

ACSSI

DATA
MANAGEMENT
ERP

Talend Studio

SEPT. 2025

The Talend logo, featuring the word "talend" in a white sans-serif font inside a red circle.

Notre démarche Qualité nous tient à cœur...

Dans le cadre de votre formation, plusieurs documents seront à compléter :

Participant

- Formulaire d'entrée : vous pouvez y exprimer vos besoins spécifiques et vous positionner par rapport au prérequis
- Formulaire de satisfaction : en fin de formation, vous pouvez y exprimer votre ressenti sur la session
- Formulaire d'évaluation : en fin de formation, un ensemble de questions théoriques vous permettront de valider les objectifs de formation
- Signer nos feuilles de présence une pour chacune des demi-journées

Formateur

- Analyse les résultats du formulaire d'entrée : comprendre vos besoins et vous adapter la formation à votre maîtrise des prérequis
- Analyse de votre satisfaction : via les résultats du formulaire de satisfaction
- Corriger avec vous le formulaire d'évaluation : Avez-vous des questions supplémentaires ? Tout est compris ?
- Formulaire de satisfaction : en fin de formation, vous pouvez y exprimer votre ressenti sur la session

PS : Les liens des formulaires sont disponibles dès qu'un mot est sous-ligné.

EN CHIFFRES

ACSSI

(ON PRONONCE « AXI » ☺)



+ 300

CLIENTS

Réactivité et Proximité



+ 270

COLLABORATEURS

Société à taille humaine



3 M€

CHIFFRE D'AFFAIRE

En 2021



33

ANNÉES D'XP

Savoir faire reconnu

Lille



Paris



Bordeaux

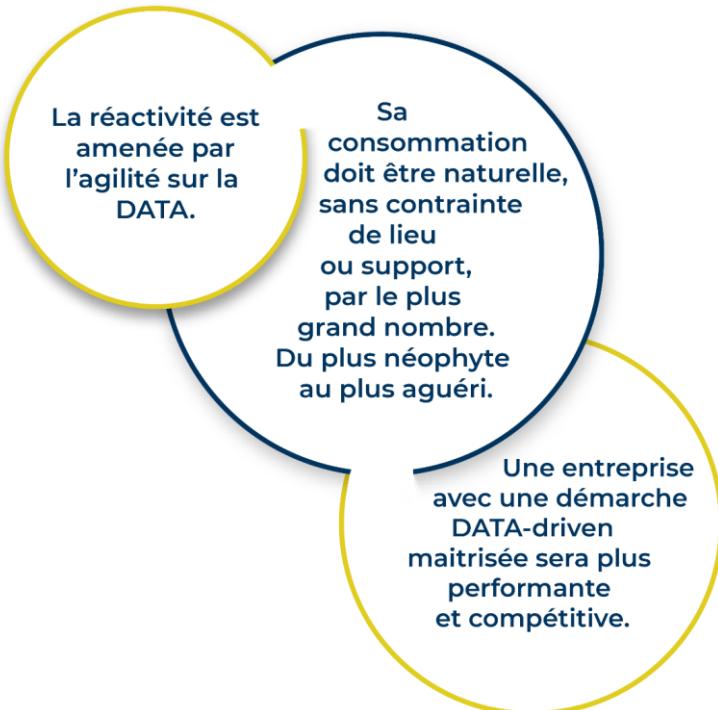


Nantes



NOTRE VISION

LA DATA AU CŒUR DE LA STRATÉGIE D'ENTREPRISE



NOS PARTENAIRES ET TECHNOLOGIES



DATA MANAGEMENT

Collecte
Intégration
Réplication
Gouvernance
Catalogage
Qualité
Préparation
ETL / ESB



DATA VIZ / ANALYTICS / INTELLIGENCE

Visualisation
Self-Service
Reporting / Alerting
IA
Science de données
Machine Learning
Algorithme



Power BI



Tableau Software

DATA CLOUD



DATA SCIENCES



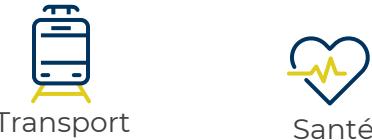
CLOUD PROVIDER



+ 1000 sources de données

DES SECTEURS D'ACTIVITÉ VARIÉS

+200 CLIENTS DATA





SUIVEZ-NOUS !

LinkedIn



YouTube

facebook



webikeo

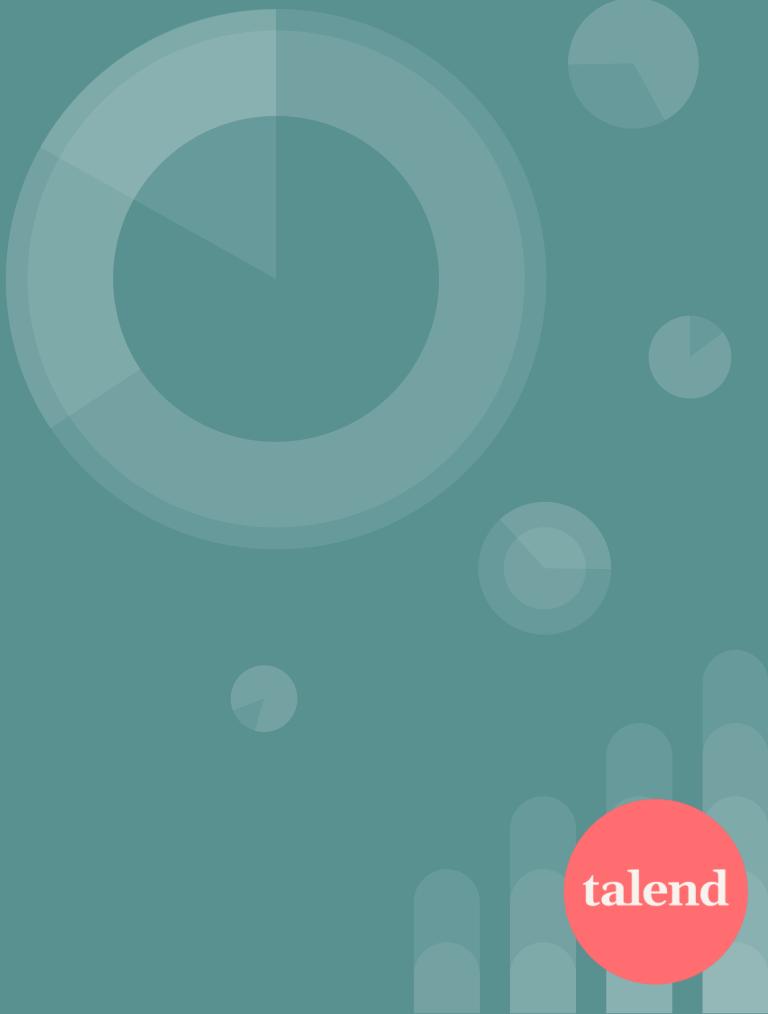
ET POUR DU CONTENU PLUS TECH'

NOS BUILDERS !

LinkedIn



Participants



Objectifs de formation

Les objectifs :

- Connaitre l'interface et le fonctionnement de Talend
- Savoir manipuler des données
- Connaitre les composants principaux de Talend
- Adopter de bonnes pratiques de développement et d'organisation

Evaluation des objectifs :

Un questionnaire d'évaluation des objectifs est réalisé en fin de formation. Celui-ci permet d'avoir un retour du niveau de compréhension de l'élève sur ce qui a été enseigné. Chaque réponse est accompagnée d'un commentaire détaillé.

Programme

1 - Présentation ACSSI

2 - Participants

3 - Talend

4- Installation et configuration du studio

5 - Présentation de Talend Open Studio

6 - Fonctions Internes

7 - Variables

8 – Exercices

9 - Bonnes pratiques

10 - Code et routines

11 - Métadonnées

12 - Compilation

13 - Cas pratique

14 – Pour approfondir

POUR CONNAÎTRE VOS ATTENTES...

For

- Avant la formation, vous avez pu remplir le formulaire suivant : <https://forms.office.com/r/jEquURRwup>
- Votre formateur du jour, a pu analyser et préparer sa formation en fonction de vos besoins et votre niveau de connaissance sur l'outil.
- Si cela n'a pas été fait, nous vous encourageons à le réaliser maintenant 😊

Questionnaire d'entrée

Index Formation Année Type de formation Société
FD 24-06-2024 Conventionné M2I ACSSI Envoi Mail Abandonné

Nombre de participants 7/8 Participants informés du programme

Objectifs prioritaires selon la formation

Objectif prioritaire	Nombre de vote
Totaux	7
Organiser une application Qlik Sense en répondant aux bonnes pratiques de l'éditeur (Dashboard, Analysis, Report)	4
Savoir agréger les données grâce aux fonctions Qlik	2
Je n'ai pas particulièrement de priorité	1

Niveau de connaissance à l'entrée

Niveau de connaissance	Nombre de personnes
3/5	1
1-2/5	6

Maîtrise des bases informatiques

Maîtrise	Nombre de personnes
Maîtrisé partiellement	1
Maîtrisé	7

Maîtrise de la visualisation de données

Maîtrise	Nombre de personnes
Maîtrisé partiellement	2
Maîtrisé	5

ACSSI

Nom Prénom	Société	Type de formation	Date Forma...	Objectif Personnel/Complémentaire	Raison de Formation	Impact sur le quotidien
Florence	[redacted]	Qlik Sense Designer	24/06/2024	Créer un onglet analysis qui soit facile d'utilisation pour les utilisateurs	L'introduction de Qlick dans l'entreprise	Analyse facilité et gain de temps sur les tâches de bord
Rolland	[redacted]	Qlik Sense Designer	24/06/2024	Savoir maîtriser à minima Qlik	La volonté pour l'entreprise de se développer informatiquement en interne et permettre à tous les utilisateurs de pouvoir analyser des données	Une expérience supplémentaire qui ne porte que du bénéfice tant d'un point de vue professionnel que personnel
Ronan	[redacted]	Qlik Sense Designer	24/06/2024	-	Besoin de capturer un maximum de données pour	Précisions des données et Gains de temps

- PS : Une application Qlik Sense est accessible et visible pour nos formateurs, [ici](#)

Acssi

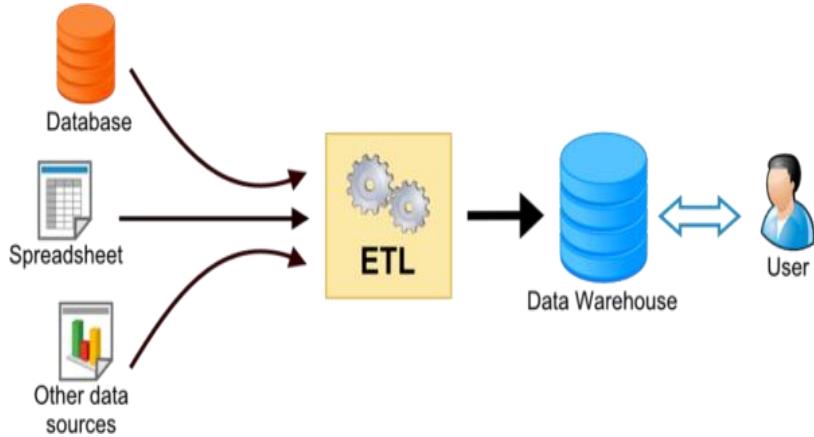
DATA
MANAGEMENT
ERP

Talend



talend

Présentation de Talend



Talend Open Studio (TOS) est une plateforme d'intégration de données Open Source, basée sur le langage **Java**.

Il a été créé en 2006 en France par Bertrand Diar et Fabrice Bonant.

TOS permet de répondre à toutes les problématiques liées au traitement des données dans la chaîne décisionnelle .

Talend est un ETL (Extract Transform and Load) qui permet d'extraire des données d'une source, de modifier ces données, puis de les recharger vers une destination. Il peut aussi être utilisé en mode ELT (Extract, Load, Transform).

La source et la destination des données peuvent être une base de données, un service web, un fichier plat et bien d'autres...

Talend peut donc être utilisé dans n'importe quel contexte où des données sont véhiculées.

ETL ou ELT ?

Ces 2 modes ont une même vocation : extraire les données, les transformer et les charger.

La différence vient du procédé utilisé.

ETL :

- Exploité dans des environnements très hétérogènes
- Pour les transformations, c'est un moteur engine autre que ceux des SGBD qui est utilisé (exemple le moteur JAVA ou PERL pour Talend)
- Il traite ligne par ligne pour faire les traitements

ELT :

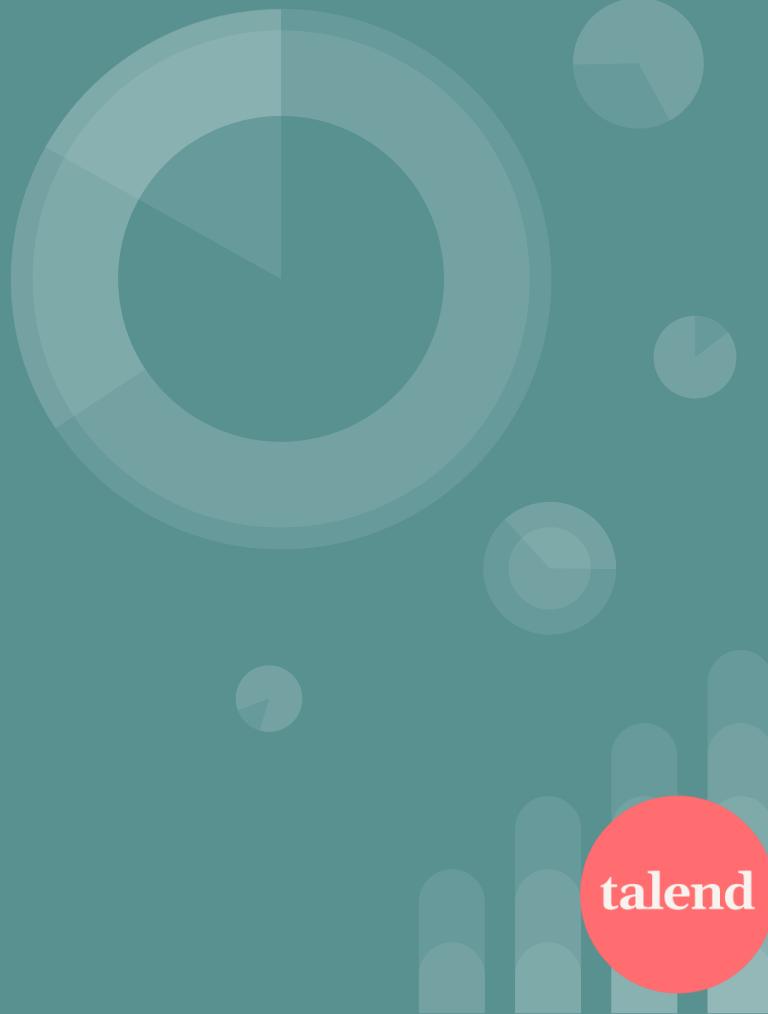
- Exploité dans des environnements homogènes
- Utilise seulement les moteurs SGBD (donc les traitements aussi sont faits par ces moteurs)
- Permet de charger plutôt de gros volumes

Il est possible de coupler les deux approches pour tirer parti des deux avantages mais encore faut-il savoir bien analyser le contexte, sinon on pourrait arriver à de grosses pertes de performances.

Acssi

DATA
MANAGEMENT
ERP

Installation & Configuration



Environnement

Serveur:

- OS: Linux / windows
- RAM Le plus possible avec une recommandation minimum de 4Go
- Espace disque : 20 Go
- Version java la plus récente préférable

Poste de travail:

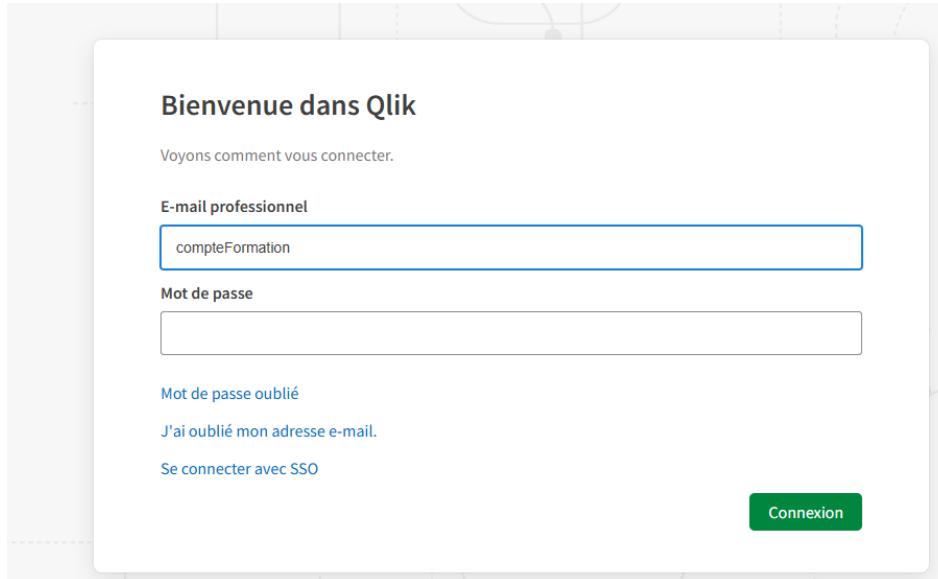
- OS: Windows ou Linux
- RAM 3Go (minimum) / 4Go (recommandé)
- Espace disque : 3Go et +
- Version java la plus récente préférable

Plus d'info ici

<https://help.talend.com/reader/UA53PA1vZV-qvw4v3FgUkQ/MDauhq9nl1m0FXZX-aHp9Q>

Installation & Configuration

- Téléchargez le studio en se connectant avec le compte de formation fourni sur le site :
<https://tmc.eu.cloud.talend.com/>



Plus d'info ici

<https://help.talend.com/reader/UA53PA1vZV-qvw4v3FgUkQ/MDauhg9nl1m0FXZX-aHp9Q>

Installation & Configuration

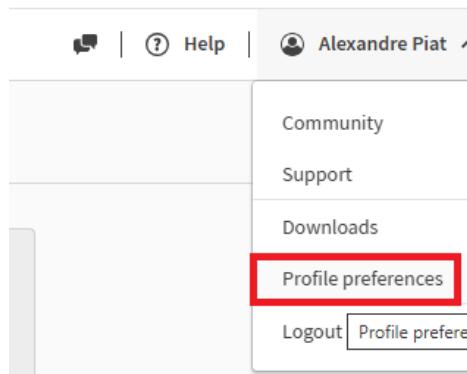
The image displays three side-by-side windows of the Talend Studio installer v8.0.1. All three windows show the 'License agreement' step.

Window 1 (Left): Shows the 'Welcome to the Talend Installation Wizard.' screen. It includes a note about hardware and software requirements and a link to the 'Talend Installation Guide'.

Window 2 (Middle): Shows the 'License agreement' screen. It displays the QLIK(r) CUSTOMER AGREEMENT text, which is a standard software license agreement. Below the text, there are two radio buttons for accepting the terms: ' I agree to all listed terms and conditions' and ' I do not agree to all listed terms and conditions'. The 'I agree to all listed terms and conditions' option is highlighted with a red border.

Window 3 (Right): Shows the 'Specify the directory where your Talend modules will be installed' screen. It shows the 'Installation directory' set to 'C:\Program Files (x86)\Talend-Studio'. The 'Next >' button is highlighted with a red border.

Installation & Configuration



This screenshot shows the "Profile preferences" page in the talend application. The title is "Profile preferences" and the subtitle is "Your personal information and preferences". On the left is a sidebar with links: "About Me", "Notifications", "Language & Region", and "Personal access tokens" (which is highlighted with a red box). On the right is a main content area with fields for "Name" (containing "presentation"), "Studio8", and a "Generate" button.

This screenshot shows the "Add personal access token" page in the talend application. It has a "Token name*" field containing "nomDuToken" (highlighted with a red box) and a "Generate" button (highlighted with a red box). Below the form is a section titled "Permissions" with the text: "Permissions for the token depend on the roles assigned to the roles of this user."

Installation & Configuration

The screenshot displays two windows side-by-side:

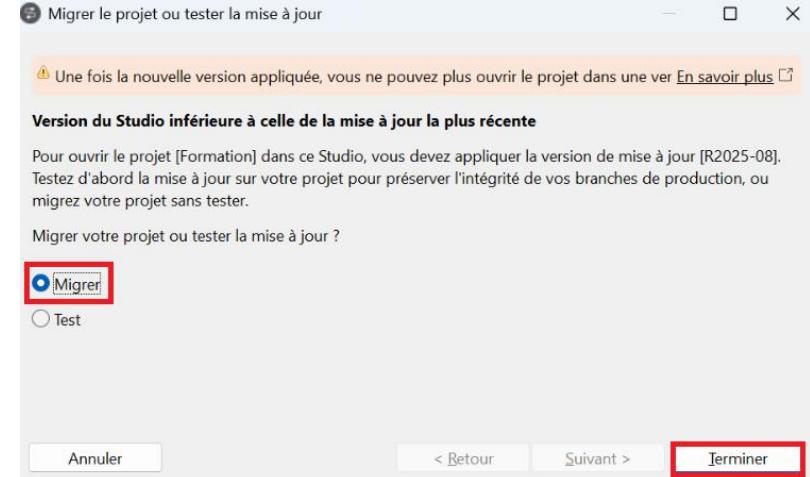
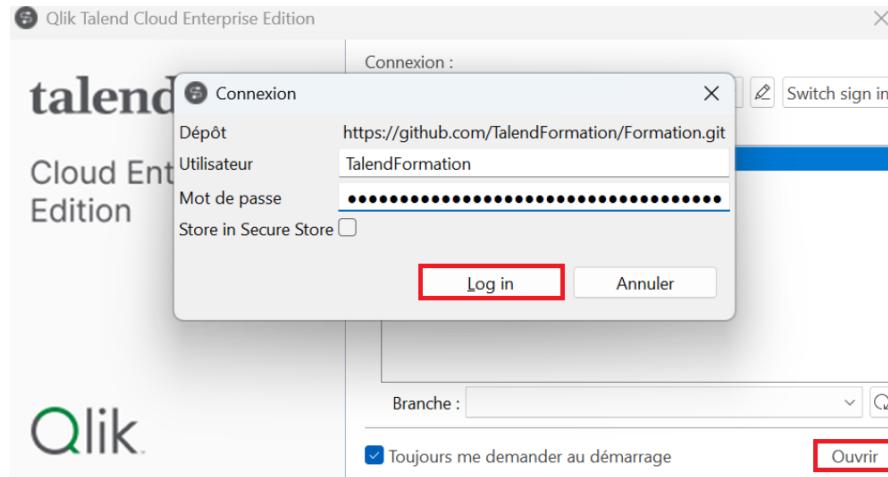
Talend Studio Window:

- Header: Welcome to Talend Studio
- Left sidebar: talend 8
- Main content:
 - Bienvenue dans le Studio Talend.
 - Vous pouvez choisir la manière dont vous vous connectez. [En savoir plus](#)
 - Buttons: Commencer à utiliser le Studio, Se connecter avec Talend Cloud, Autre mode de connexion (highlighted with a red box)
- Bottom right: Modifier les paramètres

Qlik Welcome to Studio Window:

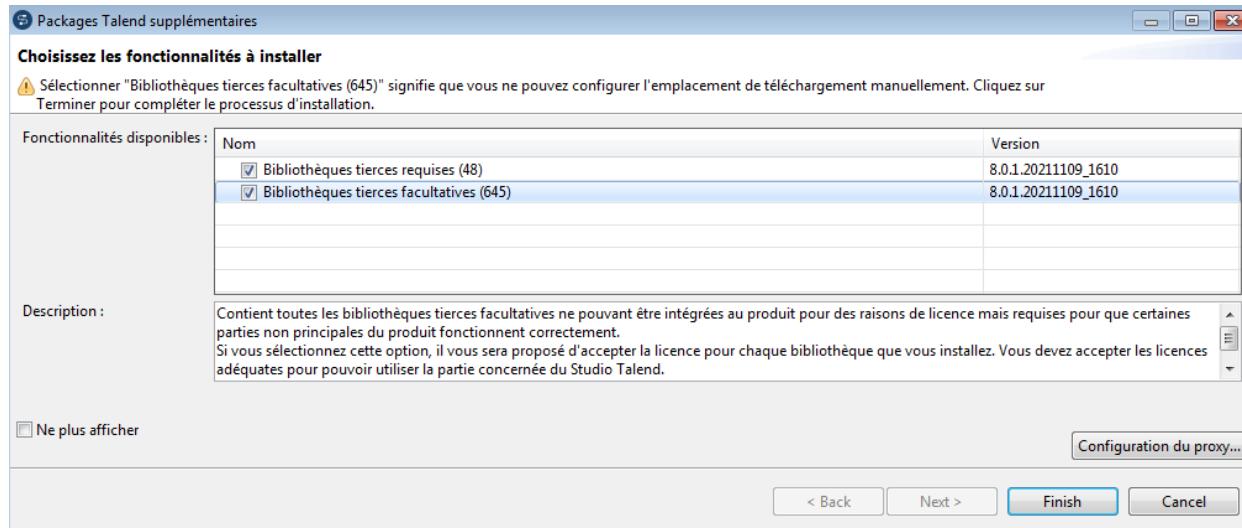
- Header: Welcome to Talend Studio
- Left sidebar: Qlik
- Main content:
 - Importez votre licence produit depuis un hôte distant ou parcourrez votre système de fichiers local pour la trouver :
 - Ma licence produit est sur un hôte distant :
 - AWS - Europe (highlighted with a red box)
 - Jeton : ...
 - Ma licence produit est dans le système de fichiers local :
- Bottom left: Votre licence pour Qlik Talend Cloud Enterprise Edition est valide.
- Bottom right: Configurer le jeton de migration (disabled), Configuration du réseau (disabled), Date d'expiration de la licence : 10/02/2025, Précédent, Suivant (highlighted with a red box)

Installation & Configuration



Installation & Configuration

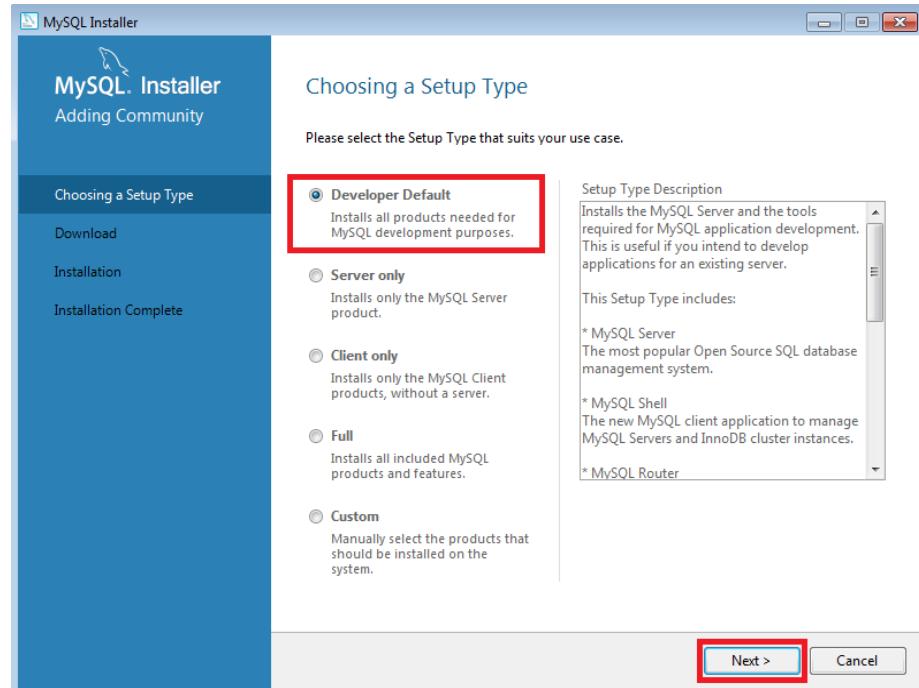
➤ Installation des bibliothèques :



- Bibliothèques requises: bibliothèques nécessaires au bon fonctionnement de Talend.
- Bibliothèques facultatives: permet l'ajout de composants supplémentaires. Peuvent être ajoutées par la suite.

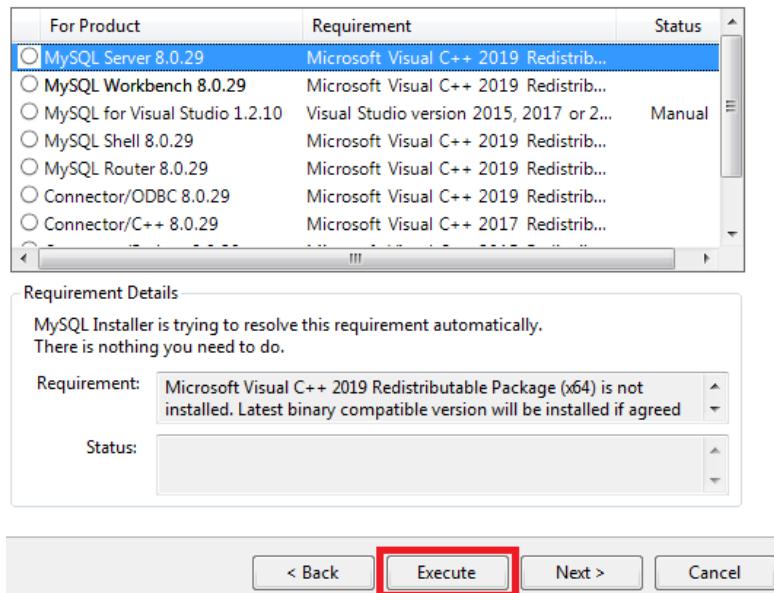
Installation d'une base de donnée :

- Téléchargez MySQL Workbench via le lien: <https://drive.google.com/file/d/15vmXWeKBDL42AO0WT6WZobjF9vAGSQTf/view?usp=sharing>



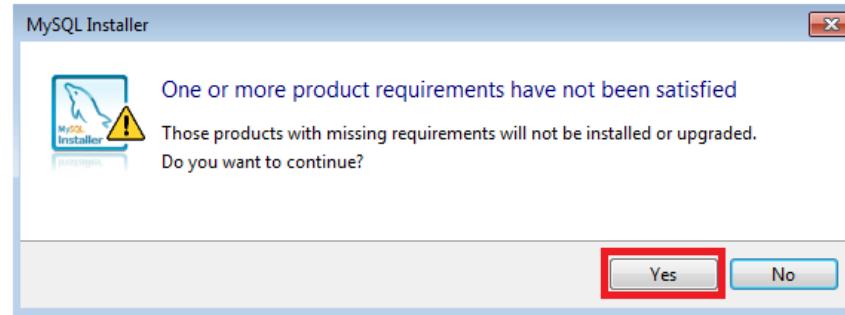
Installation & Configuration

- Installez les dépendances requises en utilisant le bouton Execute
- Acceptez la licence Microsoft
- Continuez avec le bouton Next



Installation & Configuration

- Ignorez les dépendances non obligatoires
- Exécutez les installations



	Connector/C++ 8.0.29	Ready to Install
	Connector/J 8.0.29	Ready to Install
	Connector/.NET 8.0.29	Ready to Install
	MySQL Documentation 8.0.29	Ready to Install
	Samples and Examples 8.0.29	Ready to Install

Click [Execute] to install the following packages.

< Back	Execute	Cancel
--------	---------	--------

Installation & Configuration

- Utilisez les boutons Execute/Next jusqu'à obtenir la création du mot de passe
- Continuez l'installation jusqu'au test de la connexion

Root Account Password
Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password: !

Repeat Password:

	Server	Port	Arch...	Type	Status
<input checked="" type="checkbox"/>	MySQL Server 8.0.29	3306	X64	Stand-alone Server	Connection succeeded.

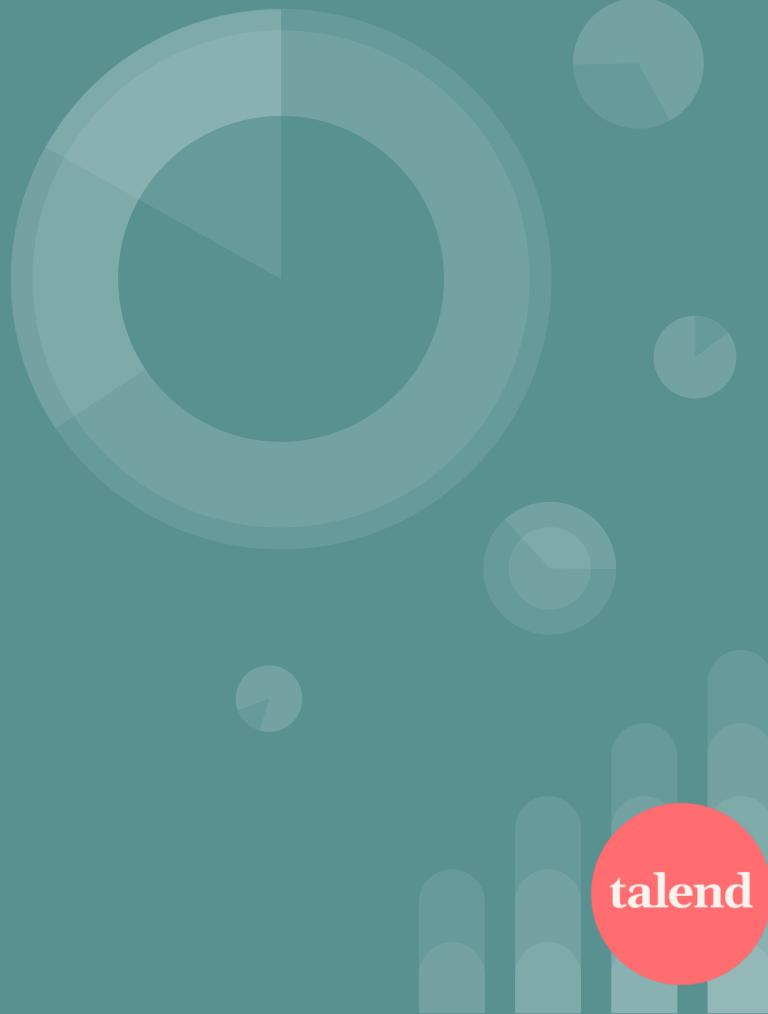
Provide the credentials that should be used (requires root privileges).
Click "Check" to ensure they work.

User name: Credentials provided in Server configuration

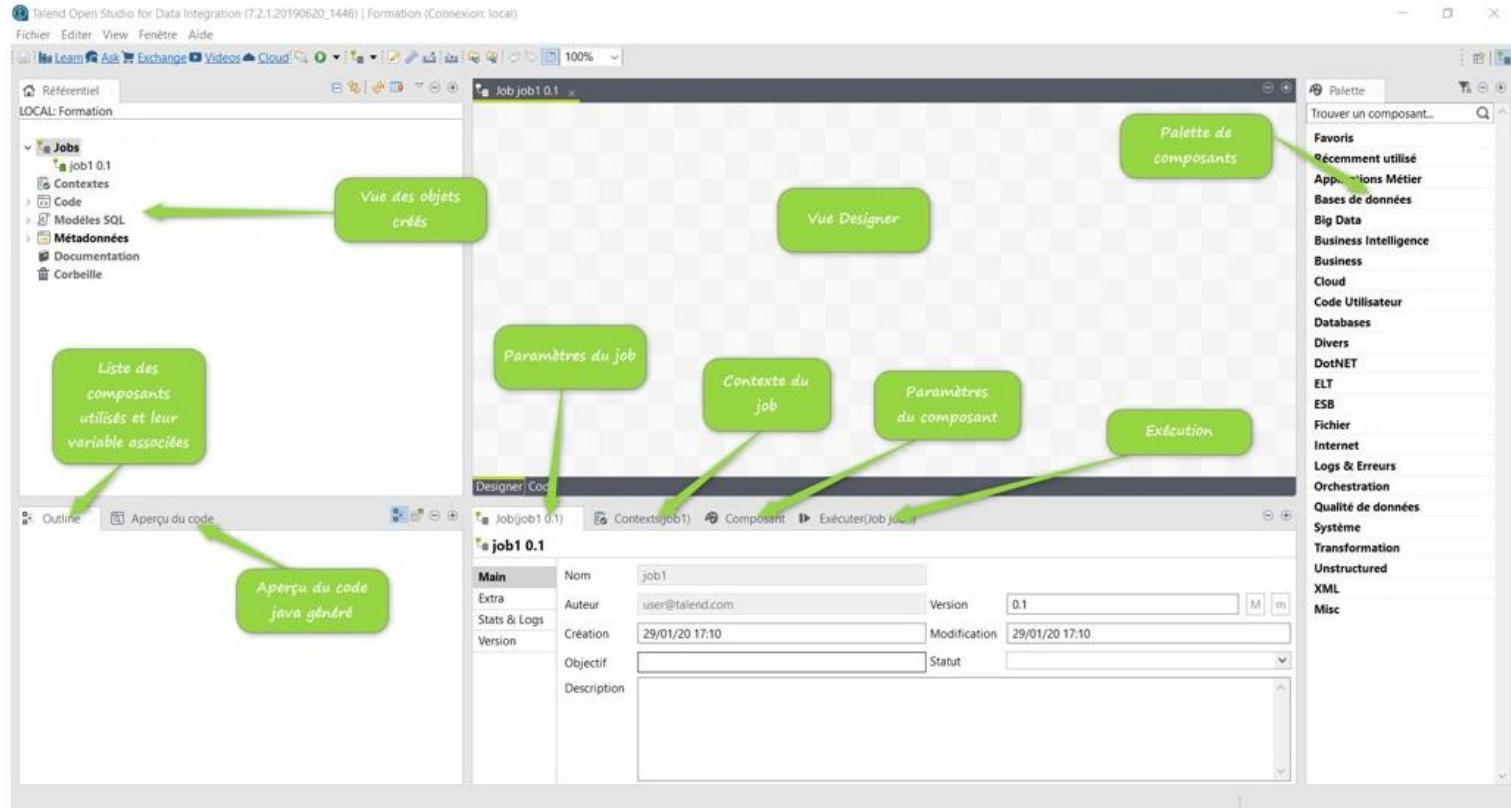
Password:

✓

Présentation du studio



Présentation du studio

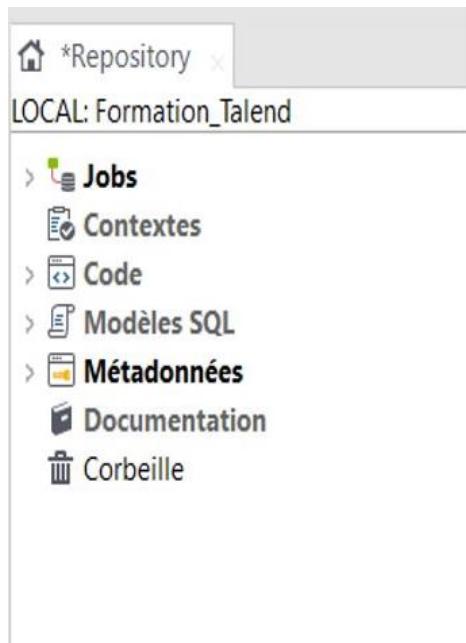


Concepts importants

- **Qu'est-ce qu'un référentiel ?** Un référentiel, repository en anglais, est un espace de stockage utilisé par **Talend Open Studio** pour regrouper toutes les données liées aux éléments techniques utilisés soit pour décrire les business models, soit pour créer les Jobs. **TOS** peut se connecter à autant de référentiel local que nécessaire.

- **Qu'est-ce qu'un projet ?** Un projet est un ensemble structuré d'éléments techniques et de leurs métadonnées associées. Ce qui signifie que tous les Jobs et business models que vous créez sont organisés en Projets. Vous pouvez créer autant de projets que vous le souhaitez dans un référentiel.

- **Qu'est-ce qu'un répertoire workspace ?** Un répertoire workspace est un répertoire dans lequel sont stockés tous les dossiers de vos projets. Vous devez avoir un répertoire workspace par connexion (connexion au référentiel). **TOS** vous permet de vous connecter à différents répertoires workspace, si vous ne souhaitez pas utiliser celui créé par défaut



Arborescence des objets

Jobs: Section qui contiendra tous les traitement (Job, Sous job, Master Jobs) que vous allez créer.

Contextes : Section qui va contenir vos <<variables>>.

Code : Section qui va contenir les routines Java que vous allez créer si besoin.

Modèles SQL :

Métadonnées : Section vous permettant de définir le schéma de vos tables, fichiers sources et cible ainsi que vos connexions BDD.

Documentation : Permet d'importer de la doc sur votre projet

Corbeille : ...

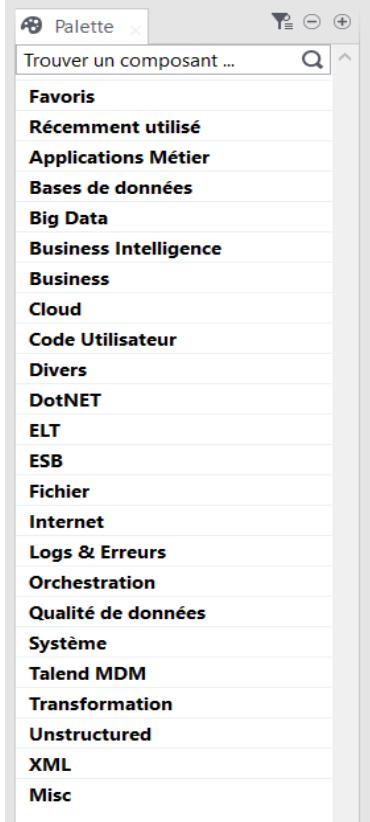
Vue Designer

**1. Onglet designer:**

Onglet qui contiendra tous les composants formant votre job

2. Onglet code:

Code java de votre job. Ne peut pas être modifié à la volée pour faire des modifications !



Palette de composants

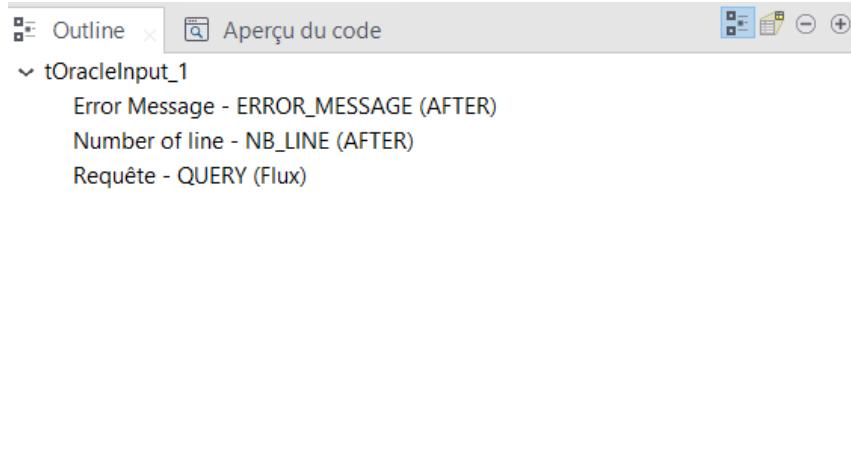
Il s'agit de la palette de composant, rangée par type de composant.

Voici quelques détails sur les types utilisés fréquemment :

- **Bases de données**: ces composants permettent d'ouvrir et fermer des connexions, récupérer des données sources, exécuter des commandes SQL et insérer/MAJ des données cible. Talend propose un large choix de base de données.
- **Fichiers**: permet la gestion de fichier (lecture/écriture etc)
- **Logs & erreur**: gestion des logs et des erreurs
- **Orchestration**: lister les fichiers d'un dossier, lanceur de job, boucles
- **Transformation**: agréger des lignes, joindre, trier, mapper etc

Vous pouvez aussi **envoyer des mails**, **générer des lignes**, **gérer le big data**, **attaquer le cloud (google drive, amazon, dropbox etc)**, **attaquer des FTP**, **pinger des serveurs** distants et plein d'autre chose

Vue Outline & Code



Il se compose de deux onglets:

- **Outline** : permet d'avoir une liste des composants utilisés dans le job et de récupérer certains des paramètres tel que : les erreurs, le nombre de ligne (lu, supprimé, maj, récupéré, etc) et pour certain composant la requête SQL utilisée.
- **Aperçu du code** : permet d'avoir un aperçu du code au fil de l'eau.

Vue Job

The screenshot shows the 'Job' view in Talend Studio. The top navigation bar includes tabs for 'Job(Connexion 0.1)', 'Contexts(Connexion)', 'Composant', and 'Exécuter (Job Connexion)'. The main area displays the 'Connexion 0.1' job details under the 'Main' tab. The job name is 'Connexion', created by 'user@talend.com' on '11/09/17 16:46', and has a version of '0.1'. The 'Description' field is empty. On the left, a sidebar lists other tabs: 'Extra', 'Stats & Logs', and 'Version'.

Cette interface possède 4 onglets:

- **Job** : permet de gérer les paramètres de votre job (versionning, date de création et MAJ, description, objectifs et logs)
- **Contexts** : permet d'importer et d'attribuer les contexts utiles pour ce job
- **Composant** : permet de paramétriser les composants de votre job
- **Exécuter** : permet l'exécution de votre job en mode normal et debug et visualiser les résultats dans la console

Vue Contextes

The screenshot shows the 'Contexts' tab in Talend Studio. The interface includes a toolbar with icons for new, delete, copy, move, and save. The main area displays a table with columns: Name, Type, Comment, and Default Value. A single row is present, labeled '1' and named 'chemin_fichier'. The 'Type' is set to 'String', and the 'Default Value' is set to '.../rcaron/talend'. At the bottom, there is a dropdown for 'Default context environment' set to 'Default'.

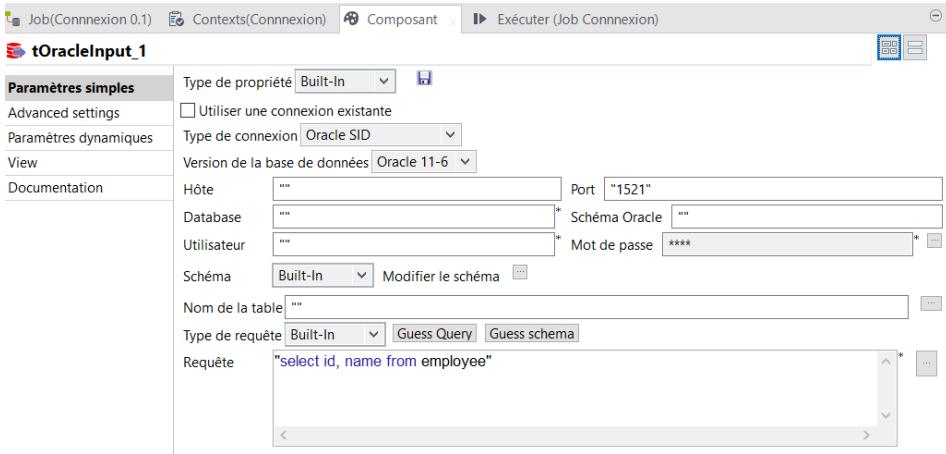
	Name	Type	Comment	Default Value
1	chemin_fichier	String		.../rcaron/talend

Cet onglet permet de gérer vos contextes et ses variables.

Par exemple ici, une variable <<chemin_fichier>> de type <<string>> avec pour valeur le chemin des fichiers plats que l'on pourrait utiliser dans nos jobs.

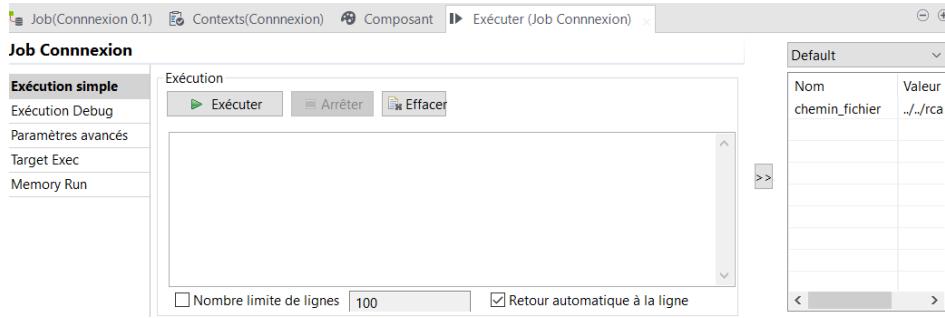
Ces variables peuvent être alimentées directement dans le traitement ou bien grâce au fichier de contexte défini ou grâce au groupe de contexte créé et importé. Enfin, vous pouvez choisir le contexte à utiliser (dev, prod, default, etc)

Vue Composants



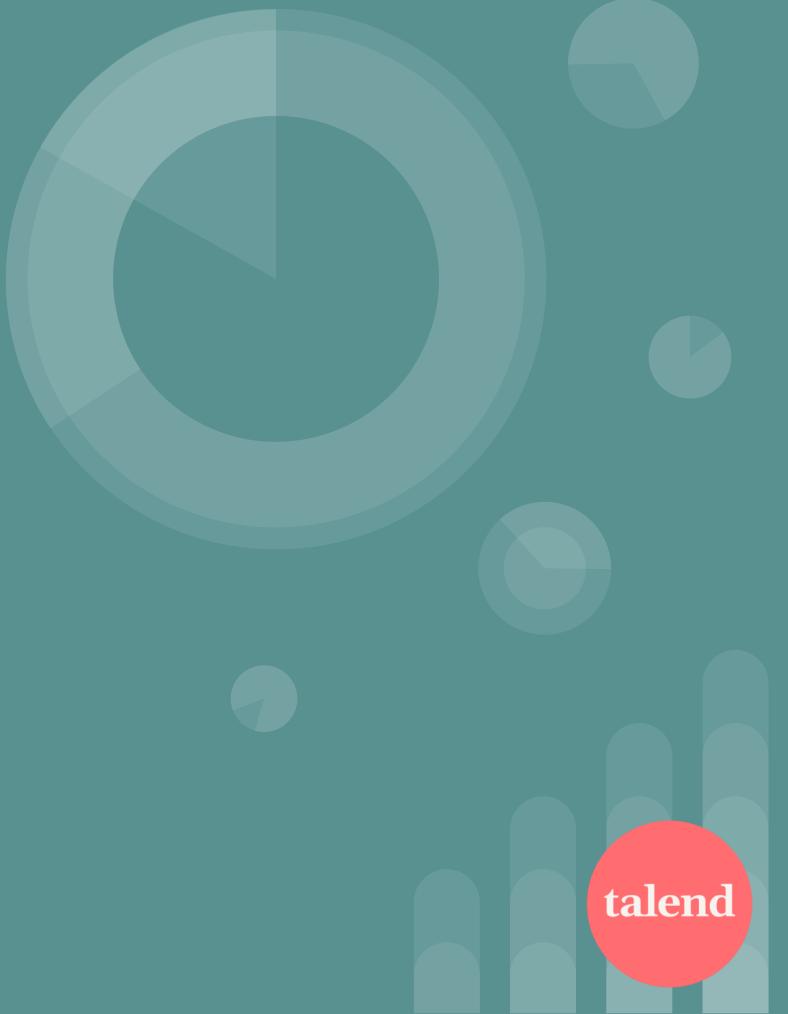
- En prenant l'exemple d'un tOracleInput permettant d'aller récupérer des données en base, vous pouvez ici paramétriser la connexion si elle n'existe pas, écrire la requête permettant de récupérer les données.
- Vous devez aussi décrire le schéma de la table si vous ne l'aviez pas inclus dans vos métadonnées.
- Les <>advanced settings<> permettent d'inclure un curseur, supprimer les espaces dans les string.
- Dans la partie <>view<> vous pouvez définir le nom du composant

Vue Exécuter

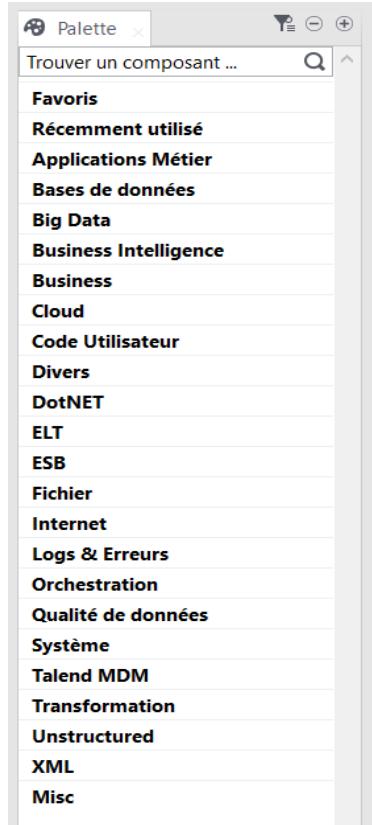


- Pour l'exécution, vous pouvez sélectionner le contexte dans lequel il faut lancer le traitement (dev, qual, prod etc)
- Vous pouvez aussi limiter le nombre de ligne d'exécution : cela vous permet de faire un test rapide avec les 100 premières lignes par exemple.
- Vous pouvez également lancer une exec en mode débogage, ce qui vous permet de voir comment réagis chaque composant.

Quelques Composants



Quelques composants



Comme vu précédemment, il existe une multitude de composants.
Chacun des composants est rangé en fonction de son domaine fonctionnel.
La nomenclature respecte dans la majorité des cas la règle suivante:

- **tDomaineFonction[Type]**
 - **t : Talend**
 - **Domaine : File / DB / XML etc**
 - **Fonction : Input / Output / Row etc**
 - **Type: Excel / Delimited etc**

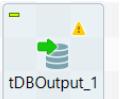
Quelques composants BDD



Composant permettant d'ouvrir une connexion à une base de données



Composant permettant de récupérer les données d'une table à l'aide d'une requête



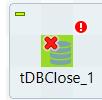
Composant permettant d'intégrer des données dans une table



Composant permettant de valider (commiter) les données dans une table



Composant permettant d'exécuter une requête type delete / truncate / etc



Composant permettant de fermer une connexion à une base de données

Quelques composants fichiers (type XML)

- **TFileInputXML:** lit un fichier XML structuré ou un flux de données et extrait les données ligne par ligne.



➤ Filename: pour indiquer le fichier structuré d'entrée.

➤ Loop XPath query: pour indiquer l'élément de boucle

➤ Mapping: pour renseigner les champs à extraire et à afficher dans la sortie

- **TFileOutputXML:** produit en sortie des données sous forme de fichier de type XML.



➤ Filename: chemin d'accès et nom de fichier de sortie

➤ Root tag : enveloppe les données et la structure entière de fichier de sortie

➤ Row tag: enveloppe les données et la structure de chaque ligne

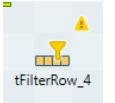
Quelques composants transformation



Le tMap permet comme son nom l'indique de faire du mapping entre les différentes entrées et sorties de données. Vous pouvez également joindre les données de plusieurs entrées, faire des filtres et des transformations de données.



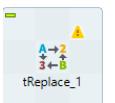
Le tUniqueRow permet de filtrer les doublons de lignes sur une ou plusieurs clés.



Le tFilterRow permet, comme son nom l'indique, de filtrer les lignes qui ne correspondent pas à certains critères définis par l'utilisateur.



Le tJoin permet comme son nom l'indique de joindre 2 flux de données.



Le tReplace permet de remplacer les valeurs d'une chaîne de caractère par une autre

tPreJob et tPostJob

- **tPreJob** : est un composant qui s'exécute avant tous les sous job d'un job. Il est important d'en avoir pour charger des informations nécessaires à l'exécution du job, les paramètres de connexion par exemple ou encore les informations sur les répertoires et même de vérifier l'existence d'un fichier.

- **tPostJob** : Lui par contre s'exécute à la fin des sous jobs. L'intérêt de l'avoir est qu'il s'exécute même s'il y a une erreur dans le job. Par ce biais, on peut débrancher des connexions, supprimer des fichiers temporaires, purger des répertoires, enregistrer le statut de certains opérations pour une réutilisation éventuelle.
A savoir que le tPostJob se déclenche également en cas d'erreur.


Les composants tJava



- **tJava** : Le composant tJava est généralement utilisé pour exécuter un morceau de code Java comme un sous-Job séparé. Normalement, le composant tJava n'a pas de flux de données d'entrée et de sortie.



- **tJavaRow** : Ce composant s'applique exclusivement à la partie principale du code généré du sous-Job et accède au flux d'entrée et transforme les données. Le code Java inséré via le tJavaRow est exécuté pour chaque ligne.



- **TJavaFlex** : Les parties de début et de fin seront exécutées seulement une fois dans le sous-Job. La partie principale sera exécutée pour chaque ligne. Les données source sont traitées à l'exécution par le tJavaFlex. Il s'agit du meilleur composant à utiliser lors de l'initialisation des opérations au début du sous-Job ou du traitement des opérations à la fin du sous-Job.
- Le tJavaRow et le tJavaFlex sont généralement utilisés pour accéder au flux d'entrée et modifient les données.



Bienvenue sur Talend Help Center

Comment pouvons-nous vous aider ?

Recherchez dans la base de connaissances et la documentation Talend.

Français ▾ 

▼ Produit ▼ Module ▼ Version ▼ Tâche

Vous trouverez le détail et liste de chaque composant et fonctionnalité sur l'aide en ligne de talend à l'adresse suivante:

<https://help.talend.com/>

Fonctions Internes



Les fonctions internes

Catégorie	Fonctions
*Défini par l'utilisateur	checkCDATAForXML(String input) : id_String - TalendString
*Tous	getAsciiRandomString(int length) : id_String - TalendString
Data Masking	removeAccents(String text) : id_String - TalendString
DataOperation	replaceSpecialCharForXML(String input) : id_String - TalendString
Data Quality	talendTrim(String origin, char padding_char, int align) : id_String - TalendString
Mathematical	unionString() : id_String - TalendString
Numeric	
Relational	
SQLike	
StringHandling	
TalendDateGenerator	
TalendDate	
TalendString	
User Defined	

Les fonctions natives de Talend sont rangées dans plusieurs catégories (Écran de gauche).

Le détails des fonctions disponibles se trouve sur l'écran de droite.

Talend offre de nombreuses fonctions, ce qui permet au développeur d'éviter le code “maison”.

Vous avez des fonctions pour les dates, les chaînes de caractères, les calculs, les relations (“=”, “>”, “<”, etc) et bien d’autres encore.

Vous pouvez aussi écrire une fonction vous-même, c'est ce que l'on appelle une routine. Nous y reviendrons plus tard.

Acssi

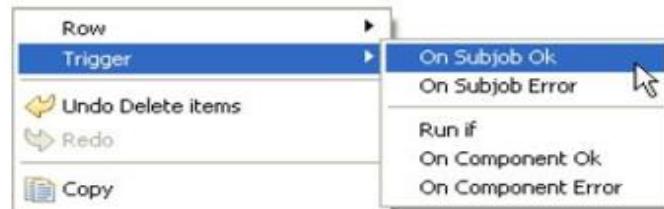
DATA
MANAGEMENT
ERP

précisions



Types de liens

- il existe divers types de liens qui définissent soit la donnée à traiter soit l'enchainement logique du Job. Dans l'éditeur graphique, lorsque vous glissez le pointeur du composant source vers le composant cible, une prise électrique symbolise le branchement en cours et un signe *interdit* indique que le composant cible n'est pas valide. Le signe interdit ne disparaît qu'une fois le composant valide atteint. Seules les connexions autorisées pour le composant sélectionné dans l'éditeur graphique sont proposées dans la liste du menu contextuel.



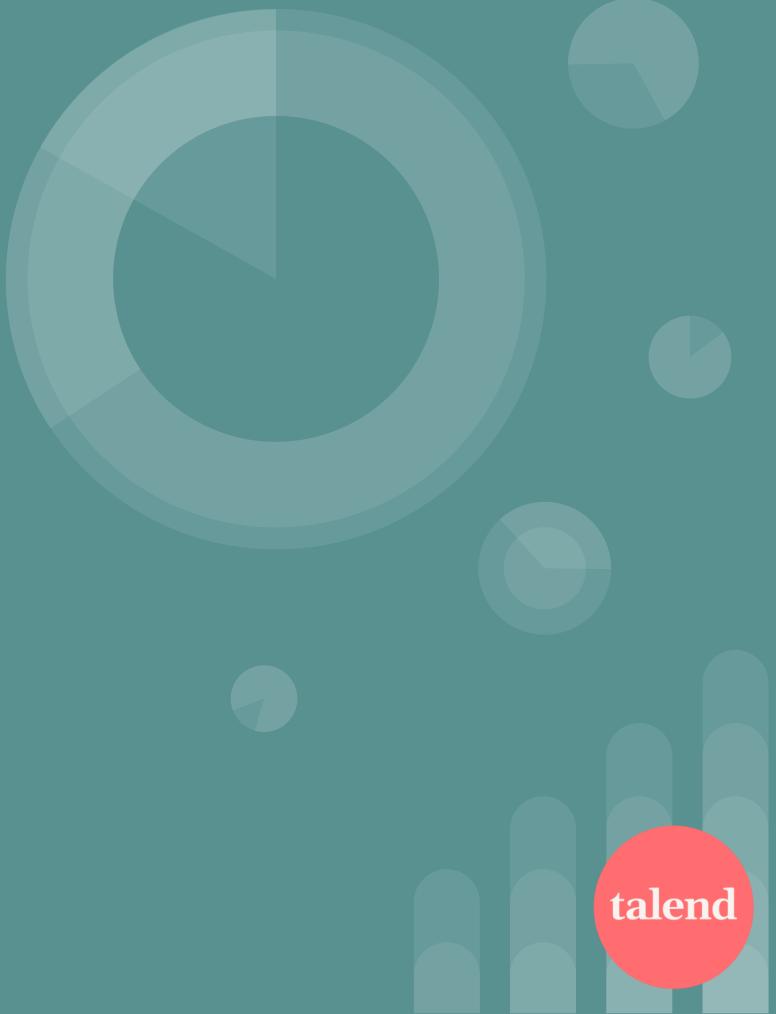
Connexion de type Row

- La connexion de type **Row** manipule les données elles-mêmes. Les liens de type Row sont **main**, **lookup**, **reject** ou **output** selon la nature du flux de données traité
- La connexion Row de type **Main** est la connexion la plus courante. Elle transmet les flux de données d'un composant à l'autre, en faisant une boucle sur chacune des lignes pour lire ou extraire les données selon la définition des propriétés du composant. Les données transférées à travers les connexions de type Row sont caractérisées par une définition du schéma qui décrit la structure des données dans le fichier d'entrée.
- La connexion de type **Lookup** est une connexion Row reliant un composant d'un flux secondaire à un composant d'un flux principal (ce composant doit être capable de recevoir plus d'un flux d'entrée).
- La connexion de type **Iterate** sert à faire une boucle sur plusieurs fichiers d'un répertoire donné, sur les lignes d'un fichier ou sur les entrées d'une base de données.
- La connexion de type **Output** est une connexion Row traitant de la donnée. Ce lien relie un composant **tMap** à un ou plusieurs composants de sortie. Les sorties d'un Job pouvant être multiples, une boîte de dialogue s'affiche pour vous demander de nommer chacune des connexions.

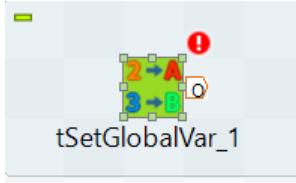
Connexion de type Trigger

- **OnSubjobOK** : Cette connexion est utilisée pour déclencher le sous-job qui suit à condition que le sous-job principal se soit exécuté sans erreur. Cette connexion s'utilise uniquement avec le composant de début (Start) de votre Job.
- **OnSubjobError** : cas inverse
- **OnComponentOK** ne déclenche l'exécution du composant cible qu'une fois l'exécution du composant source terminée correctement. Il peut servir à déclencher un sous-job de notification par exemple.
- **OnComponentError** : cas inverse
- **Run if** déclenche un sous-job ou un composant si les conditions définies sont réunies. Cliquez sur la connexion pour afficher les propriétés de votre connexion **If**, puis définissez vos conditions en Perl ou en Java en fonction du langage de génération de code choisi pour votre projet. Le raccourci clavier **Ctrl+Barre d'espace** vous donne accès à toutes les variables globales et de contexte.

Variables



Les variables globales



Fonction:

Le composant tSetGlobalVar permet de créer et mettre à jour des variables. Ces variables ne sont valables que pour le job dans lequel elles sont déclarées.

A screenshot of the configuration screen for the 'tSetGlobalVar_1' component. On the left, a sidebar shows 'Paramètres simples' and links to 'Advanced settings', 'Paramètres dynamiques', 'View', and 'Documentation'. The main area is titled 'Variables' and contains a table with two columns: 'Clé' (Key) and 'Valeur' (Value). There is one row in the table with the key 'myKey' and the value 'myValue'. At the bottom of the table are several icons for managing rows: a plus sign for adding, a minus sign for deleting, and arrows for moving rows up or down. To the right of the table are two small icons: a grid and a double arrow.

Paramétrage:

- Définir une variable
- Définir sa valeur

Pour l'appeler:

`((String)globalMap.get("myKey"))`

Il existe un autre type de variable : les variables de contextes.

Les variables de contexte

Screenshot of the Talend Studio interface showing the 'Contexts' component for a job named 'MonPremierJob'. The table lists one variable: 'new1' of type 'String' with a default value. Red annotations explain the steps: 'Ajouter une variable' points to the '+' icon; 'Nommer la variable' points to the 'Name' column; 'Typer la variable' points to the 'Type' column; and 'Valoriser la variable' points to the 'Default Value' column.

	Name	Type	Comment	Default Value
1	new1	String		

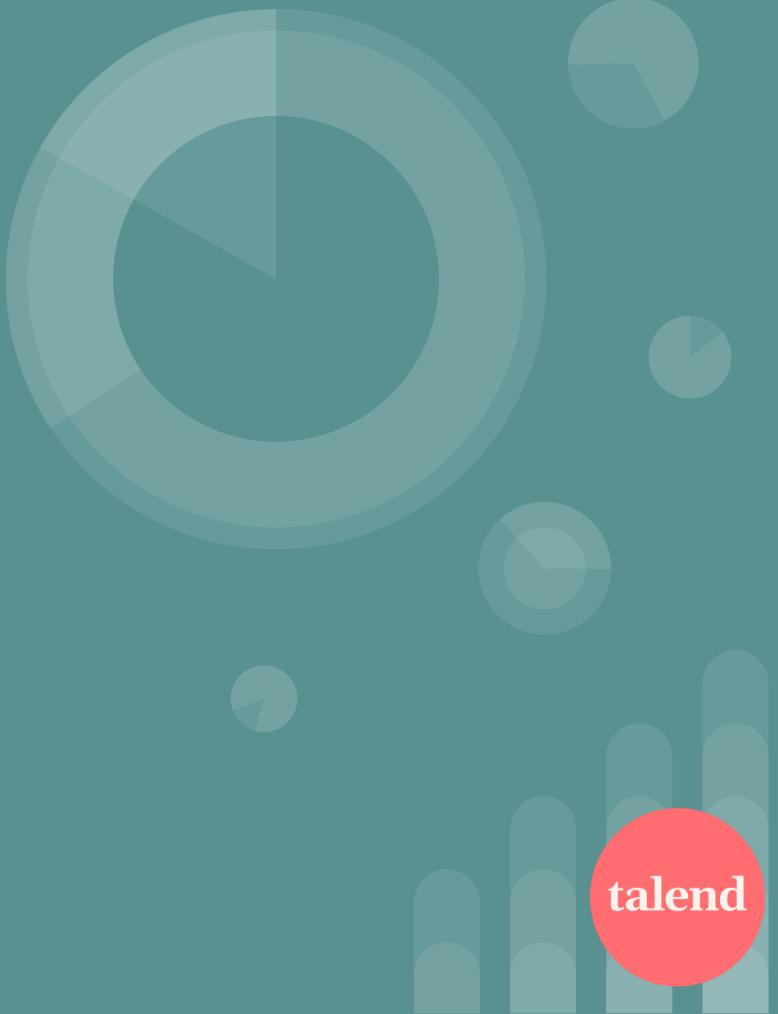
Ajouter une variable Nommer la variable Typer la variable Valoriser la variable

Default context environment Default

Toutes les variables de contexte déclarées dans un job sont utilisable uniquement dans celui-ci. Pour l'appeler ou même lui assigner une nouvelle valeur, on utilise la syntaxe suivante :

context.NomVariable (Appel)
context.NomVariable = "MaValeur" (Valoriser)

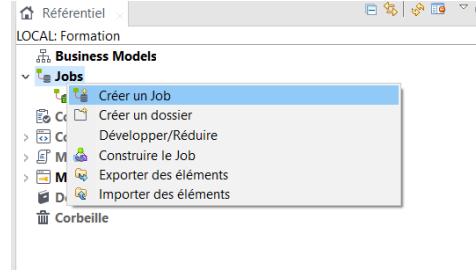
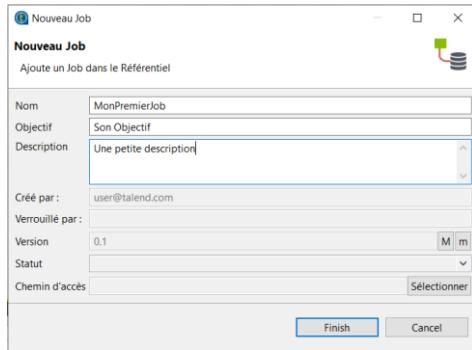
Création d'un job



talend

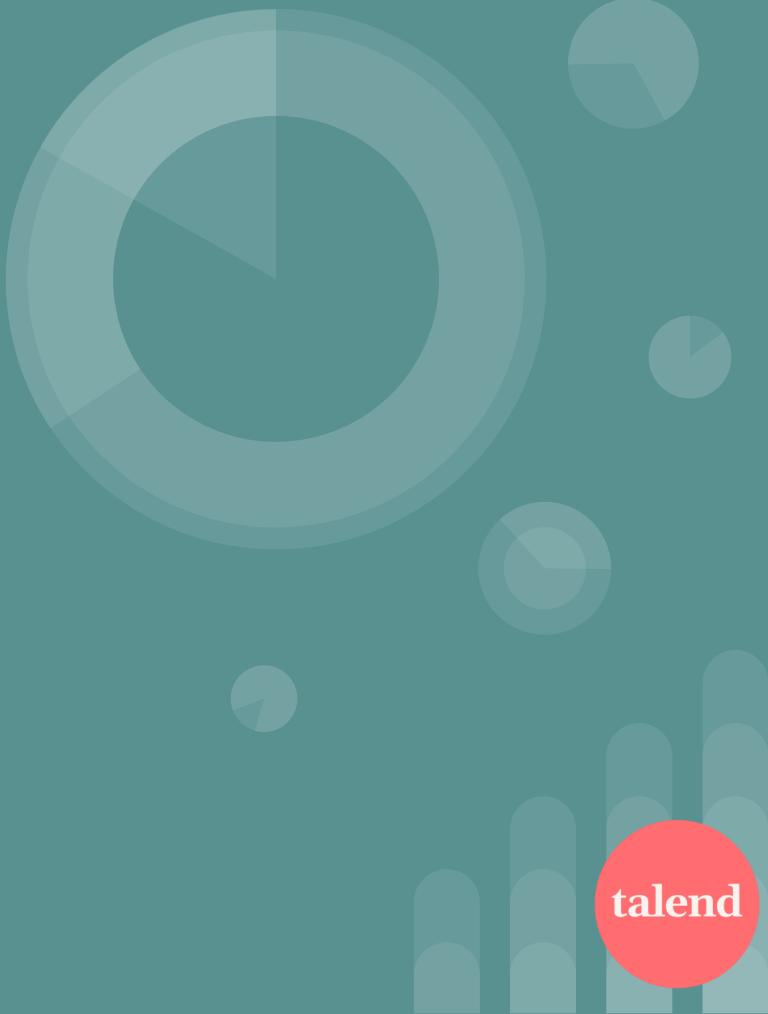
Créer un job

1 - Clic droit sur job > Créer un job



2 - Renseignez les champs et finish

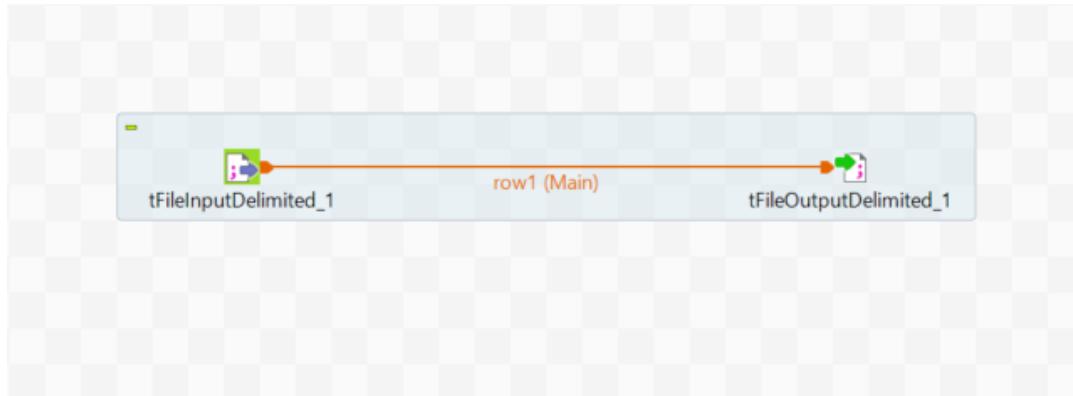
Quelques applications



Exercice 1

Nous allons ici commencer simplement.

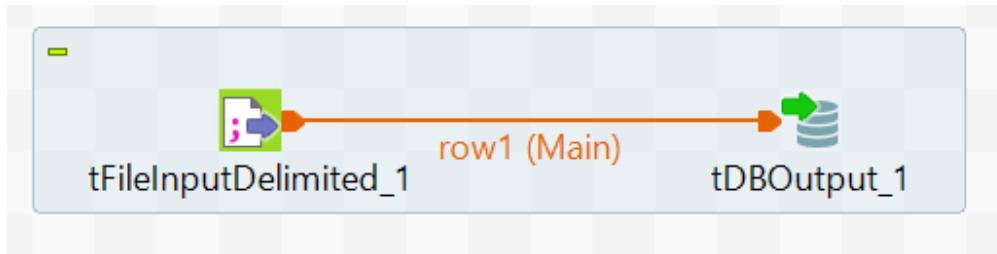
Notre premier job aura pour but de copier le fichier Clients.csv présent dans le dossier exo1/input.



Il nous faut ensuite configurer les composants.

Exercice 2

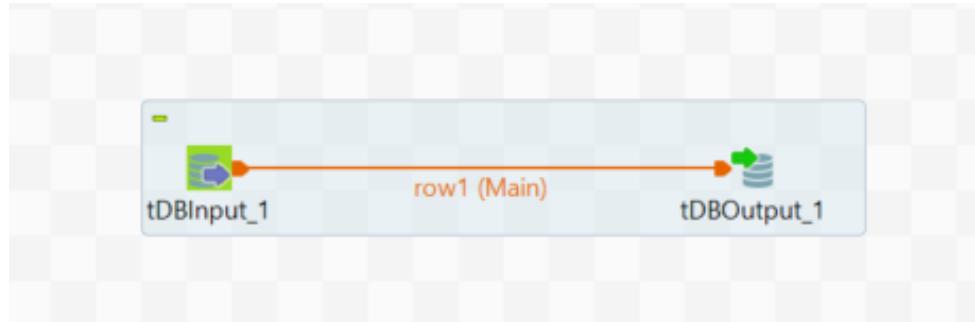
À l'image de l'exercice précédent, nous allons charger ici les données du fichier chiens mais dans une table chiens de notre base de donnée.



Exercice 3

Nous allons maintenant faire un job qui servira à copier les données de la table chiens et les stocker dans une autre table.

Les composants à utiliser se trouvent donc dans la rubrique bases de données



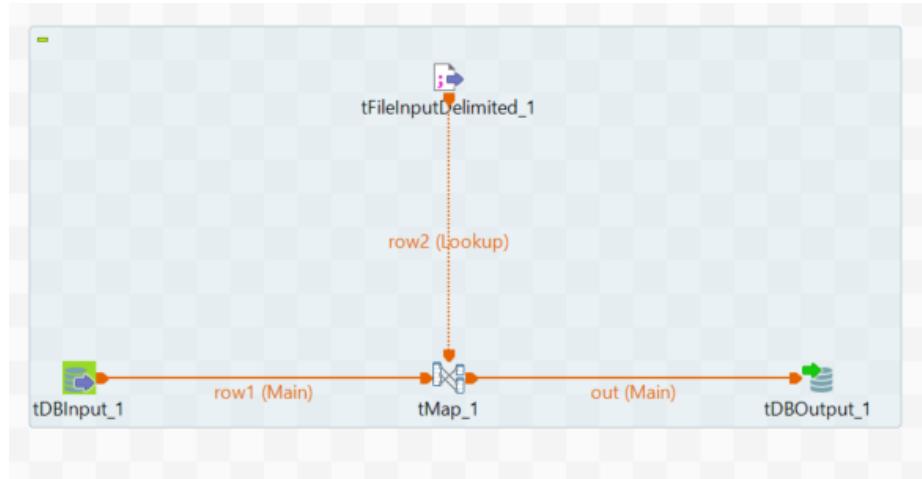
Exercice 4

Combinons maintenant les trois premiers exercices.

Maintenant que nous sommes capable de lire et écrire dans des fichiers/ bases de données.

Le but de cet exercice sera de lire le fichier Client.csv, lire la table chiens et de les regrouper dans une seule table.

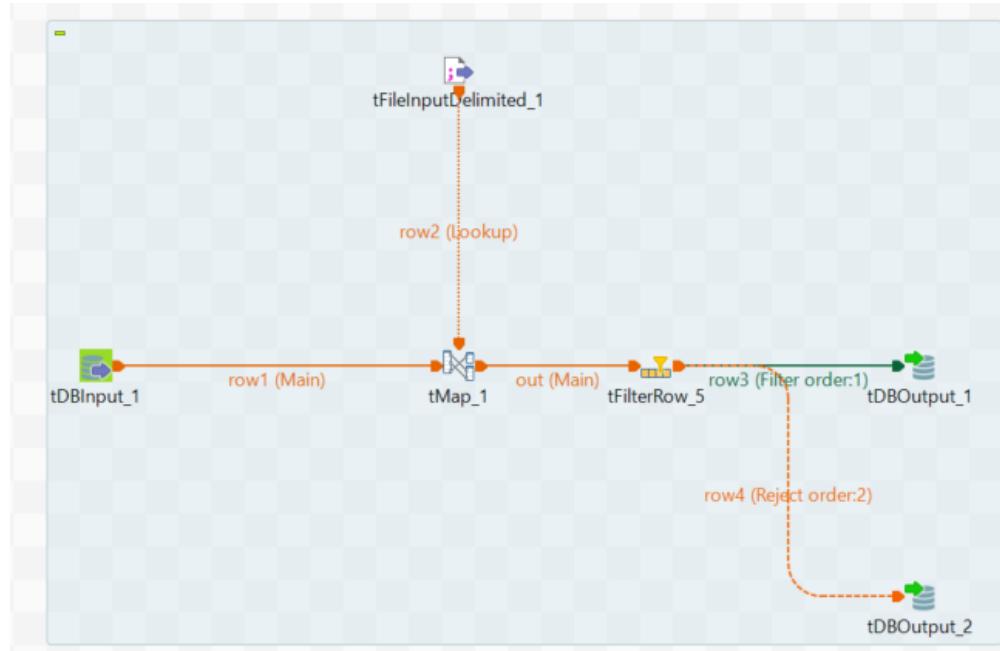
Il existe plusieurs façons de répondre à cette demande, nous verrons comment utiliser le composant polyvalent tMap pour faire cette jointure.



Exercice 5

Complexifions maintenant l'exercice 4 en filtrant par rapport à l'âge du chien.

Nous allons ici regrouper les jeunes chiens ($\text{Age_chien} < 7$) et les vieux chiens ($\text{Age_chien} \geq 7$) dans deux tables différentes.



Exercice 6

Pour cet exercice, nous allons créer un job plus complexe.

Nous allons dans un premier temps vérifier si le fichier Clients.csv existe dans le dossier exo6/input.

- Si il est absent nous allons arrêter le job en renvoyant un message d'erreur.
- Si il est présent, il nous faudra trier le fichier par ID client et enregistrer le résultat dans le dossier exo6/output. Une fois fait, nous devrons déplacer le fichier du dossier input vers le dossier archive.

Le job devenant plus complexe, nous devons commencer à faire attention à sa lisibilité.

À vous de jouer !

Exercice 7

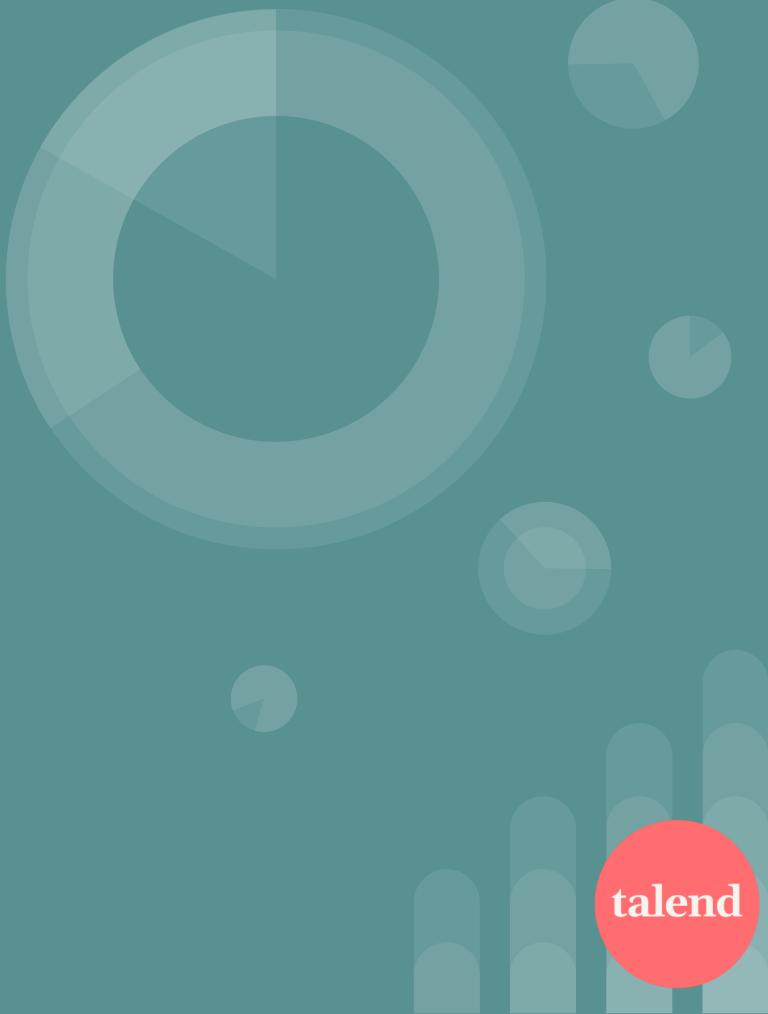
Nous terminerons ces petits exercices sur un job ayant pour but de modifier les données.

L'objectif est de lire la table chiens et de:

- Transformer la colonne Pelage_Chien en ne conservant que les deux premières lettres
 - Ajouter une colonne avec pour valeur "Jeune" si le chien à moins de 7ans et "Vieux" sinon
- Le résultat pourra être enregistré dans une table.

C'est à vous !

Bonnes pratiques



Bonnes pratiques

- Plus un projet Talend est structuré, meilleure la compréhension et la maintenance seront ! On peut inclure les jobs dans des dossiers et sous dossiers. Mais attention à ne pas en faire de trop !
- Bien réfléchir en amont à la manière dont vous allez gérer vos paramètres (BDD, fichier plat etc...). Il n'est pas possible de tout mettre en base comme il est peu recommandé de tout mettre dans un fichier plat !
- Ne pas mettre un traitement complet (Extraction / Transformation / Insertion ...) dans un même job ! Cela peut causer des problèmes de performance, de maintenabilité et de lisibilité.
- Définissez dès le départ une manière de construire vos traitements: je mets les composants de haut en bas ou de gauche à droite.
- Documenter vos jobs : une description du job simplifie la compréhension et la maintenabilité pour vous et les prochains développeurs. Bien nommer les composants est aussi agréable pour ceux qui prendront le relais.
- Sécuriser les requêtes (SQL...) dans les connexions pour empêcher toutes les modifications accidentnelles.
- Pour le nommage des jobs, c'est en fonction des règles au sein du projet. Mais voici un exemple:
 - MJ_Source_Cible_Libre
 - SJ_Source/Cible_Action_Libre
 - Source/Cible: Source et cible de données
 - Action : Extraction, transformation, Insertion
 - Libre: ce que vous voulez
- Ces règles de nommages ne sont pas exhaustives, elles peuvent varier selon les types de projets

Quelques notions

Job : Objet Talend permettant de faire un ou plusieurs traitements. Il existe plusieurs types :

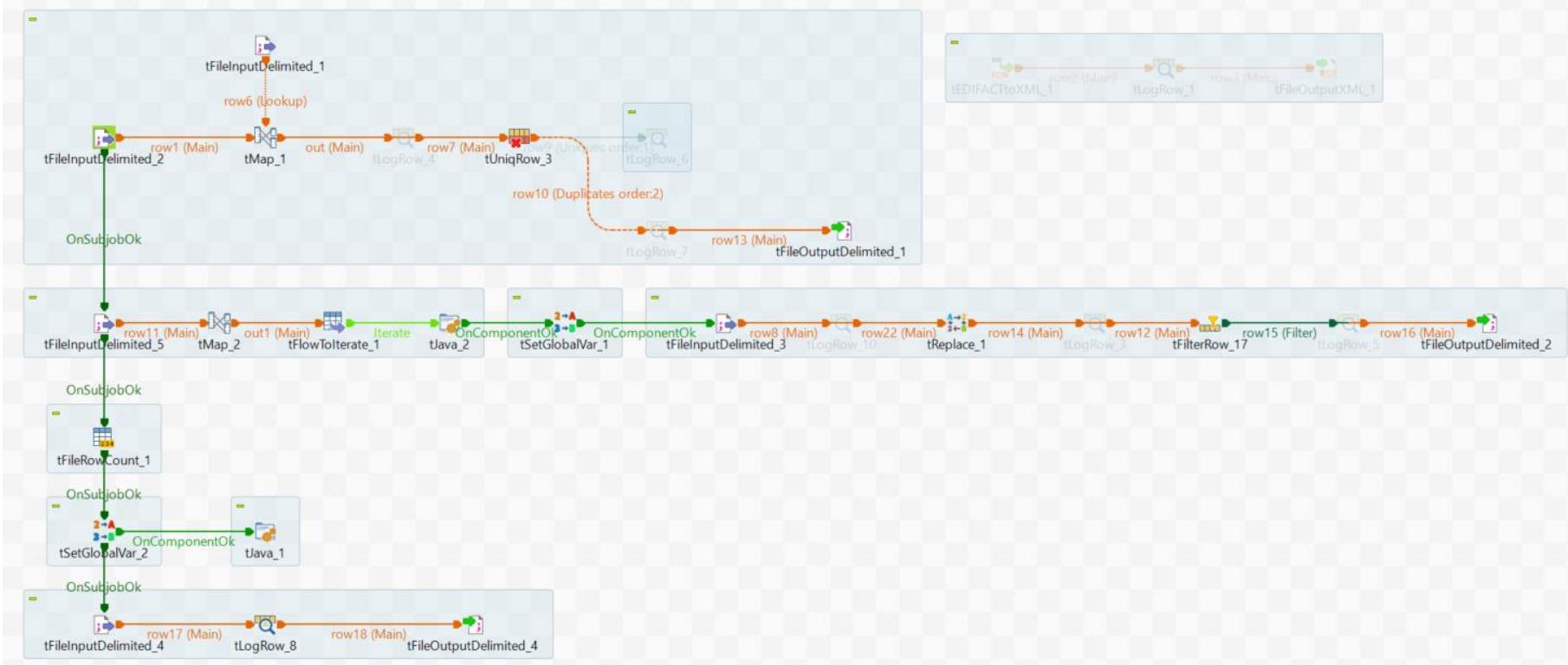
- **Master Job** : Ou Job Maître dans sa traduction, aussi appelé job père. Il permet généralement de lancer plusieurs sous traitements, créer et gérer différents paramètres (Variable(s), context(s)). Il sert parfois de mini ordonnanceur.
- **Sous Job** : Ou job fils. Il contiendra le/les traitements que vous voudrez mettre en place.
- **Pre-job** : composant Talend qui exécutera un «pré-traitement» avant le traitement principal.(ex: connexion au BDD, récupération du fichier de contexte etc) . Ce composant est conseillé dans la majorité des jobs.
- **Post-Job** : composant Talend qui exécutera un «post-traitement» à la fin du traitement principal.(ex: fermeture de connexion BDD, logs etc)

Le master et le sous job sont défini par le développeur à travers le nom : MJ_**** pour les MasterJob et SJ_**** pour les sous jobs. En effet, il n'existe pas de type de job prédéfini master ou sous job.

Contexte : Ensemble de paramètres/variables généraux définis pour le projet (ex : chemin vers fichiers, paramètres de connexion etc) . Ils peuvent être valorisés au fil de l'eau et/ou par un **fichier de contexte**

Fichier de contexte : Fichier plat contenant différents paramètres et leurs valeurs. Généralement, il existe 1 fichier de contexte par environnement (Dev, Recette, Prod ...) .

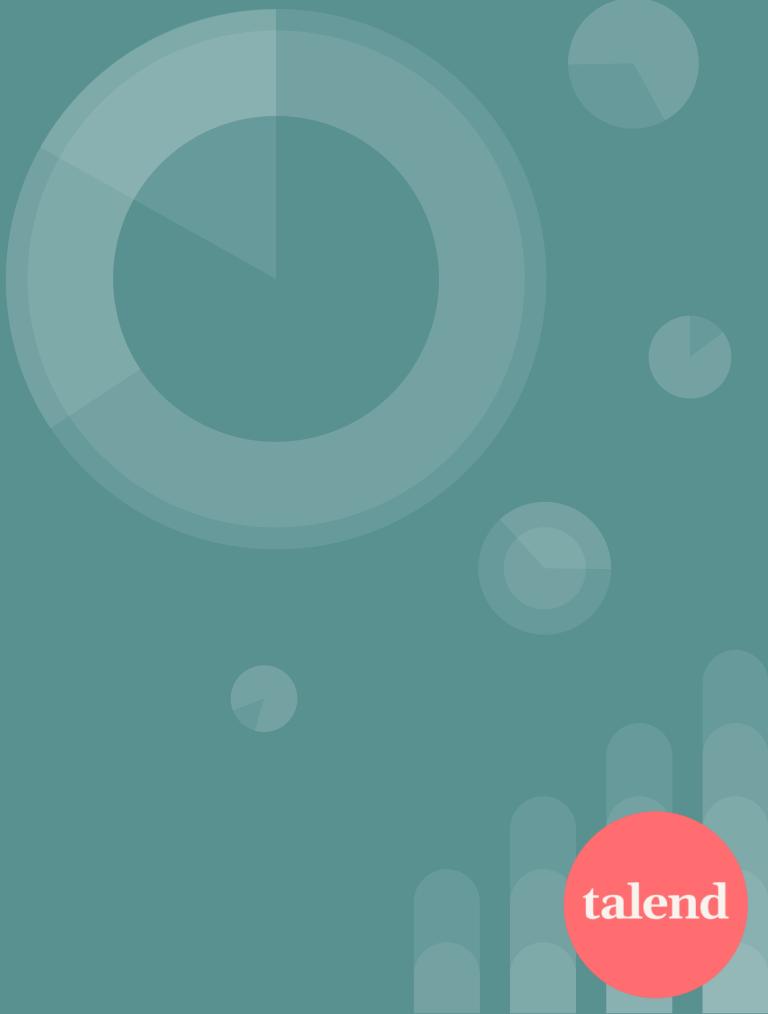
Exemple de flux : flux/circulation des données de la gauche vers la droite et du bas vers le haut



ACSSI

DATA
MANAGEMENT
ERP

Code & Routines

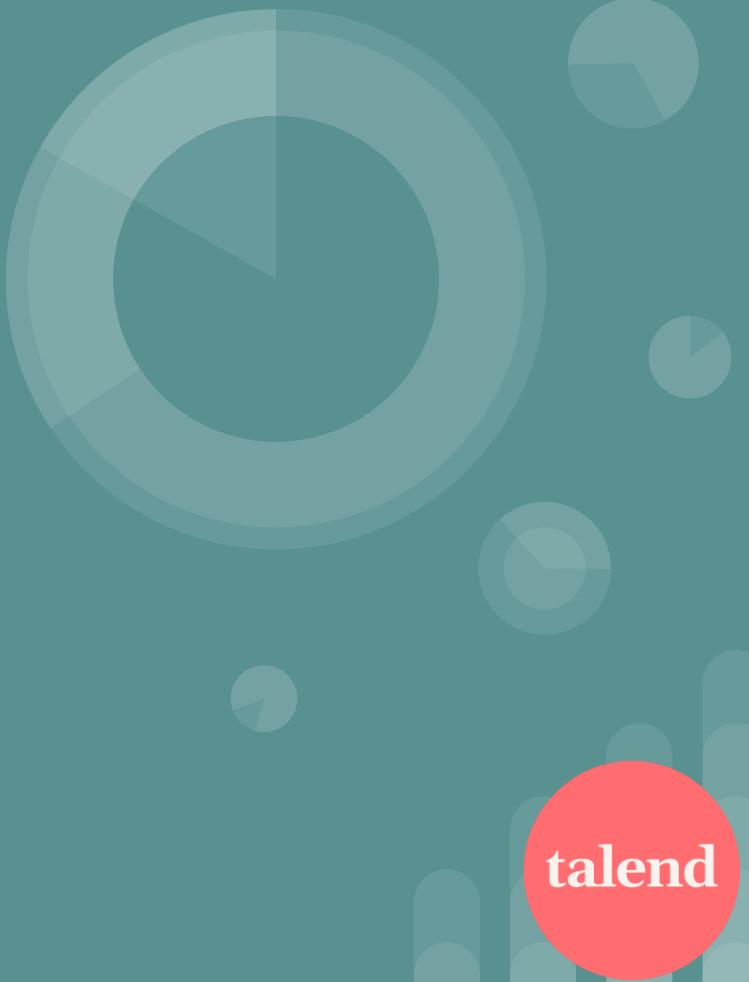


talend

routine

- Les routines sont des fonctions Java plus ou moins complexes, généralement utilisées pour factoriser du code. Elles permettent d'étendre les possibilités de traitement des données dans un ou plusieurs Job(s) technique(s).

Métadonnées



Importer un groupe dans un job

The screenshot shows the Talend Data Management Platform's repository browser. The left sidebar lists various connection types and their configurations:

- Modèles SQL
- Métadonnées
 - Connexions aux bases de données
 - MysqlConnection 0.1
 - Requêtes
 - Schémas des tables
 - Divisions
 - Paramétrage
 - Schémas des vues
 - Schémas synonymes
 - Fichier délimité
 - Fichier positionnel
 - Fichier regex
 - Fichier XML
 - Fichier Excel
 - Faits
 - Commande_Details 0.1
 - Commande_Entete 0.1
 - Référentiel
 - Articles 0.1
 - Clients 0.1
 - Client_xlsx
 - Colonnes(11)

Fonctions:

Les métadonnées permettent d'enregistrer des fichiers et des tables de BDD avec des schémas spécifiques.

Ils vous permettent aussi de créer des connexions BDD pour ne pas avoir à les renseigner plusieurs fois dans vos dev.

Ceci va vous permettre de récupérer les schémas dont vous avez besoin sans avoir à les écrire à chaque fois !

De plus, cela vous permettra si besoin est, de faire une modification à un seul endroit au lieu d'aller retrouver et modifier chaque traitement.

Dans l'exemple ci-contre, j'ai importé tous mes fichiers excel en tant que métadonnées ainsi que ma connexion Mysql.

J'ai ensuite pu utiliser cette connexion en tant que contexte. (ci-dessous).

Vous voyez aussi que tout est en mode référentiel, ce qui permet d'aller re-rechercher nos métadonnées.

The screenshot shows the Talend Job Designer with a configuration window for the **tFileInputExcel_1** component. The configuration pane is divided into several sections:

- Paramètres simples**:
 - Type de propriété: Référentiel, set to EXCEL:Clients.
 - Lire un fichier au format excel2007 (xlsx) checked.
 - Nom de fichier/Flux: "C:/Users/récaron/Desktop/TOS_BD-20170623_1246-V6.4.1/workspace/in/Clients.xlsx".
 - Toutes les feuilles checked.
 - En-tête: 1.
 - Affecte chaque feuille (en-tête et pied de page) checked.
 - Arrête en cas d'erreur checked.
 - Première colonne: 1.
 - Dernière colonne: 1.
 - Schéma: Référentiel, set to EXCEL:Clients - Client_xlsx.
- Advanced settings**
- Paramètres dynamiques**
- View**
- Documentation**

Acssi

DATA
MANAGEMENT
ERP

Compilation



talend

Compilation

Votre job est terminé, mais maintenant qu'est ce que vous en faites ?

Vous avez 2 possibilités:

1. Vous exporter les sources pour que quelqu'un puisse les récupérer et les exploiter (si vous n'avez pas de projet collaboratif)
2. Vous construisez et exporter vos jobs pour les déposer sur le serveur afin de les lancer.

Nous allons voir la 2ème solution, qui semble la plus intéressante.

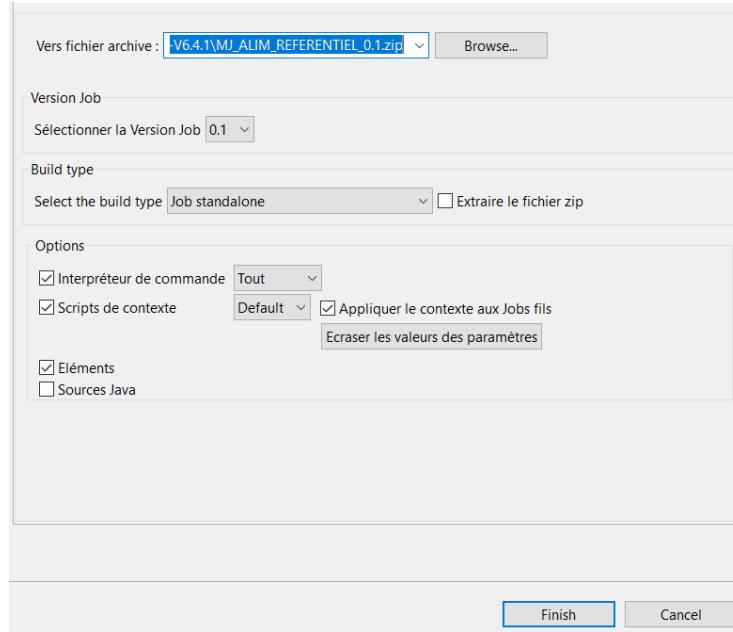
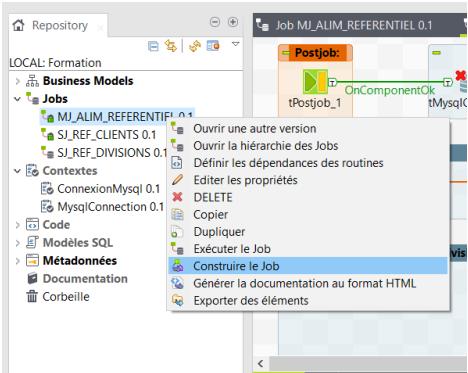
Vous n'exportez sur les serveurs que les jobs indépendants. C'est à dire les master job et les jobs indépendants. En effet, en exportant uniquement le master job, Talend va vous construire un paquet composé du master job et de tous ses jobs fils sous forme de process.

Talend exporte et construit ses jobs au format compressé (.zip).

Il vous est possible de sélectionner différentes options d'export.

Importer un groupe dans un job

Clic droit -> Construire le job



1. Sélectionner l'endroit où vous souhaitez exporter.
2. Sélectionner la version du job
3. Sélectionner le type de build
 - a. Job standalone
 - b. Service Web Axis (ZIP)
 - c. Service Web Axis (WAR)
 - d. Bindle OSGI pour ESB
4. Choisissez si vous souhaitez extraire le fichier zip ou non
5. Sélectionner dans quel interpréteur de commande vous souhaitez lancer vos job (Unix / Windows)
6. Choisissez d'inclure ou non les contextes. Si oui lequel. Choisissez aussi d'appliquer le contexte aux job fils.
7. Choisissez d'exporter les éléments (dépendance, librairie etc)
8. Choisissez d'exporter les sources java.
9. Finish

Cas pratique

[TP1 - Talend Basics.pptx](#)

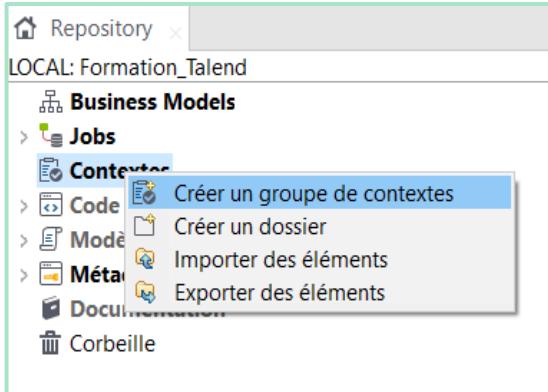


talend

Contexts

Le but ce chapitre est de nous apprendre à créer un context ou un groupe de context en mode built in. On va également apprendre à récupérer des variables d'un fichier de configuration externe et de s'en servir comme context.

Les groupes de contextes



Faire un clic droit sur <<contextes>> dans le menu à gauche, créer un groupe de contexte.

Etape 1:

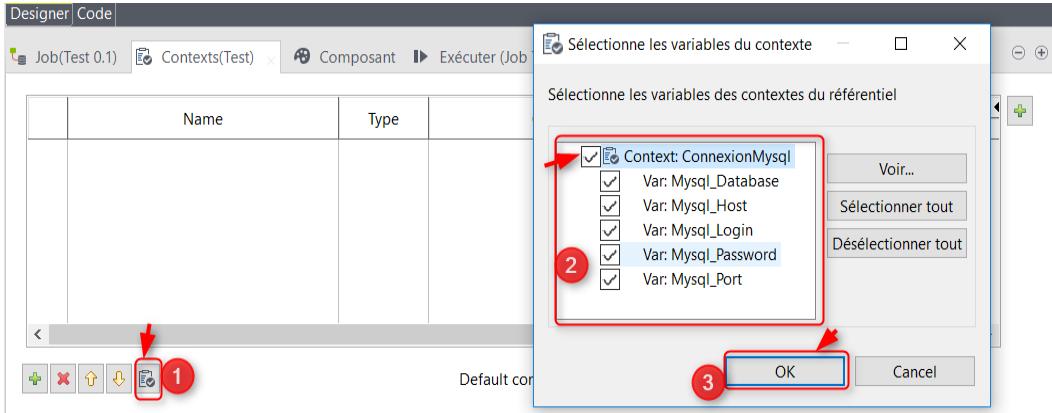
Donner un nom à votre groupe et une description. Suivant.

This is a screenshot of the 'Créer / Editer un groupe de contextes' dialog for 'Etape 1 sur 2'. It shows fields for 'Nom' (ConnexionMysql), 'Objectif' (empty), and 'Description' (information sur la/les connexions aux bases de données Mysql). A red box highlights the 'Nom' field (1), another highlights the 'Description' field (2), and a red arrow points to the 'Next >' button at the bottom (3).

This is a screenshot of the 'Créer / Editer un groupe de contextes' dialog for 'Etape 2 sur 2'. It shows a table of context variables: Mysql.Host, Mysql_Port, Mysql_Login, Mysql_Password, and Mysql_Database, all set to String type. A red box highlights the 'Default Value' column (2). Below the table, a red box highlights the 'Default context environment' dropdown (1) and a red arrow points to the 'Finish' button at the bottom right (3).

Etape 2: Ajouter des variables avec une description courte. Ne renseignez pas de valeur, ce sera fait dynamiquement par la suite. Finish.

Importer un groupe dans un job



1. Cliquer sur l'icône en bas à droite
2. Sélectionner les variables de contexte à importer ou sélectionner tout.
3. OK

Votre contexte <>ConnexionMysql<> a été ajouté au contexte du job. Vous pouvez utiliser ses variables dans votre job.

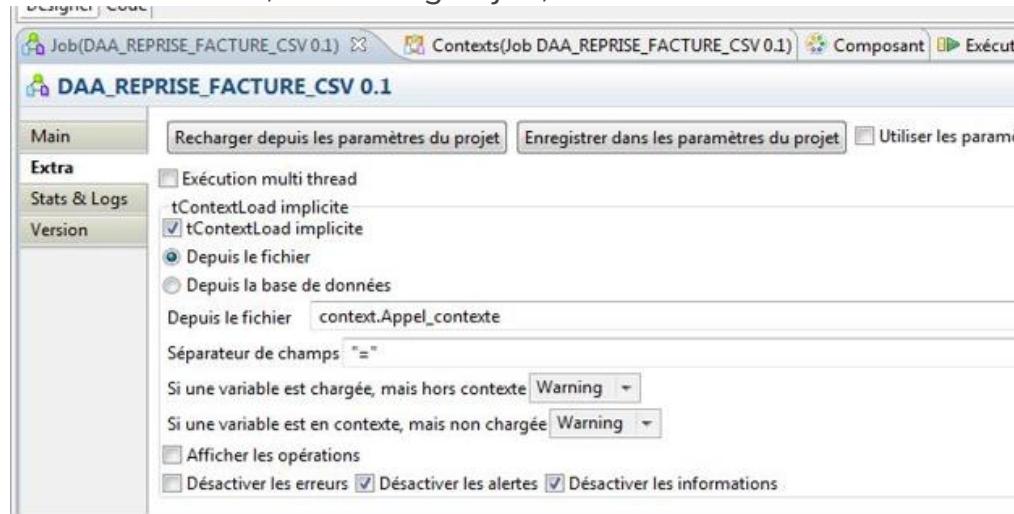
The screenshot shows the Talend Designer interface with the 'Designer' tab selected. The 'Contexts(Test)' tab is active. A context environment table is displayed, listing variables imported from 'ConnexionMysql'. The table has columns: Name, Type, Comment, and Default Value. The variables listed are: 'Chemin Fichier' (String, Default Value: D:\Talend), 'ConnexionMysql (from repository context)' (String, Default Value: ConnexionMysql), 'Mysql_Database' (String, Default Value: Nom de la base), 'Mysql_Host' (String, Default Value: HostName), 'Mysql_Login' (String, Default Value: Login de connexion), 'Mysql_Password' (String, Default Value: Mot de passe de connexion), and 'Mysql_Port' (String, Default Value: Numéro de port). Three numbered arrows point to specific elements: arrow 1 points to the context icon in the toolbar; arrow 2 points to the table header; and arrow 3 points to the 'OK' button in the bottom left corner of the table area.

Name	Type	Comment	Default Value
			Value
Chemin Fichier	String	Chemin du fichier	D:\Talend
ConnexionMysql (from repository context)	String		
Mysql_Database	String	Nom de la base	
Mysql_Host	String	HostName	
Mysql_Login	String	Login de connexion	
Mysql_Password	String	Mot de passe de connexion	
Mysql_Port	String	Numéro de port	

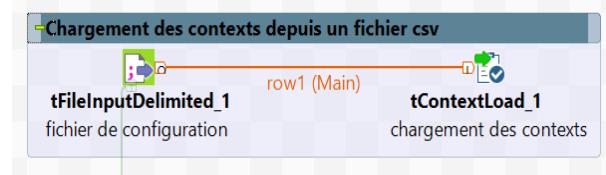
Utiliser une source externe pour les variables

Il est souvent utile d'utiliser un fichier (ou une table) avec différents paramètres.

Pour cela, dans l'onglet job, sélectionner la vue Extra



On peut également intégrer les valeurs du fichier comme indiqué ci-dessous.



Métadonnées

L'utilisation des métadonnées nous permet de :

- Récupérer des données depuis un fichier csv, excel, xml, json, positional...
- Créer des connexions à différentes sources de données.

Importer une métadonnées

Talend Open Studio for Data Integration (7.3.1.20200219_1130) | Local_Project (Connexion: local)

Fichier Editer Fenêtre Aide

[Learn](#) [Ask](#) [Exchange](#) [Videos](#) [Cloud](#)

Référentiel

LOCAL: Local_Project

- Modèles SQL
- Métadonnées
 - Connexions aux bases de données
 - Fichier délimité
 - config 0.1
 - customer 0.1
 - détails 0.1
 - entete 0.1
 - in 0.1
 - SAP_Stock 0.1
 - state 0.1
 - vehicule 0.1
 - Fichier positionnel
 - Fichier regex
 - Fichier XML
 - Fichier Excel

Outline Aperçu du code

An outline is not available.

Nouveau fichier délimité

Fichier - Etape 1 de 4

Il est recommandé de ne pas laisser le champ Objectif vide.

Nom	formation_CSV_import
Objectif	
Description	
Créé par :	user@talend.com
Verrouillé par :	
Version	0.1
Statut	
Chemin d'accès	<input type="button" value="Sélectionner"/>

< Back Next > Finish Cancel

- Dérouler "métadonnées",
- Choisir le type de données,
- Paramétriser la métadonnées en définissant un nom, un répertoire etc,
- Définir le schéma (très utile pour la suite)

Après avoir spécifié le répertoire du fichier et validé par suivant, cette fenêtre s'ouvre pour le paramétrage et la visualisation.

Nouveau fichier délimité

Fichier - Etape 3 de 4

Add a Metadata File on repository
Define the setting of the parse job

Paramètres de fichier

Encodage: US-ASCII

Séparateur de champs: Semicolon

Séparateur de lignes: Standard E

Lignes à ignorer: Si des lignes doivent être ignorées, spécifiez les paramètres

En-tête: 1

Pied de page: Ignorer les lignes vides

Limité de lignes: Si le nombre de lignes doit être limité, spécifiez ce nombre

Paramètres du caractère d'échappement

CSV: Délimité

Caractère d'échappement: Vide

Entourage du texte: Vide

Découper la ligne avant le champ

Aperçu Sortie

Definir les lignes d'en-tête comme nom de colonnes Rafraîchir l'aperçu

id	CustomerName	CustomerAddress	idState	id2	RegTime	RegisterTime
2500	Block Boats and Lures	629 Kincaid Ave.	5	7	18/03/99 23:33	1990-07-26 03:40:00.000
2501	Kermit the Pet Shop	456 Groverland Ave.	1	38	21/02/00 07:35	2002-05-04 00:08:16.000
2502	Big Bad Wolf	264 Main St.	10	20	22/02/00 15:07	1999-05-22 00:12:00:000

Exporter en tant que contexte Revenir au contexte précédent

< Back Next > Finish Cancel

Routines

A la fin de ce chapitre, nous allons être en mesure de créer une routine et de l'appeler comme une fonction pour une utilisation ultérieure.

Création d'une routine pour le pelage de chat

```
public class pelage {
    public static String getPelageChat() {
        String[] list = { "Blanc", "Noir", "Roux", "Tigré", "Gris", "Brun"};
        Integer random = 0 + ((Long) Math.round(RandomUtils.random() * (list.length - 1 - 0))).intValue()
        return list[random];
    }
}
```

Talend Open Studio for Data Integration - tRowGenerator - tRowGenerator_1

Schéma		Fonctions	
Colonne	Type	Fonctions	Variables d'environnement
id_chat	int	Numeric.sequence(String,int,int)	sequence identifier=>"s1" ; st...
nom_chat	String	TalendDataGenerator.getFirstName()	
age_chat	int	Numeric.random(int,int)	min value=>1 ; max value=>...
pelage_chat	String	pelage.getPelageChat(String)	input=> ;

Colones ▾ Nombre de lignes pour RowGenerator

Paramètres de la fonction Aperçu

helloExample: not return value, only print "hello" + message.

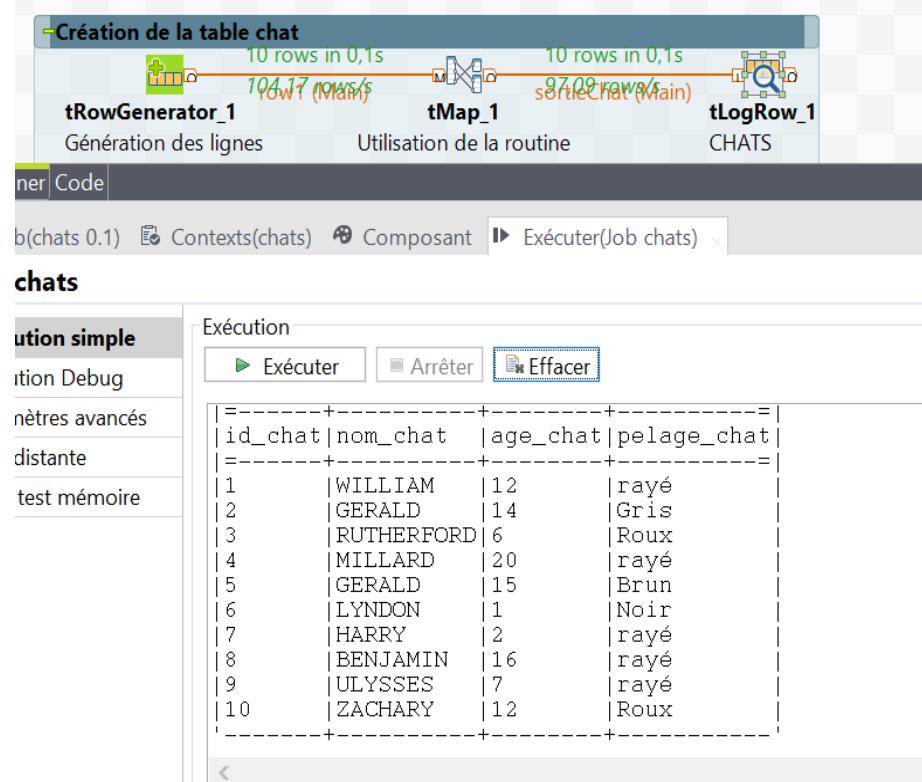
Paramètre	Valeur	Commentaire
input		The string need ...

1. A partir du composant tRowGenerator, générer une table de 10 lignes ou plus contenant des chats.
2. En partant d'une routine system, créer une routine qui nous génère de façon aléatoire le pelage de chat.
3. Utiliser cette routine pour renseigner le champ pelage

Après avoir spécifier le répertoire du fichier et validé par suivant, cette fenêtre s'ouvre pour le paramétrage et la visualisation.

Création d'une routine pour le pelage de chat

1. Créer le job chat y déposer les composants : tRowGenerator, tMap, tLogRow.
2. Les relier par un lien main et donner un nom à la sortie.
3. On obtient ceci :



Formulaires



FORMULAIRE DE SATISFACTION

Vous voulez nous aider à nous améliorer ?

Vous souhaitez vous exprimer au sujet de la formation ?

Compléter en quelques minutes notre questionnaire de satisfaction !

<https://forms.office.com/r/QRnNLuXwWC>



Le formulaire pour nos formateurs [ici](#) !

FORMULAIRE D'ÉVALUATION

Vous voulez vous tester pour connaitre vos points forts ?

Vous souhaitez connaître vos axes d'améliorations ?

Complétez en quelques minutes notre formulaire des objectifs opérationnels !

<https://forms.office.com/r/BtTsWLSh91>



Le formulaire pour nos formateurs [ici](#) !



DATA
MANAGEMENT
ERP

COMMENT LES SUPERMARCHÉS YATOU COLLECTENT ET ANALYSENT LEURS DONNÉES

YATOU

Expert DATA à Lille – Paris – Nantes et Bordeaux



DataGalaxy

talend



snowflake

Qlik



SUPERMARCHÉS YATOU

LE CONTEXTE YATOU



LE CONTEXTE

La chaîne de supermarchés YATOU a réalisé une partie de sa croissance en faisant plusieurs acquisitions d'autres chaînes concurrentes. Aujourd'hui, ce sont près de 250 magasins répartis sur 5 zones géographiques qui proposent un plus de 15k produits à la vente pour les particuliers.



Croissance externe
5 bases de données différentes
(base de données principale + 4 obtenues via les acquisitions successives)



Forte population d'utilisateur·rices
50 collaborateur·rices en centrale
Près de 750 personnes en magasin



Manque de maîtrise des données existantes et de leur qualité
Nombreux traitements Excel
Tout est concentré au contrôle de gestion
Culture data très faible en magasins

PRINCIPAUX OBJECTIFS



Proposer une analyse consolidée à la direction et aux équipes commerciales

A ce jour, seules des analyses par zone géographique sont possibles



Prévoir une architecture simple, évolutive et au juste prix
Peu de compétences infrastructure dans l'entreprise
Simplifier l'intégration de nouvelles structures



Compléter les analyses avec des données externes

Météorologie, implantation de la concurrence, zone de chalandise...



Réduire l'effet de silo dans l'entreprise
Tout doit pouvoir être partagé (données brutes ou transformées, échange entre applications métiers, analyses, documentation...)



Automatiser les manipulations de données

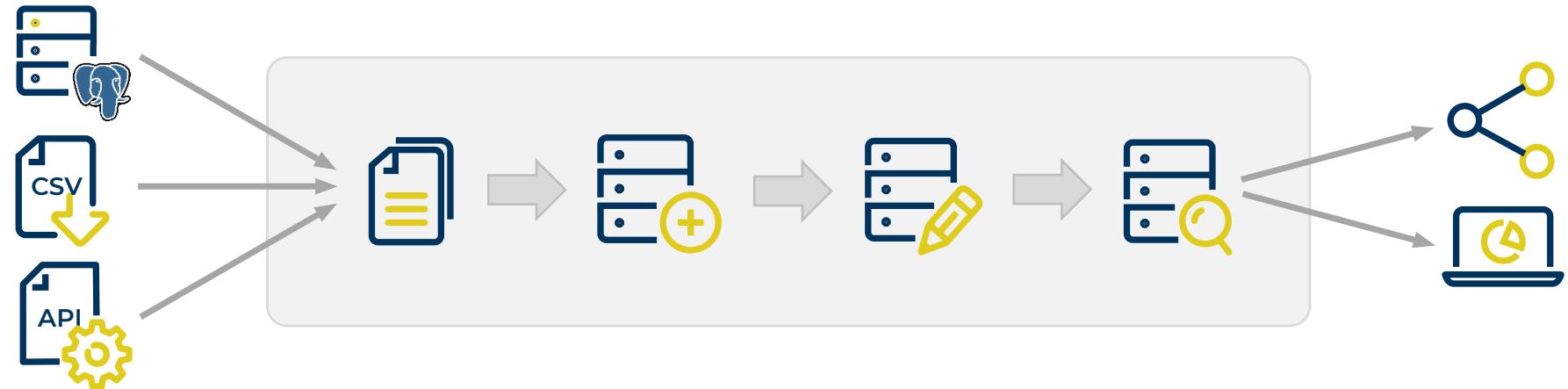
Encore beaucoup de traitements manuels et manipulations sur Excel



Donner de la confiance et de l'autonomie au métier

Le métier doit maîtriser et être responsable de sa donnée pour savoir construire des analyses pertinentes.

ARCHITECTURE



DONNÉES
SOURCES

DATA LAKE
Centralisatio
n

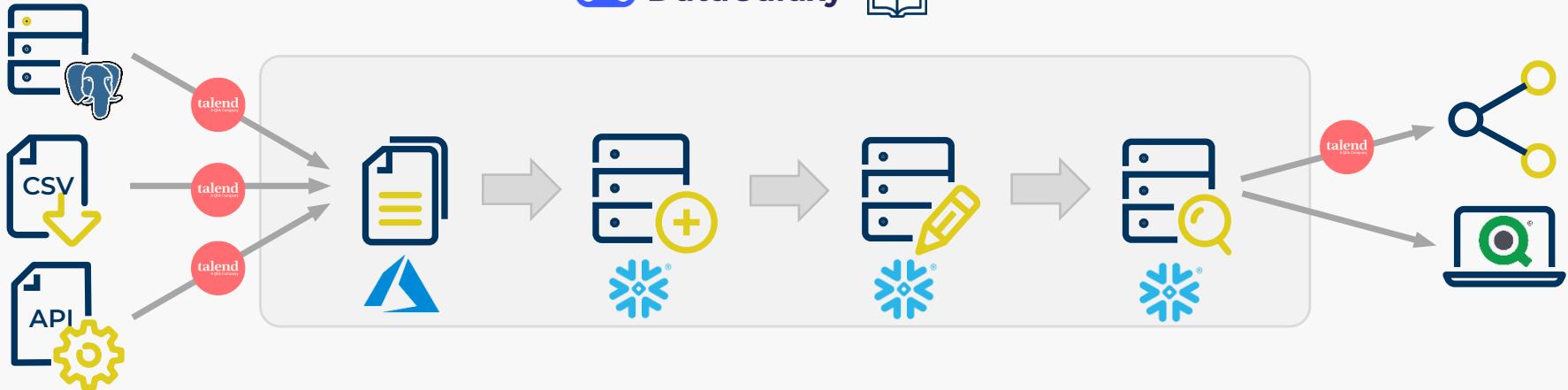
BRONZE
Réplication
Structure tabulaire

SILVER
Clarification
Consolidation
Transformation
Uniformisation

GOLD
Exposition
Mise à
disposition
Agrégation

USAGES
Analyse
Partage
...

SOLUTIONS TECHNIQUES



ETL · ELT
Ordonnancement
Partage · Reverse ETL



Stockage non structuré



Stockage structure
Transformation
Partage



Restitution · Analyse
Partage · Alerte
Automatisation process



Catalogage
Glossaire
Dictionnaire
Linéage
Usages

The background of the slide is a blurred photograph of an office space. Several people are visible, some sitting at desks working on computers, others standing or walking. The office has large windows and a modern, open-plan layout.

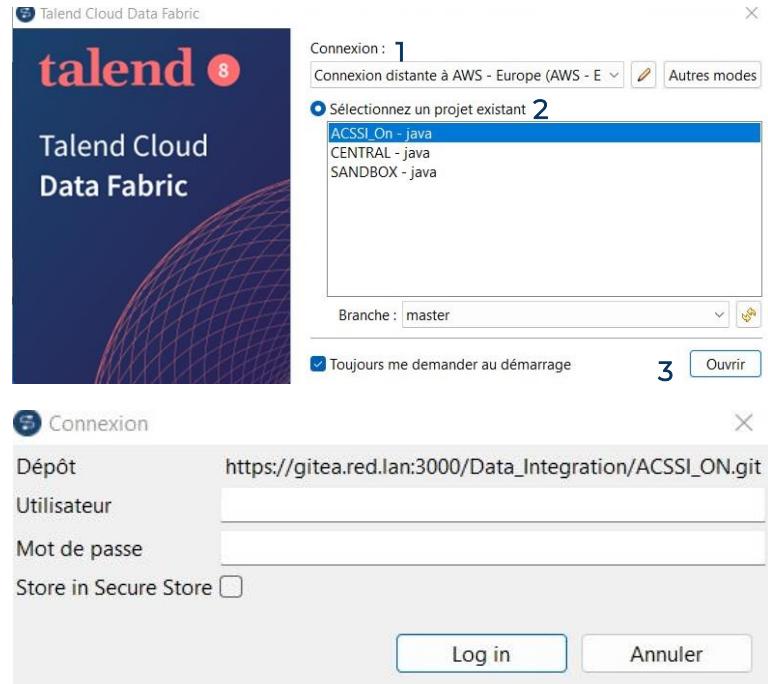
ACSSI

TALEND

ACCES AU PROJET ACSSI ON

Accès au projet Talend ACSSI ON :

- A terme une VM avec une licence unique sera mise en place pour pouvoir accéder au projet Talend. Cela évitera de devoir attribuer une licence à chaque personne qui voudrait accéder aux jobs pour une démonstration.
- Au lancement de Talend une fenêtre s'ouvre. Il faut sélectionner "Connexion à distance à AWS – Europe" dans la liste déroulante "Connexion" en haut (1).
- Sélectionner ensuite "ACSSI_On" dans la liste des projets (2) et cliquer sur "Ouvrir" (3)
- Une fenêtre d'identification s'ouvre.



La partie Jobs/Standard du référentiel

• **Dossier "AlimentationBDDSource"** : Contient les jobs qui alimentent les BD PostGre à partir des fichiers CSV. Ces jobs ne font pas partie du démonstrateur.

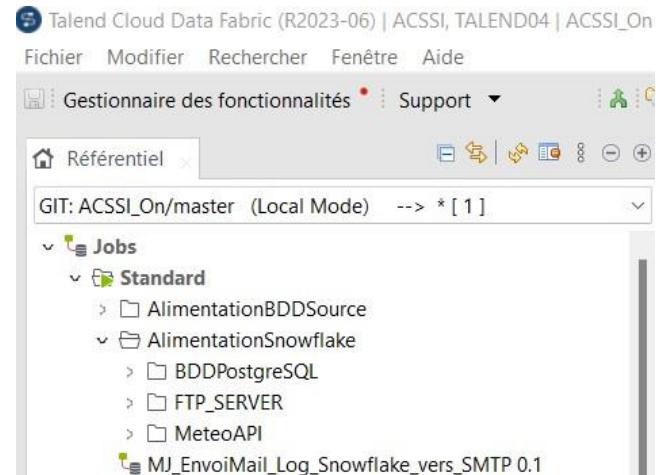
• **Dossier "AlimentationSnowflake"** : Contient les jobs qui alimentent Snowflake à partir de 3 sources de données (3 sous-dossiers) :

- BDDPostgreSQL : les jobs qui se connectent aux 5 BD PostGre* (5 zones). Les lignes sont réunies dans une table Snowflake.

- FTP SERVER : les jobs qui se connectent au serveur FTP pour accéder aux fichiers csv Objectifs et les stocker dans une table Snowflake.

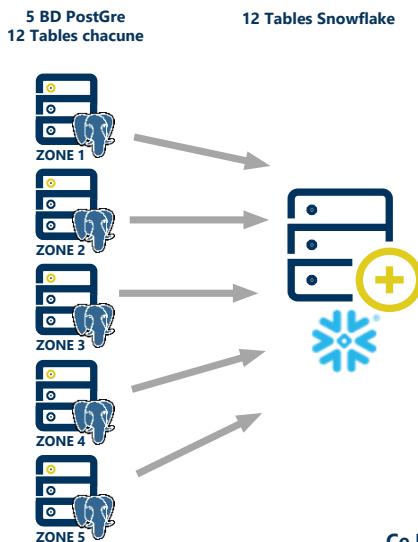
- MeteoAPI : le job qui requête l'API météo et stocke la réponse dans une table Snowflake.

- **"MJ_EnvoiMail_Log_Snowflake_vers_SMTP"** : Job déclenchant l'envoi d'e-mails après l'exécution de tous les jobs du dossier AlimentationSnowflake. Les e-mails reprennent les nouvelles lignes de la table LOGS de Snowflake par rapport à la dernière exécution.



PREMIÈRE SOURCE DE DONNÉES : POSTGRE

Chaque BD PostGre comporte 12 Tables
 Pour chacune des tables, le but est de concaténer les lignes provenant des 5 BD dans une seule et même table Snowflake



- Les tables qui ont des lignes en commun (Produits, Sous_categories, Categories, Rayons, Département, Regions, Zones). Il faudra donc enlever les doublons.
- Les tables dont toutes les lignes sont différentes pour chacune des zones (Clients, Concurrents, Magasins).
- Les tables dont toutes les lignes sont différentes pour chacune des zones, mais dont le chargement est incrémental, seules les nouvelles lignes sont ajoutées à la table Snowflake (Ventes_Entete et Ventes_Lignes).

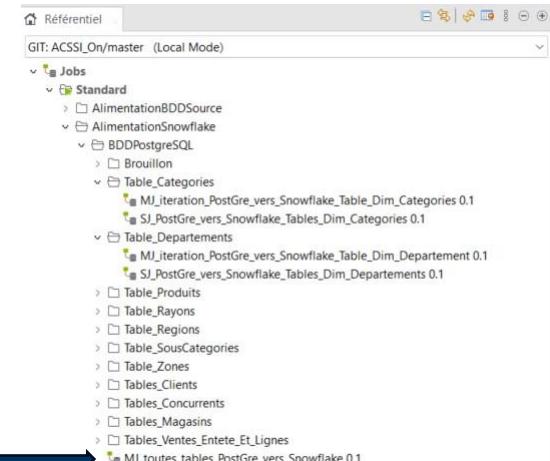
Trois types de table = Trois types de chargement :

Un sous-dossier par table qui contient ...

- **un master job (MJ)** : Le master job permet d'exécuter le sous job 5 fois, avec des paramètres de connexion différents à chaque itération, pour ainsi atteindre les 5 BD.
- **et un sous job (SJ)** : Le sous job est exécuté 5 fois. A chaque fois il se connecte à une BD différente et récupère les lignes de la table concernée.

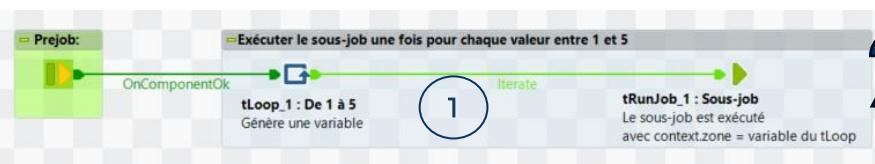
Selon le type de table la structuration du master job et du sous job diffère quelque peu

Ce Master Job permet d'exécuter les 12 Master Jobs du sous-dossier BDDPostgreSQL



PREMIÈRE SOURCE DE DONNÉES : POSTGRE LES TABLES SANS LIGNES COMMUNES

MASTER JOB



Le fonctionnement des jobs pour les tables : Clients, Concurrents et Magasins

Lancement du MasterJob :

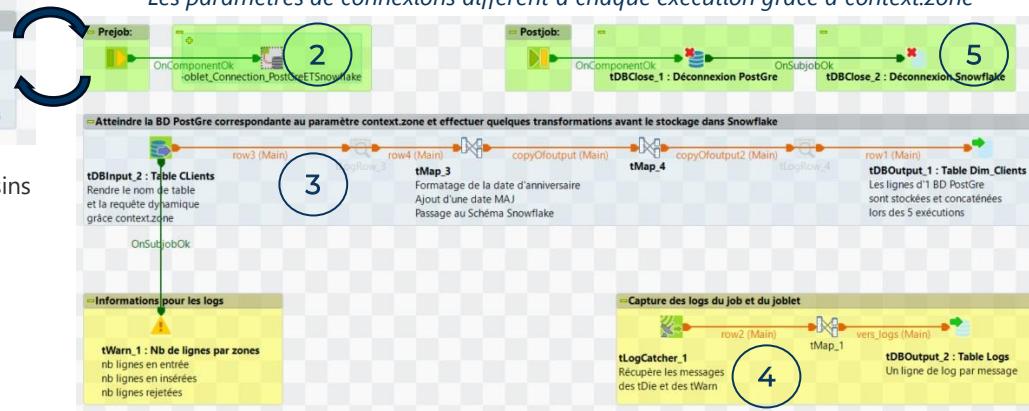
- Etape 1 : Boucle d'exécution du sous-job avec context.zone (pour tous les entiers entre 1 et 5)

Les étapes 2 à 5 sont répétées lors des 5 exécutions du SJ :

- Etape 2 : Connexion à la BD PostGre et à Snowflake (cf Joblet)
- Etape 3 : Requêtage de la BD PostGre
- Etape 3 bis : Transformations si nécessaire et ajout d'une date MAJ
- Etape 3 ter : Vers Snowflake
- Etape 4 : Ecriture des Logs dans Snowflake
- Etape 5 : Déconnexion PostGre et Snowflake

SOUS JOB X5

Les paramètres de connexions diffèrent à chaque exécution grâce à context.zone



Légende des jobs :

Prejob et Postjob

Traitements des logs

Traitements principaux

YATOU

PREMIÈRE SOURCE DE DONNÉES : POSTGRE LES TABLES AVEC LIGNES COMMUNES

Le fonctionnement des jobs pour les tables :

- Produits
- Sous_categories
- Categories
- Rayons
- Départements
- Regions
- Zones

Lancement du MasterJob :

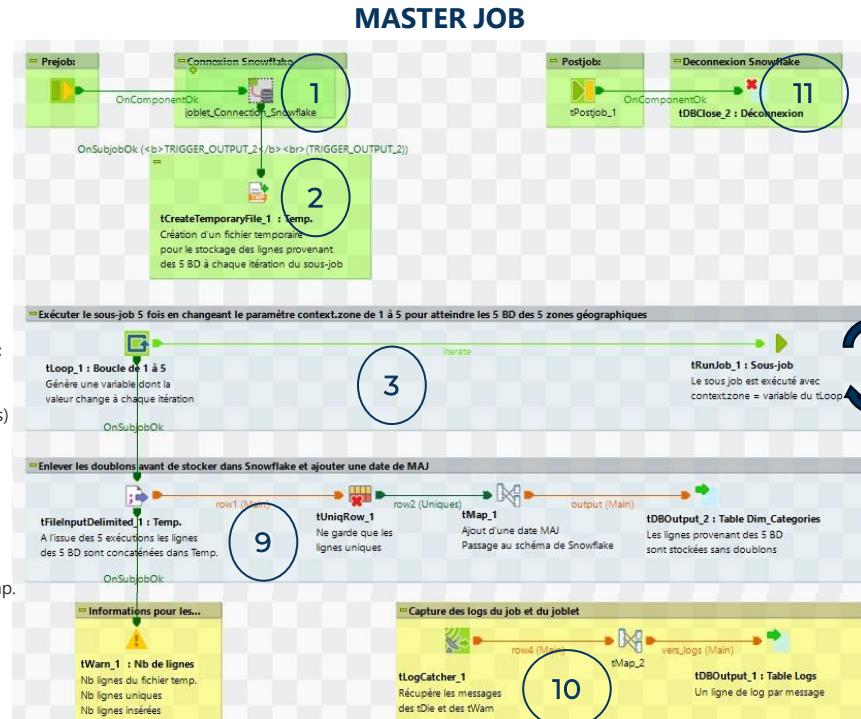
- Etape 1 : Connexion à Snowflake (cf Joblet)
- Etape 2 : Création d'un fichier temporaire
- Etape 3 : Boucle d'exécution du sous-job avec :
 - context.zone (pour tous les entiers entre 1 et 5)
 - chemin vers fichier temporaire (le même)

4 à 8 sont répétés lors des 5 exécutions du SJ :

- Etape 4 : Connexion à la BD PostGre (cf Joblet)
- Etape 5 : Récup. de la connexion Snowflake (Logs)
- Etape 6 : Requête de la BD PostGre
- Etape 6 bis : Vers fichier temporaire
- Etape 7 : Ecriture des Logs dans Snowflake
- Etape 8 : Déconnexion PostGre

Retour au MasterJob :

- Etape 9 : Elimination des doublons du fichier temp.
- Etape 9 bis : Ajout d'une date MAJ
- Etape 9 ter : Vers Snowflake
- Etape 10 : Ecriture des Logs dans Snowflake
- Etape 11 : Déconnexion de Snowflake



Légende des jobs :

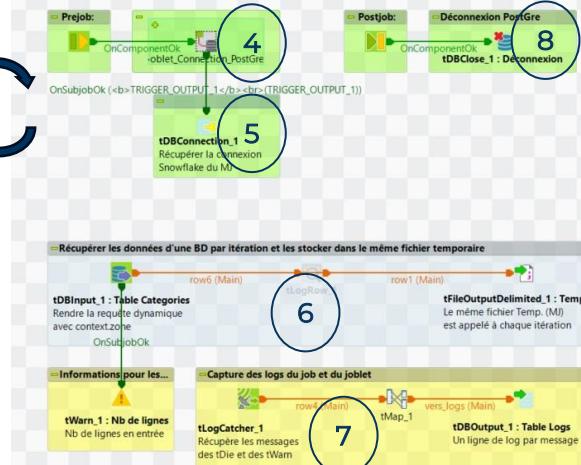
Prejob et Postjob

Traitement des logs

Traitement principal

SOUS JOB X5

Les paramètres de connexions diffèrent à chaque exécution grâce au paramètre de contexte context.zone

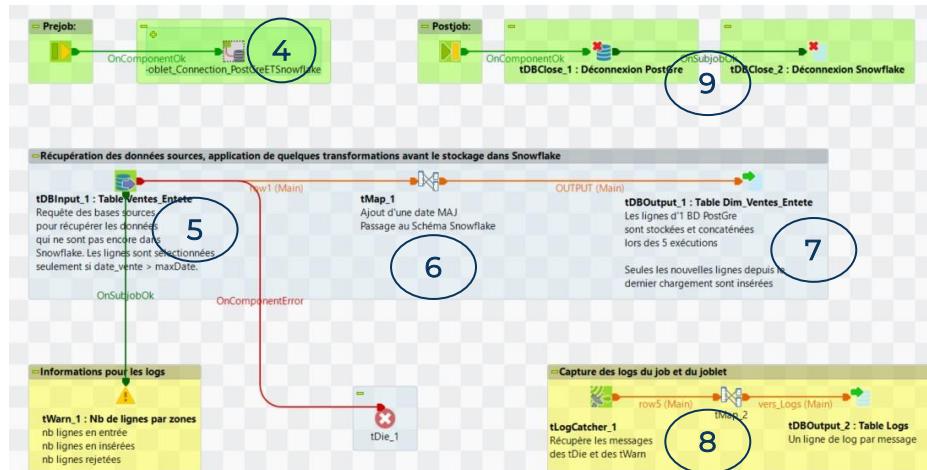


YATOU

PREMIÈRE SOURCE DE DONNÉES : POSTGRE LES TABLES SANS LIGNES COMMUNES + INCRÉMENTAL

SOUS JOB X5

Les paramètres de connexions diffèrent à chaque exécution grâce à context.zone



Le fonctionnement des jobs pour les tables : Ventes_Entete et Ventes_Lignes

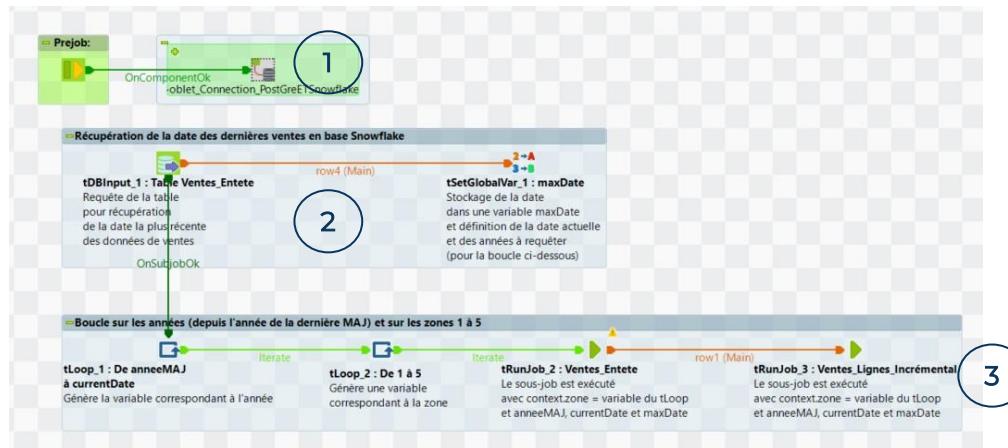
Les étapes 4 à 9 sont répétés lors des 5 exécutions du SJ :

- Etape 4 : Connexion à PostGre et Snowflake (cf Joblet)
- Etape 5 : Requête de la BD PostGre pour les lignes où la date de vente est :
 - supérieure à maxDate
 - inférieure à currentDate
- Etape 6 : Transformations si nécessaire et ajout d'une date MAJ
- Etape 7 : Vers Snowflake
- Etape 8 : Ecriture des Logs dans Snowflake
- Etape 9 : Déconnexion PostGre et Snowflake

Les deux sous-job fonctionnent de la même manière

PREMIÈRE SOURCE DE DONNÉES : POSTGRE LES TABLES SANS LIGNES COMMUNES + INCRÉMENTAL

MASTER JOB



Le fonctionnement des jobs pour les tables : Ventes_Entete et Ventes_Lignes

Lancement du MasterJob :

- Etape 1 : Connexion Snowflake (cf Joblet)
- Etape 2 : Requête de la date vente la plus récente (ventes_entete)
- Etape 2 bis : Stockage dans des variables globales de :
 - la date de vente la plus récente (maxDate)
 - la date du jour (CurrentDate)
- Etape 3 : Boucle sur les années et sur les zones pour l'exécution des sous-jobs :
 - context.zone (pour tous les entiers entre 1 et 5)
 - maxDate (toujours la même)
 - currentDate (toujours la même)
 - annneeMAJ (pour chaque année entre maxDate et currentDate)

La boucle sur les années permet de ne pas saturer la mémoire lors de linitialisation (le 1er remplissage de la table Snowflake) car il y aurait trop de lignes d'un coup

YATOU

2ÈME SOURCE DE DONNÉES : .CSV SUR FTP

3ÈME SOURCE : API

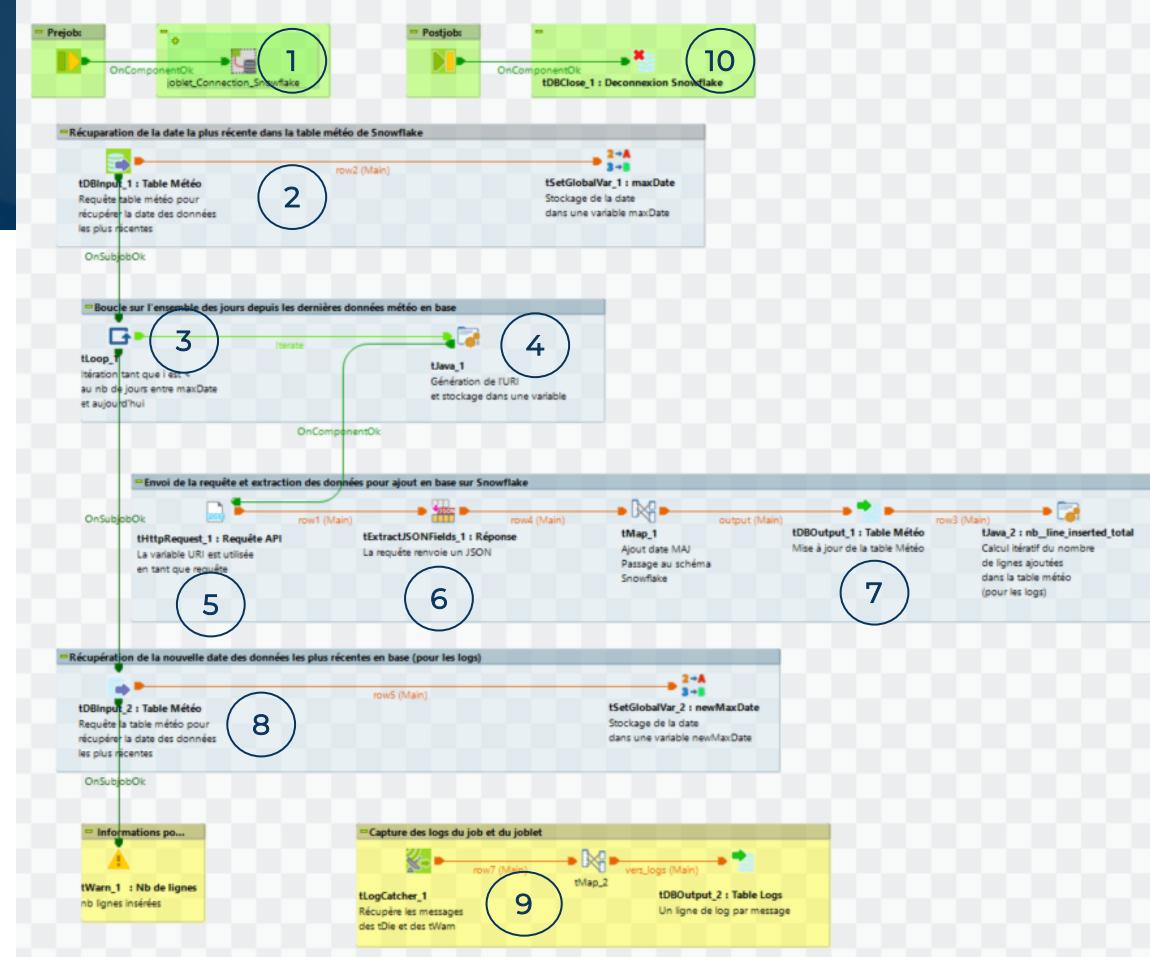
Le fonctionnement du job requérant l'API Météo :

Ce job permet à la fois d'initier la table Météo et de la mettre à jour

- Etape 1 : Connexion Snowflake (cf Joblet)
- Etape 2 : Requête de la date météo la plus récente
- Etape 2 bis : Stockage dans des variables globales de :
 - la date la plus récente (maxDate)
 - la date du jour (CurrentDate)
- Etape 3 : Boucle pour chaque jour entre maxDate et currentDate

Les étapes 4 à 7 sont répétées autant de fois que le nombre de jours requêtés

- Etape 4 : Construire les URI dans le tJava
- Etape 5 : Requêtage grâce à l'URI
- Etape 6 : Extraction des champs JSON de la réponse
- Etape 7 : Stockage des données dans Snowflake
- Etape 8 : Paramétrage de la date pour les logs
- Etape 9 : Stockage des logs
- Etape 10 : Déconnexion de Snowflake

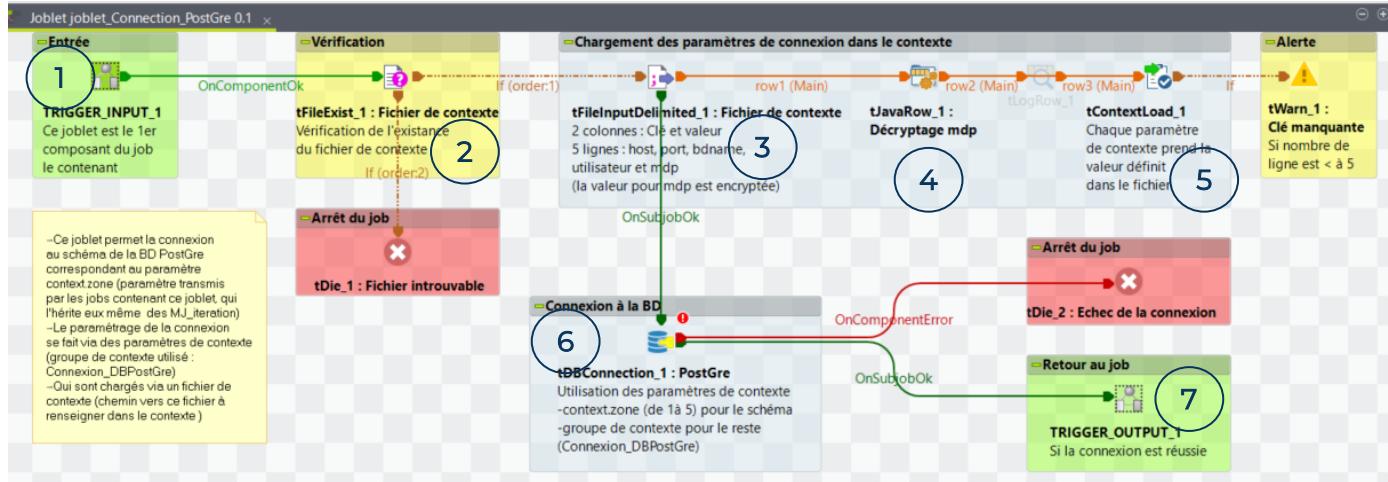


LES JOBLETS

Le fonctionnement des joblets :

Les joblets sont des mini-job que l'on peut appeler dans un job.
Cela permet d'éviter de développer les mêmes parties de jobs plusieurs fois.

Il partage les paramètres de contexte avec le job qui le contient



- Etape 1 : Début du joblet
- Etape 2 : Vérification de l'existence du fichier de contexte, sinon arrêt du job (tDie)
- Etape 3 : Chargement du fichier de contexte stocké en local
- Etape 4 : Décryptage des mots de passe grâce à une routine Java
- Etape 5 : Les paramètres de contexte prennent les valeurs du fichier de contexte
- Etape 6 : Connexion grâce aux paramètres de contexte, arrêt du job si erreur (tDie)
- Etape 7 : Retour au job principal

FOCUS SUR LES COMPOSANTS

tDBConnection :

- Permet de paramétriser une connexion à tout type de BD
- Qui peut être réutilisée à plusieurs endroit dans le job
- Les alias de connexion permettent de transmettre la connexion d'un job à un autre
- Tous les paramètres sont variabilisables (ce qui permet d'exécuter le même job pour plusieurs connexions)



Database: PostgreSQL Appliquer

Type de propriété: Built-in

Version de la base de données: V9 et plus

Hôte: context.host

Port: context.port

Base de données: context.bdname

Schéma: "zone_."+context.zone

Utilisateur: context.utilisateur

Mot de passe: context.mdp

tDBInput :

- Permet de paramétriser une connexion à tout type de BD
- Ou de récupérer la connexion créée dans un tDBConnection
- C'est ce composant qui porte la requête SQL
- Les paramètres du composant sont variabilisables
- La requête est également variabilisable



Schéma: Référentiel Bases de données (POSTGRESQL): Modifier le schéma

Nom de la table: "Categories"

Type de requête: Built-in Guess Query Guess schema

Requête:

```
"SELECT
    id_categorie,
    categorie,
    id_rayon
FROM zone_."+context.zone+".\"Categories\""
```

FOCUS SUR LES COMPOSANTS

tLoop :

- Permet d'initier une boucle de type For ou While
- Les paramètres du composant sont variabilisables
- Le numéro d'itération devient une variable utilisable dans le job (ex : context.zone)
- Utiliser avec un tRunJob il permet d'exécuter le même job en faisant varier des paramètres



Type de boucle	<input type="radio"/> Pour <input checked="" type="radio"/> While
Déclaration	int i=0
Condition	i<10
Itération	i++

tRunJob :

- Permet l'exécution d'un Sous-job dans un Master-job. Les jobs sont donc imbriqués
- Il est possible de transmettre des paramètres de contexte aux jobs fils (ex : context.zone)



Job	Version	Latest	Contexte
<input type="checkbox"/> Utiliser un processus indépendant pour exécuter le sous-Job <input checked="" type="checkbox"/> Arrêter en cas d'erreur de l'enfant <input type="checkbox"/> Transmettre le contexte complet			
Paramètres de contexte	Paramètres	Valeurs	

FOCUS SUR LES COMPOSANTS

**tMap :**

- Permet de réorganiser les colonnes
- Et d'appliquer des fonctions Java
- Permet de faire des jointures
- Peut avoir plusieurs entrées, mais une seule entrée principale
- Permet de filtrer les lignes et les colonnes
- Permet de récupérer les rejets
- Peut avoir plusieurs sorties, il peut donc splitter les lignes en sortie

The screenshot displays the Talend Studio interface with a tMap component selected. The 'Input' pane on the left lists columns from a source table named 'row2', which include 'id_produit', 'nom_produit', 'id_souscategorie', and 'cout_produit'. The 'Output' pane on the right lists columns for the target table, including 'ID_PRODUIT', 'NOM_PRODUIT', 'ID_SOUSCATEGORIE', 'COUT_PRODUIT', and 'DATE_MAJ'. Below these panes are detailed views of the schema and expression editors for both input and output columns.

Colonne	Type	Modèle de date (SI)	Longueur	Precision	Par déf.	Comment.
id_produit	int		10	0		
nom_produit	String		100	0		
id_souscategorie	Integer		10	0		
cout_produit	Float		10	1		

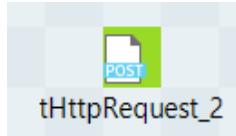
Colonne	Type	Modèle de date (SI)	Longueur	Precision	Par déf.	Comment.
ID_PRODUIT	Integer		10	0		
NOM_PRODUIT	String		100	0		
ID_SOUSCATEGORIE	Integer		10	0		
COUT_PRODUIT	Float		10	1		
DATE_MAJ	Date	'dd-MM-yy HH..	10	0		

FOCUS SUR LES COMPOSANTS



tUniqRow :

- Permet d'éliminer les doublons dans un ensemble de ligne
- Il est possible de définir ce qu'est un doublon en sélectionnant les colonnes qui doivent être identiques pour que deux lignes soient considérées comme des doublons



tHttpRequest :

- Permet de requêter une API en GET ou en POST
- L'URI de la requête peut être une variable (ex : job API Météo)



tJava et tJavaRow :

- Permet d'exécuter du code Java directement dans le job
- Le tJava permet d'exécuter du code Java pour chaque ligne
- S'il n'existe pas de composant pour une action donnée, le tJava est souvent la solution

FOCUS SUR LES COMPOSANTS



tWarn :

- Permet d'insérer des messages de plusieurs niveaux de priorités (Info, Avertissement, ...) conditionnés par des liens "Run if"
- Les variables sont utilisables dans le message paramétré
- Ce composant permet de rendre les Logs du job plus verbeux, par exemple en affichant l'interprétation d'une variable



tDie :

- Même principe que le tWarn, mais ce composant va en plus stopper le job si la condition du "Run if" est remplie



tLogCatcher :

- Permet de récupérer tous les messages paramétrés dans les tWarn et les tDie du job, joblets inclus
- Pour ensuite les stocker dans une table ou un fichier de Log

A photograph of two women in an office hallway. One woman is in the foreground, mid-jump, wearing a dark jacket and jeans. The other woman is further back, also jumping, wearing a grey sweater and blue jeans. There are several purple balloons floating in the air. The background shows office doors and windows.

ALORS, VOUS AVEZ EU TOUT BON ?

CORRECTION

CORRECTION

7. Un projet Talend est représenté par * (1 point)

Objectif opérationnel : Connaitre l'interface et le fonctionnement de Talend

- un export ZIP d'un job Talend
- l'ensemble des composants disponibles
- une structure regroupant jobs, business model, métadonnées, documentations... ✓



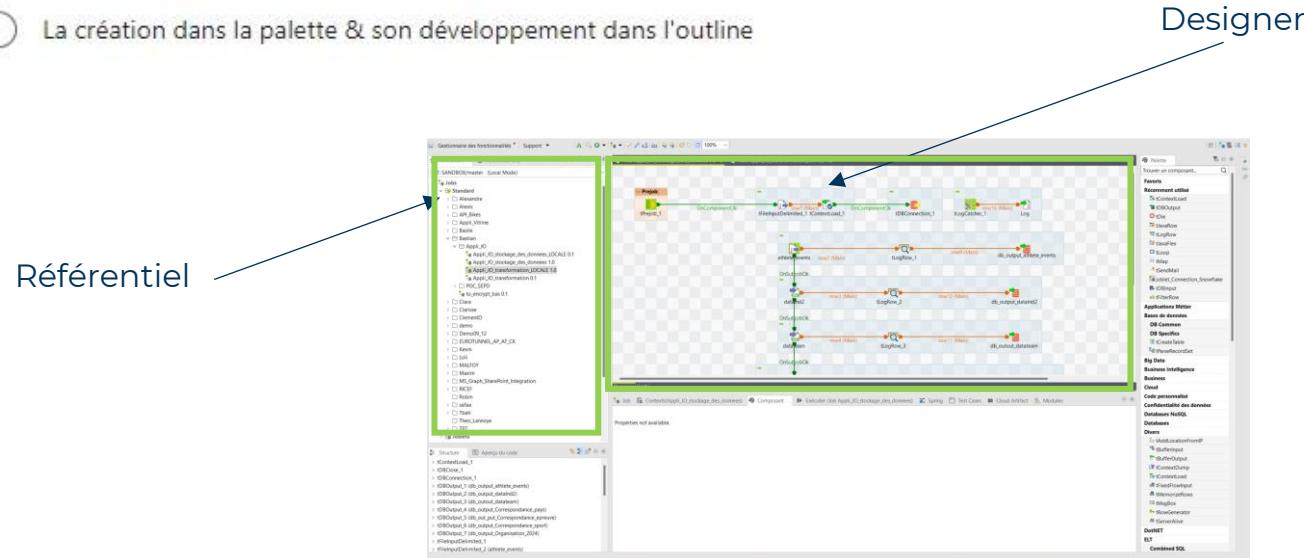
Vous retrouverez tous les éléments de votre projet dans la partie gauche de l'écran, dans le référentiel.

CORRECTION

8. La création d'un job est géré comme ceci: * (1 point)

Objectif opérationnel : Connaitre l'interface et le fonctionnement de Talend

- La création dans la palette & son développement dans le designer
- La création dans le référentiel/repository & son développement dans le designer ✓
- La création dans la palette & son développement dans l'outline



CORRECTION

9. Est-il possible de supprimer un projet Talend après y avoir crée des jobs? * (1 point)

Objectif opérationnel : Connaître l'interface et le fonctionnement de Talend

Oui ✓

Non

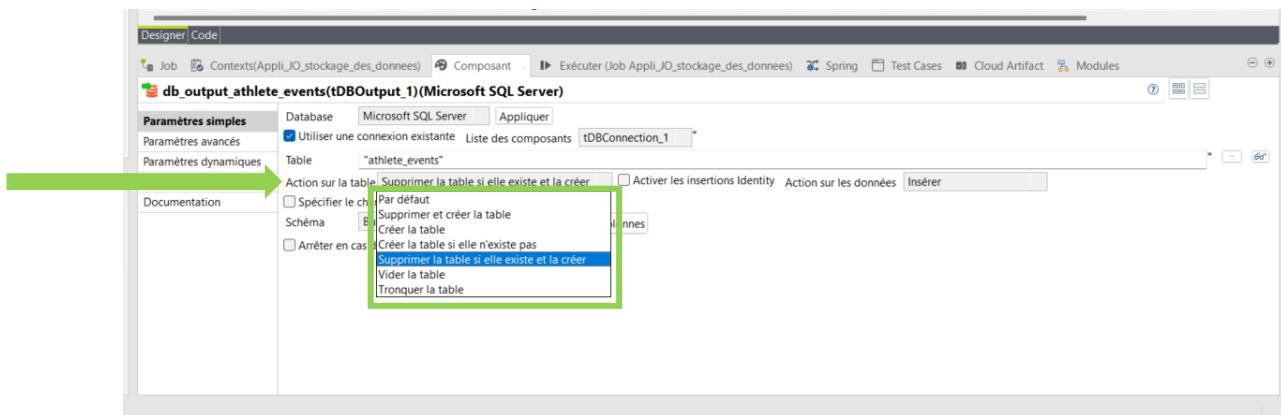
La suppression d'un projet entraîne la suppression de tous les éléments composants le projet (jobs, métadonnées, routines, ...)

CORRECTION

10. Quelle option n'est pas présente lors de l'écriture de données dans une table * (1 point)

Objectif opérationnel : Savoir manipuler des données

- Créer la table
- Supprimer la table & la créer
- Supprimer la table si elle existe et la créer
- Modifier la table ✓
- Tronquer la table
- Créer la table si elle n'existe pas



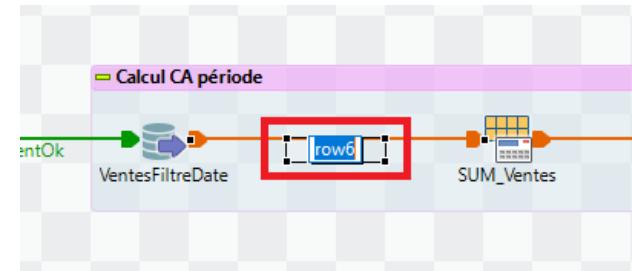
CORRECTION

11. Pour renommer une liaison entre deux composants: * (1 point)

Objectif opérationnel : Connaitre l'interface et le fonctionnement de Talend

- Cela n'est pas possible
- On clique deux fois de façon intermittente sur le nom du lien ✓
- Un clic droit sur le lien permet de retrouver l'option "renommer"

Lorsque vous renommez vos liens, attention aux composants où ces noms sont utilisés (tMap, tJava, tSetGlobalVar, ...).

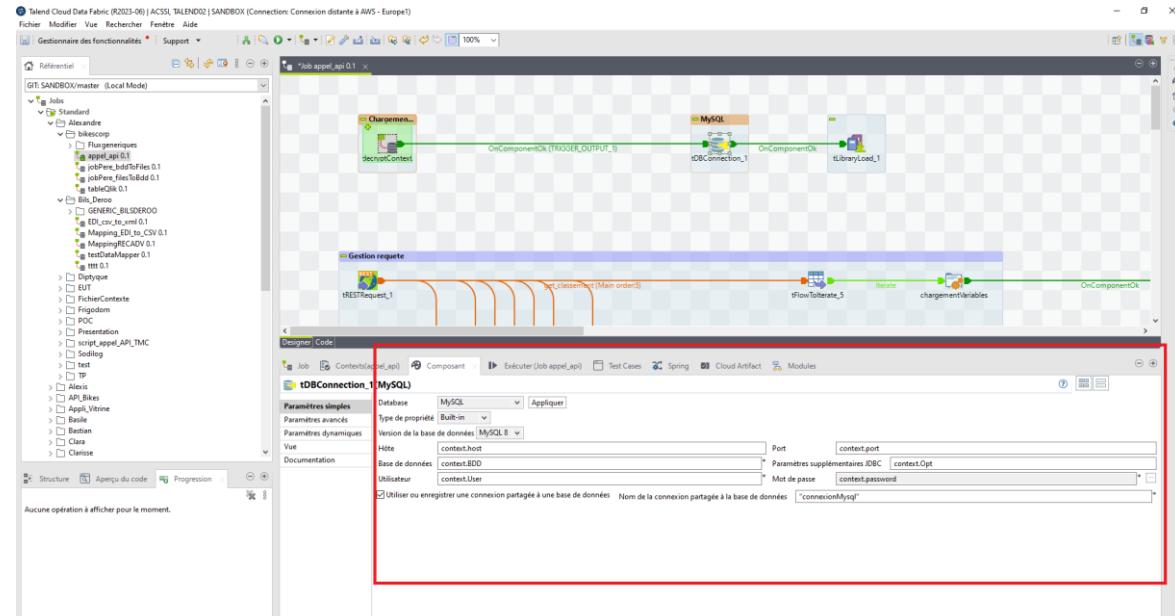


CORRECTION

12. Le principal onglet utilisé pour le développement des flux est : * (1 point)

Objectif opérationnel : Connaître l'interface et le fonctionnement de Talend

- L'onglet Composant ✓
- L'onglet Context
- L'onglet Execute



CORRECTION

13. Par défaut, une jointure créée dans un tmap est du type: * (1.5 points)

Objectifs opérationnels :

- Savoir manipuler des données
- Connaître les composants principaux de Talend

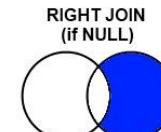
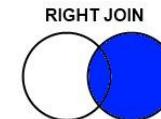
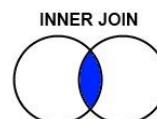
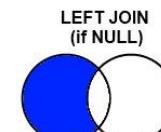
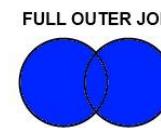
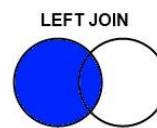
RIGHT JOIN

INNER JOIN

UNION

LEFT JOIN ✓

VenteFiltreProduct	
Property	Value
Lookup Model	Charger une fois
Match Model	Correspondance unique
Join Model	Left Outer Join
Store temp data	false
Clé d'expr.	Column
VentesfiltreDate.SalesOrderDetailID	Ventes SalesOrderDetailID



CORRECTION

14. Dans un tmap, il est possible d'effectuer les jointures suivantes: * (1.5 points)

Objectifs opérationnels :

- Savoir manipuler des données
- Connaitre les composants principaux de Talend

RIGHT JOIN

INNER JOIN ✓

Les jointures ne sont pas disponibles

LEFT JOIN ✓

Right join n'est pas disponible, mais il suffit d'inverser les tables et d'utiliser le Left Join

CORRECTION

15. Dans un TDBOutput, il est possible d'agir sur les données présentes dans une table mais également sur la table en elle même * (1 point)

Objectifs opérationnels :

- Savoir manipuler des données
- Connaître les composants principaux de Talend

Vrai ✓

Faux

 tDBOutput_2(MySQL)

Paramètres simples	Database	MySQL	Appliquer
Paramètres avancés	<input checked="" type="checkbox"/> Utiliser une connexion existante Liste des composants tDBConnection_1 - connexionMySQL_1 *		
Paramètres dynamiques	Table	"mesures"	
Vue	Action sur la table	Créer la table	Action sur les données Insérer
Documentation	Schéma	Built-in	Modifier le schéma ... Sync colonnes
	<input type="checkbox"/> Arrêter en cas d'erreur		

CORRECTION

16. Quelles liaisons sont présentes dans Talend? * (2 points)

Objectif opérationnel : Connaitre l'interface et le fonctionnement de Talend

- OnSubjobOk ✓
- If / run if ✓
- OnComponentError ✓
- While
- Main / principal ✓
- Iterate / Iterer ✓

Le type de liaison disponible dépend du composant. Clic droit sur un composant pour voir les types de liaisons possibles

CORRECTION

17. Quel composant peut remplacer au mieux le tmap si celui-ci est utilisé pour son côté transformation de données & non mapping ? * (1.5 points)

Objectifs opérationnels :

- Savoir manipuler des données
- Connaître les composants principaux de Talend

- tJava
- tHashMap
- tJavaRow ✓
- tSendMail

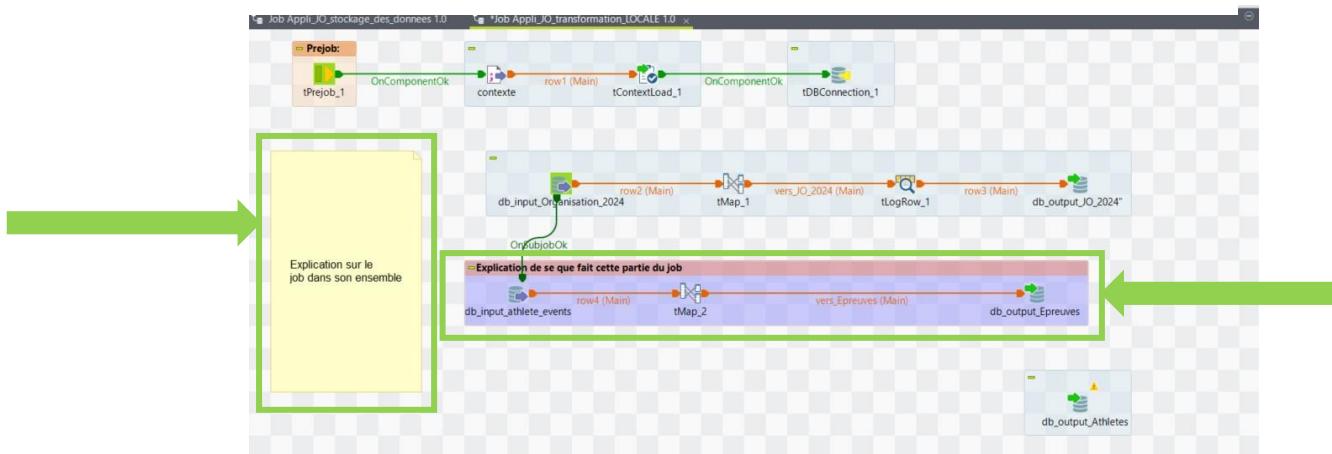
Le tJavaRow permet d'exécuter des instructions directement en code java sur chaque ligne de donnée. Attention à ne pas confondre avec le tJava qui exécute des instructions une seule fois et sans modifier les données.

CORRECTION

18. Pour faciliter la lecture des flux, il est possible: * (1 point)

Objectif opérationnel : Adopter de bonnes pratiques de développement et d'organisation

- de commenter les blocs sousjob (titre & couleurs) ✓
- d'ajouter des commentaires globaux via le composant Note ✓
- Il n'est pas possible de faciliter la lecture des flux



CORRECTION

19. La création d'un job n'est possible qu'au moment de la création du projet * (1 point)

Objectif opérationnel : Connaitre l'interface et le fonctionnement de Talend

- Vrai
- Faux ✓

Vous pouvez à tout moment créer et supprimer des jobs de votre projet, ainsi que des dossiers de job

CORRECTION

20. Le raccourci permettant de parcourir les variables de contexte est: * (1 point)

Objectifs opérationnels :

- Connaitre l'interface et le fonctionnement de Talend
- Adopter de bonnes pratiques de développement et d'organisation

Maj+Entrée

Ctrl + Espace ✓

Alt + F4

Plus précisément, Ctrl + Espace est le raccourci d'autocomplétion. Il vous permet de parcourir les variables de context mais également les différentes méthodes java disponibles.

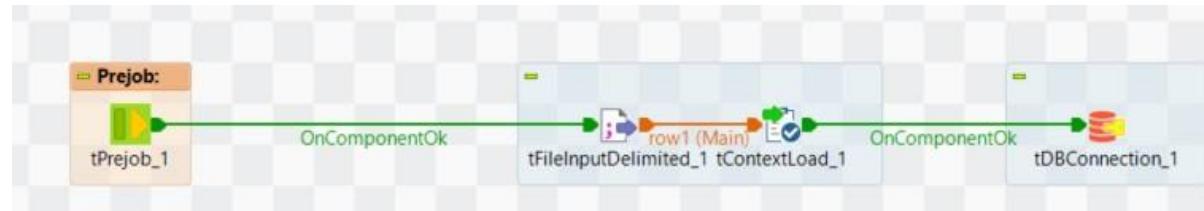
CORRECTION

21. Le composant permettant de charger des variables de contexte depuis un fichier est le : *
(1 point)

Objectif opérationnel : Connaitre les composants principaux de Talend

- tContextLoad ✓
- tChargeContext
- tMap

Il est conseillé de relier la partie chargement du contexte au composant tPrejob



CORRECTION

22. Le tMap permet de créer : * (1 point)

Objectif opérationnel : Connaitre les composants principaux de Talend

- Un flux de sortie
- Un nombre indéfini de sorties ✓

Pensez à bien nommer vos sorties pour éviter les confusions

CORRECTION

23. Est-il possible d'importer des jobs développés dans d'autres projets dans un nouveau projet?

* (1.5 points)

Objectif opérationnel : Connaître l'interface et le fonctionnement de Talend

- Oui
- Non
- Oui, attention à la version & la compatibilité ✓

Via les boutons en haut de l'écran ou via un clic droit dans le référentiel (importer des éléments)

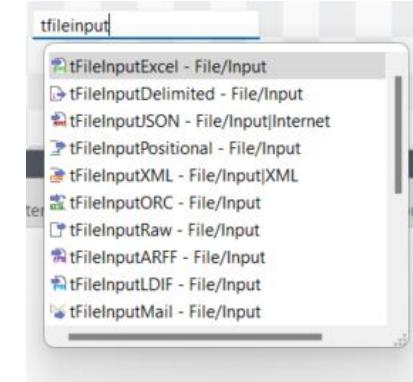


CORRECTION

24. Quel composant permet de lire en entrée un fichier csv ? * (1 point)

Objectif opérationnel : Connaitre les composants principaux de Talend

- tFileInputDelimited ✓
- tDBOutput
- tJava



Il existe des tFileInput pour de nombreux types de fichier

CORRECTION

25. Quel composant permet d'insérer des lignes dans une base de données ? * (1 point)

Objectif opérationnel : Connaitre les composants principaux de Talend

- tDBOutput ✓
- tJava
- tFileOutputDelimited

Ce composant permet d'insérez, de supprimer ou de mettre à jour des lignes

CORRECTION

26. Le composant tDBConnexion/tDBConnection permet: * (1 point)

Objectif opérationnel : Connaitre les composants principaux de Talend

- de centraliser une connexion à une base de données pour l'ensemble du job ✓
- de centraliser une connexion à une base de données pour l'ensemble du projet
- Ce composant n'existe pas

La connexion est paramétrée une seule fois pour tout le job. Il suffit de cocher "utiliser une connexion existante" dans le tDBOutput et tDBInput et de choisir la connexion correspondante.

CORRECTION

27. Il est possible de modifier la version d'un job au moment de sa création * (1 point)

Objectif opérationnel : Connaître l'interface et le fonctionnement de Talend

- Oui
- Oui, mais cela peut également être fait plus tard ✓
- Non

Versionner les jobs permet de revenir à une étape précédente du développement en cas d'erreur ou de modification

CORRECTION

28. Quel composant permet d'appeler un job fils dans un job père? * (1 point)

Objectif opérationnel : Connaitre les composants principaux de Talend

-
- tRunJob ✓
 - tJava
 - tMap
 - tHashMap

Le tRunJob permet par ailleurs de transmettre des paramètres de contexte aux jobs fils

CORRECTION

29. Le composant tMap permet également de récupérer les rejets * (1 point)

Objectif opérationnel : Connaître les composants principaux de Talend

Vrai ✓

Faux

Tout comme le tFilter

CORRECTION

30. Des variables peuvent être créées et/ou récupérées depuis: * (2.5 points)

Objectif opérationnel : Connaître les composants principaux de Talend

- la vue Outline ✓
- le composant tSetGlobalVar ✓
- la vue contexte ✓
- le composant tJava ✓
- le composant tVarSet

Le composant tVarSet n'existe pas

CORRECTION

31. Il est possible de transmettre les valeurs des variables de contexte d'un job à l'autre *
(1 point)

Objectif opérationnel : Connaître l'interface et le fonctionnement de Talend

Vrai ✓

Faux

▶ **tRunJob_1**

Paramètres simples	Schéma	Built-in	Modifier le schéma	...	Copier le schéma du Job enfant
Paramètres avancés	<input type="checkbox"/> Utiliser un Job dynamique				
Paramètres dynamiques	Job <input type="text" value="appel_api"/>				
Vue	<input type="checkbox"/> Utiliser un processus indépendant pour exécuter le sous-Job				
Documentation	<input checked="" type="checkbox"/> Arrêter en cas d'erreur de l'enfant				
	<input checked="" type="checkbox"/> Transmettre le contexte complet				
Paramètres de contexte	Paramètres	Valeurs			
	host	'localhost'			
	port	'8088'			

CORRECTION

32. Le composant tUniqRow permet de: * (1 point)

Objectif opérationnel : Connaître les composants principaux de Talend

- Supprimer les doublons ✓
- Trier les données
- limiter le résultat à sa première ligne (similaire à un LIMIT 1 ou TOP 1)

Il est possible de définir ce qu'est un doublon en sélectionnant les colonnes qui doivent être identiques pour considérer des lignes comme des doublons



CORRECTION

33. Le composant permettant de trier les données est le: * (1 point)

Objectif opérationnel : Connaitre les composants principaux de Talend

tSortRow ✓

tFilterRow

tJava

- Dans un tSortRow il faut définir la colonne de tri et le type de tri souhaité
- Le tFilterRow permet de filtrer
- Le tJava exécute du code Java



CORRECTION

34. Quelle est la différence entre une variable locale & globale? * (1 point)

Objectif opérationnel : Connaitre l'interface et le fonctionnement de Talend

- Il n'y a aucune différence
- Une variable globale peut être utilisée dans l'ensemble du job tandis qu'une variable locale est visible seulement depuis le composant dans lequel elle est créée
- Une variable locale est défini dans les métadonnées & ne peut pas être utilisé dans les sous-job



En règle générale, les variables locales sont générées dans les composants java et ne sont pas exploitables dans les autres composants du flux.

A l'inverse, les variables globales peuvent être exploitées dans l'ensemble du flux. On retrouve par exemple, les variables de context et les variables présentent dans l'onglet outline/structure.

CORRECTION

35. Quel composant permet de créer un jeu de données rapidement? * (1 point)

Objectif opérationnel : Connaître les composants principaux de Talend

- tInsertRow
- tRowGenerator ✓
- tMap



- Le composant tInsertRow n'existe pas
- Le tMap de génère pas de données, il lui faut au moins une entrée et une sortie

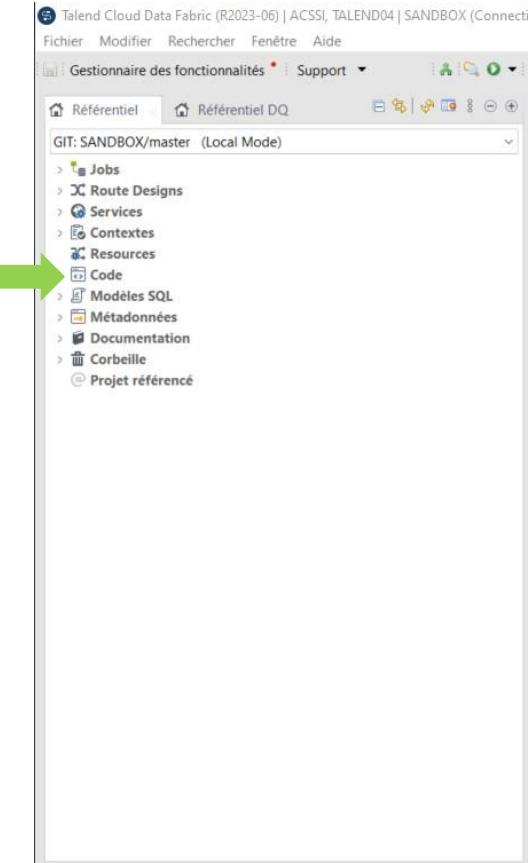
CORRECTION

36. A quel endroit pouvons nous créer des routines JAVA ? * (1 point)

Objectifs opérationnels :

- Savoir manipuler des données
- Connaître les composants principaux de Talend

- Dans la vue Référentiel/Code ✓
- Dans la vue Palette
- Dans la vue Designer



CORRECTION

37. Quels composants sont utiles pour la gestion des logs ? * (1 point)

Objectifs opérationnels :

- Connaître les composants principaux de Talend
- Adopter de bonnes pratiques de développement et d'organisation

- tWarn / tDie, tLogCatcher ✓
- tJava / tMap / tSendMail
- tFileInputDelimited / tDBOutput / Note

- tWarn : Permet de générer des messages d'information ou d'avertissement.
- tDie : Permet d'arrêter le job en cas d'erreur sur un composant ou un sous job.
- TLogCatcher : Permet de récupérer tous les messages des tWarn et tDie d'un job

Les tWarn et tDie permettent de définir des messages d'information, avertissement ou erreur comprenant par exemple des variables. C'est une façon de rendre les logs d'un job plus verbeux

CORRECTION

38. Quel composant permet de déplacer un fichier d'un répertoire A vers un répertoire B ? *
(1 point)

Objectif opérationnel : Connaître les composants principaux de Talend

- tMoove
- tFileCopy ✓
- tFileList



- Le composant tMoove n'existe pas
- Le tFileList permet de lister tous les fichiers d'un répertoire

CORRECTION

39. Quel composant permet d'écrire & executer directement des requêtes SQL depuis Talend? *

(1.5 points)

Objectifs opérationnels :

- Connaître les composants principaux de Talend
- Adopter de bonnes pratiques de développement et d'organisation

tDBRow ✓

tJava

tDBInput



CORRECTION

40. Lors d'un lookup, comment sont stockées les données de celui-ci par défaut ? * (2 points)

Objectif opérationnel : Connaitre l'interface et le fonctionnement de Talend

- en mémoire vive ✓
- dans un fichier plat temporaire

De manière Générale, les logiciels utilisent au possible la mémoire vive. Cela permet de gagner en performance : l'accès en lecture et en écriture est plus rapide en RAM que sur disque.

A group of approximately 15 young adults, both men and women, are gathered together in a festive environment. They are smiling and posing for a photo. In the background, there are several balloons, including yellow and blue ones. The overall atmosphere is celebratory and friendly.

MERCI POUR VOTRE PARTICIPATION

À BIENTÔT !

ACSSI

DATA
MANAGEMENT
ERP

www.acssi.fr





Avez-vous des questions ?



CE FÛT UN PLAISIR
A BIENTÔT !

HISTORIQUE DU DOCUMENT

Version	Date	Auteur
7.2	Janvier 2020	Romaric Caron
7.3	Août 2021	Naima Iguemraou
7.3	Novembre 2021	Thierno BAH
7.3	Janvier 2022	Tony MALFOY, correction
8.0	Mars 2022	Alexandre PIAT
8.0	Décembre 2022	Alexis Taine, Alexandre PIAT, mise à jour
Cloud	Septembre 2025	Alexandre PIAT