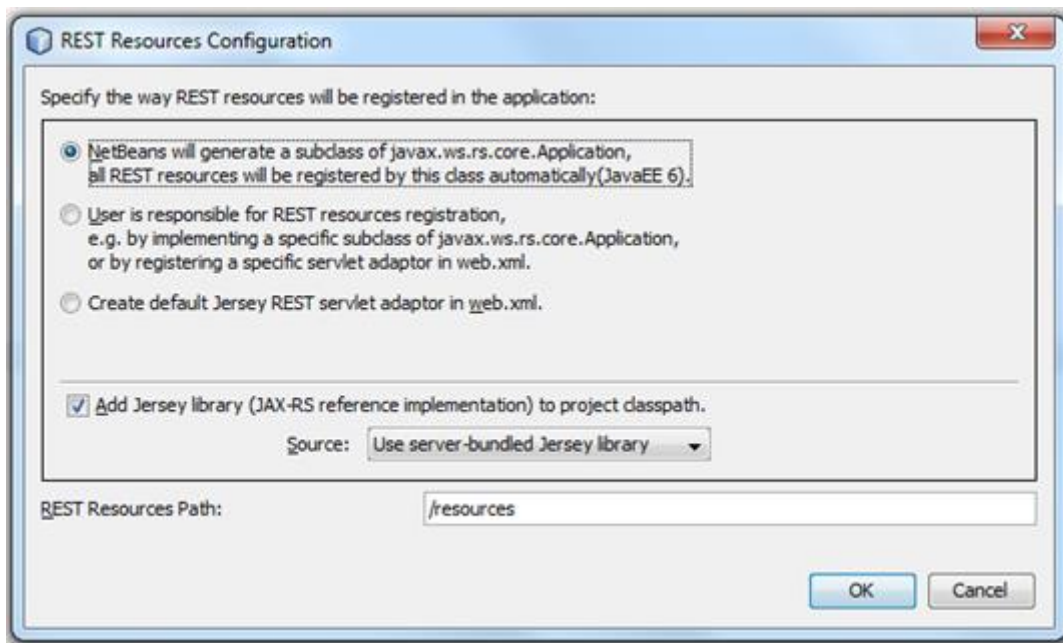
La création d'un service RESTful ressemble à la création d'un service WS-\* puisqu'il faut créer une classe Java et l'annoter (ici l'API Java utilisée est JAX-RS).

Cependant, la philosophie REST implique des différences fondamentales dans la façon de concevoir les services. En effet, pour être conforme à la vue "ressources" de REST lorsqu'on développe un service, il ne s'agit pas de concevoir une classe qui met à disposition des méthodes, mais de concevoir un ensemble de ressources (on utilise généralement une classe pour le conteneur et des classes pour les ressources contenues) que l'utilisateur peut consulter et modifier.

Pour cet exercice, vous implémenterez un service RESTful qui représente une ressource de type message de bienvenue.

- Créer un projet de type Web Application.
- Créer une classe Java qui représente la ressource "message de bienvenue".
- Annoter la classe pour l'exposer sous la forme d'un service RESTful répondant aux requêtes GET, la réponse étant alors le contenu du message de bienvenue.

Si NetBeans demande de quelle façon vous souhaitez enregistrer votre service, comme indiqué sur la figure ci-dessous, choisir la première option comme indiqué ci-dessous pour que NetBeans "enregistre" automatiquement votre service :



L'enregistrement signifie que le serveur d'applications génère une application chargée de traiter les requêtes HTTP entrantes et les redirige vers le service, et inversement pour les réponses fournies par le service. Le nom de cette application générée par NetBeans (par défaut : "resources") apparaîtra dans l'URL du service juste après le nom du projet.

- Tester le service avec un client HTTP avec l'une des deux méthodes vues dans l'exercice précédent.

**Dans ce dernier exercice, vous avez découvert comment créer un Web Service RESTful basique.**

Des outils de génération automatique de code permettent de créer des services et des clients plus complets que ceux vus dans cet exercice. Jersey est une implémentation de JAX-RS fournie par Sun (Oracle) et permet de faire de genre d'opérations. Vous pouvez tester Jersey en utilisant les assistants de NetBeans suivants :

- File > New File > Web Services > RESTful Web Service from patterns (choisir "Simple Root Resource" pour le plus simple des services) pour créer un service ;
- File > New File > Web Services > RESTful Java Client pour créer un client.

# Chapitre 7 : Exécution des processus

Ces exercices ont pour objectif de vous familiariser avec l'exécution de processus de manière concrète. Pour cela, on s'intéresse à une entreprise qui souhaite informatiser son processus de gestion des demandes de congés dans le cadre d'une démarche SOA.

La solution retenue est d'utiliser un moteur de workflow appelé Bonita BPM sur lequel le processus sera adossé et qui permettra de faire des appels à des services applicatifs exposés au niveau du SI de l'entreprise.

## La démarche proposée est la suivante :

1. Analyser l'existant de l'entreprise : processus existant, applicatif existant, services existants
2. Tester le moteur de workflow et déterminer comment l'interfacer avec un web service
3. Réaliser le modèle complet du processus de façon à ce qu'il puisse être exécuté par le moteur de workflow
4. Ajouter des fonctionnalités d'envoi de mails au processus.

## Analyse de l'existant de l'entreprise

Cette partie présente différentes informations concernant le système de gestion des congés existant dans l'entreprise. Votre objectif est de prendre connaissance de ces informations pour pouvoir définir une façon d'adosser le processus de gestion des congés de l'entreprise sur le moteur de workflow.

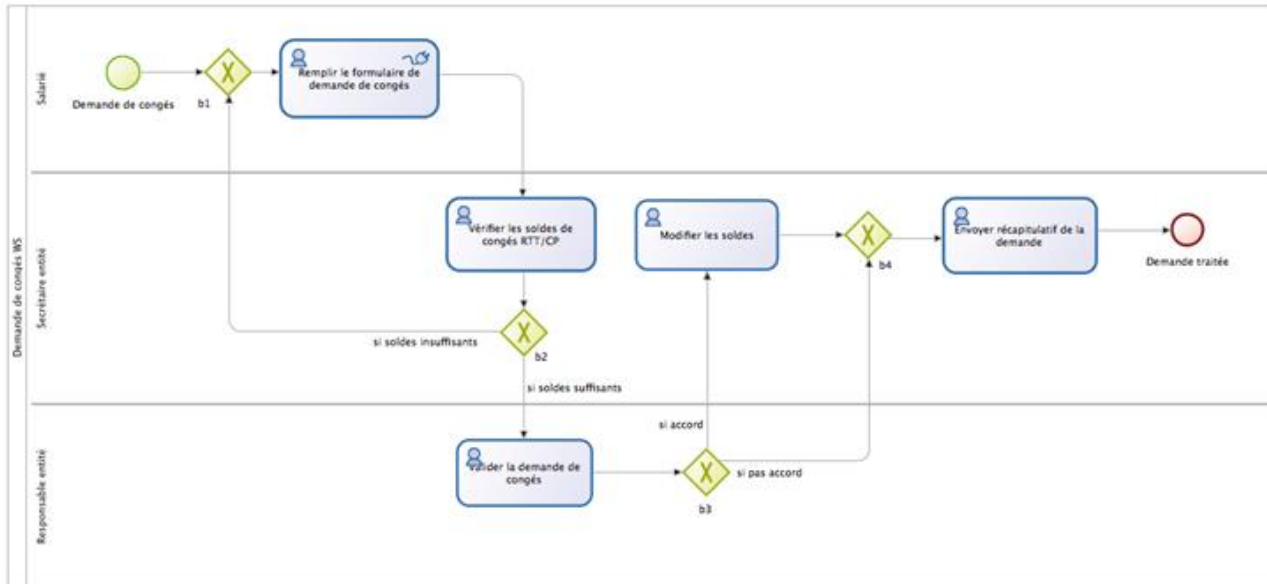
### Processus de gestion des congés

Le processus de gestion des congés actuel de l'entreprise est entièrement manuel et repose sur un formulaire papier qui circule entre l'employé qui demande des congés, la secrétaire d'entité qui gère les soldes des employés et le responsable d'entité qui valide les demandes.

- ✓ Créer un document texte simple qui vous permettra de sauvegarder vos réponses aux questions au fur et à mesure des questions.
- ✓ Lister les activités du processus qui peuvent être totalement automatisées, c'est-à-dire réalisées par des services exposés au niveau du SI de l'entreprise.

L'objectif des étapes suivantes est de prendre connaissance de l'existant applicatif de l'entreprise et d'identifier les services disponibles sur lesquels le processus pourrait être adossé.

Un modèle BPMN du processus a été réalisé :



## Application de gestion des congés

L'entreprise dispose d'une application web pour la gestion des congés. Cette application est utilisée par la secrétaire d'entité pour vérifier et modifier les soldes CP et RTT des salariés.

Gestion des congés						
Liste des comptes congés						
Identifiant ▲	Prénom ◄	Nom ◄	Solde CP ◄	Solde RTT ◄		
bunny	Bugs	Bunny	25	10	Modifier	Supprimer
duck	Donald	Duck	25	10	Modifier	Supprimer
mouse	Mickey	Mouse	25	10	Modifier	Supprimer
Créer un compte						

Pour pouvoir tester le fonctionnement de cette application web, vous allez **l'installer sur votre poste** et la déployer sur le serveur GlassFish que vous avez déjà utilisé auparavant.

- ✓ Le dossier **conges.zip** contient l'application que vous devez l'enregistrer dans "C:\Utilisateurs\[votre login]\Documents\NetBeansProjects" (ATTENTION, cette étape est très importante sans quoi vous ne pourrez pas déployer votre module sur le serveur).

Si le répertoire "NetBeansProjects" n'existe pas dans "C:\Utilisateurs\[votre login]\Documents\", créez le.

- ✓ Dézipper **conges.zip**, vous devez obtenir un répertoire nommé "Conges".
- ✓ Lancer NetBeans, puis faire "File > Open Project", aller chercher le répertoire de l'application ("Conges") et cliquer sur "Open project".
- ✓ Dans l'onglet "Projects" de NetBeans, cliquer droit sur le projet "Conges", sélectionner "Clean and build" puis "Deploy", et enfin "Run".
- ✓ Vérifier que l'application se lance et que vous pouvez accéder à la page web de gestion des congés. **Tester le fonctionnement de l'application.**



## Web services pour la gestion des congés

Dans une étape de prototypage, la DSI de l'entreprise a réalisé un premier web service qui permet d'exposer certaines fonctionnalités de l'application de gestion des congés.

- ✓ Ou se trouve le web service en question dans l'application ?
- ✓ Dans votre fichier texte de réponse, indiquer quelles sont les opérations exposées via ce service. Comment peuvent-elles être utilisées dans le processus de gestion des congés ?
- ✓ Tester le web service et afficher son contrat WSDL.

Dans la partie suivante, vous devrez **réutiliser le contrat WSDL du service, ainsi que les messages SOAP échangés avec le service** lors de ces tests.

### Test du moteur de workflow Bonita avec appel de web service

L'objectif de cette première étape est de comprendre comment Bonita fonctionne : informations qu'il est nécessaire de renseigner pour rendre un processus exécutable, génération automatique de formulaires, personnalisation des formulaires, interface utilisateur...

### Création et exécution d'un processus simple avec Bonita

- ✓ Lancer Bonita et créer un nouveau processus.
- ✓ Pour le moment, *ne créer que la première tâche du processus*, "remplir le formulaire de demande de congés", comme indiqué sur la figure ci-dessous.



- ✓ Cliquer sur la pool du processus. Dans l'onglet "Général", partie "Données", **créer trois variables** : une pour l'identifiant du salarié (chaîne de caractères), une pour le solde courant de RTT dont dispose l'employé (entier) et enfin une pour le nombre de jours de RTT que le salarié souhaite poser (entier).
- ✓ Lancer le processus pour tester le résultat.
- ✓ Vous devez obtenir l'affichage successif de 2 formulaires contenant chacun 3 champs : un pour l'identifiant, un pour le solde de RTT de l'employé, et un pour le nombre jours de RTT demandés.
- ✓ Personnaliser les formulaires (il est possible de ne pas faire apparaître certains champs ou de rendre certains champs non modifiables par exemple) et ajouter d'autres données (pour gérer les CP notamment).

A ce stade, vous avez modélisé un processus très simple et vous avez rendu ce processus exécutable. Vous avez pu tester l'exécution de votre processus par le moteur de workflow Bonita. L'objectif des étapes suivantes est de connecter ce processus au web service de l'application de gestion des congés.



## Appel du web service

L'objectif de cette étape est de permettre au salarié qui fait sa demande de congés de voir **son solde courant de jours de RTT** s'afficher lorsqu'il remplit le formulaire de demande de congés. Pour cela, nous utiliserons le web service découvert ci-dessus.

Il est possible d'appeler un web services dans une activité Bonita en utilisant le connecteur "Client Web Service" de la catégorie "Web Services".

- ✓ Sur l'unique tâche de votre processus, ajouter un connecteur "Client Web Service" : dans l'onglet "Général", partie "Connecteur", faire "ajouter" puis sélectionner le connecteur "Client Web Service" dans la catégorie "Web Services".
- ✓ Donner un nom au connecteur, et sélectionner l'événement "enter". Cela signifie que le web service sera appelé à l'entrée dans la tâche, ce qui permettra d'afficher le résultat de l'appel au web service directement dans le deuxième formulaire présenté à l'utilisateur.
- ✓ Pour configurer l'accès au web service, trouver les informations suivantes dans le **contrat WSDL** du service :
  - Dans le champ "Cible NS / Target name space" du formulaire, mettre la valeur de l'attribut "**targetNamespace**" de l'élément "**<definitions>**" (premier élément du fichier WSDL).
  - Dans le champ "Nom du service / Service name" du formulaire, mettre la valeur de l'attribut "**name**" de l'élément "**<service>**".
  - Dans le champ "Nom du port / Port name" du formulaire, mettre la valeur de l'attribut "**name**" de l'élément "**<port>**".
  - Dans le champ "Requête / Request" du formulaire, mettre la requête SOAP à envoyer au service :

Vous pouvez utiliser l'une des requêtes SOAP générées lors des tests du service à l'étape précédente. Voici un exemple :

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:c="http://www.oorsprong.org/websamples.countryinfo">
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <c:CapitalCity>
      <c:sCountryISOCODE>FR</c:sCountryISOCODE>
    </c:CapitalCity>
  </soap:Body>
</soap:Envelope>
```

Dans cette requête, vous devez **fournir l'identifiant d'un salarié** pour que le web service renvoie son solde de RTT. Vous avez deux options :

1. donner un identifiant prédéfini, par exemple bunny ;
  2. ou utiliser la variable que vous avez créée pour stocker l'identifiant de l'utilisateur, par exemple \${identifiantSalarie}.
- Dans le champ "Adresse endpoint / Endpoint address" du formulaire, mettre l'URL indiquée en valeur de l'attribut "**location**" de l'élément "**<address>**" (ou "**<soap:address>**").
  - Dans le champ "Binding / Binding" du formulaire, mettre <http://schemas.xmlsoap.org/wsdl/soap/>
  - Dans le champ "SOAP Action" du formulaire, ne **rien** mettre.

Il faut ensuite configurer la sortie du connecteur de manière à récupérer dans une variable de votre processus la valeur qui est retournée par le service. En effet, le web service renverra une réponse SOAP comme celle-ci :

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
  <m:CapitalCityResponse xmlns:m="http://www.oorsprong.org/websamples.countryinfo">
    <m:CapitalCityResult>Paris</m:CapitalCityResult>
  </m:CapitalCityResponse>
</soap:Body>
</soap:Envelope>
```

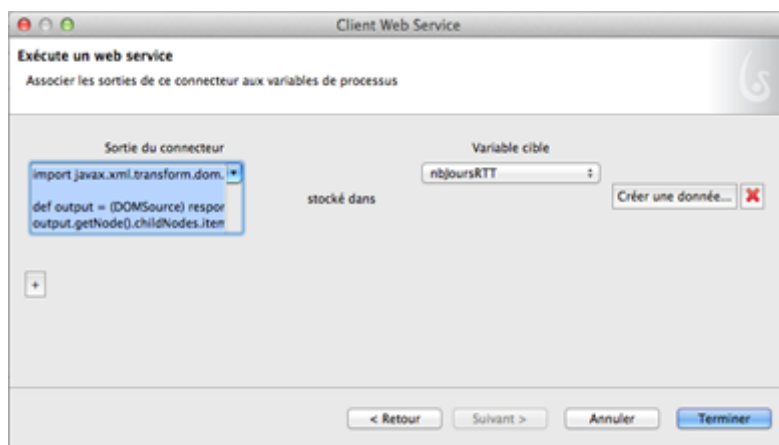
Il faudra donc extraire la valeur retournée par le service du message SOAP obtenu.

- ✓ Configurer la sortie du connecteur de manière à stocker le résultat de l'appel du service dans une variable du processus :
- dans la partie "sortie du connecteur", sélectionner "editer l'expression" et copier le script ci-dessous à la place de "response". Ce script, écrit avec Groovy un langage de script construit sur Java, manipule l'arbre DOM du message SOAP retourné par le service :

```
import javax.xml.transform.dom.DOMSource;

def output = (DOMSource) response
output.getNode().childNodes.item(0).textContent
```

- dans la partie "variable cible", sélectionner la variable représentant le solde de RTT.



- ✓ Lancer le processus pour tester le résultat.

Lors de l'affichage du deuxième formulaire, **le champ pour le solde de RTT doit être rempli avec le solde courant du salarié** que vous avez choisi.

**Vous avez maintenant un processus exécutable qui fait appel à un web service ! Il ne vous reste plus qu'à terminer de modéliser le processus de gestion des congés...**

## Processus de gestion des congés complet

Il s'agit maintenant d'obtenir un modèle complet et exécutable du processus de gestion des congés dans Bonita.

- ✓ Compléter le modèle du processus de gestion des congés : ajouter les branchements, les autres tâches, configurer les données nécessaires, expliciter les conditions sur les flux sortant des branchements (les opérateurs de test ont une syntaxe proche de Java : ==, !=, >, >=, ...), personnaliser les formulaires...
- ✓ Compléter le web service de l'application de gestion des congés pour qu'il expose l'ensemble des fonctions nécessaires pour le processus. S'inspirer des fonctions déjà exposées dans le service.
- ✓ Configurer les connecteurs sur les tâches du processus qui nécessitent l'accès au web service.
- ✓ Et bien sûr, **tester régulièrement** l'exécution de votre processus !

## Envoi d'emails

L'entreprise souhaite qu'un email soit envoyé au salarié à la fin du processus de demande de congés avec le récapitulatif de sa demande et de la décision obtenue.

- ✓ Ajouter une tâche au processus pour modéliser l'envoi du mail.
- ✓ Sur cette tâche, dans la partie "Connecteurs", ajouter un connecteur d'envoi d'email (catégorie "messaging").
- ✓ Configurer le connecteur. Vous pouvez utiliser les informations suivantes :
  - Hôte : le serveur SMTP à utiliser (pour Supélec il s'agit de smtp.supelec.fr)
  - Port : 465
  - Authentification : votre login/mot de passe
- ✓ Créer un email type. Vous pouvez utiliser les variables du processus. Attention : pour le champ "À" **utiliser une adresse email de destinataire que vous pourrez consulter** (pour les tests...) !
- ✓ Tester votre processus.

Dans ces exercices, vous avez pu découvrir un outil d'exécution de processus orienté workflow appelé Bonita. Il est possible de déployer ce type d'outils sur un ESB de manière à supporter à la fois les orchestrations et les workflow avec une grande palette de connecteurs. Vous avez pu vous rendre compte que l'exécution d'un processus nécessite une configuration très fine des utilisateurs, des données et des connecteurs en plus de la modélisation et que toutes les constructions de BPMN ne sont pas supportées par un outil d'exécution tel que Bonita.