

Stratégie d'importation des données – Application Salesforce LTP



1. Introduction

Ce livrable décrit la **stratégie globale d'importation des données** dans Salesforce pour l'application Lightning LTP. Il couvre :

- Les **imports initiaux** (one-shot)
- Les **flux récurrents automatisés** (transporteurs)
- L'**intégration temps réel par API REST**
- Les **outils techniques**, les **tests**, les **logs** et la **sécurité**

2. Typologie des données importées

Type de données	Objet(s) Salesforce	Volumétrie estimée
Comptes	Account	~2 123 000
Contacts	Contact, AccountContactRelation	~3 239 870
Opportunités	Opportunity	~234 000
Produits	Product2, Pricebook2, PricebookEntry	~56 700
Produits vendus	OpportunityLineItem	≥ Opportunités
Livraisons	Livraison__c	N/A (flux continu/API)
Transporteurs	Transporter_Config__c	3 (paramétrage technique)

3. Outils d'import utilisés

Outil / Plateforme	Rôle principal	Utilisation LTP
Salesforce Data Loader CLI	Import massif initial (CSV)	Chargement comptes, produits, opp...
Salesforce Workbench	Requêtage / vérification / test	Contrôles post-import, tests API REST/BULK
REST API / BULK API	Import automatisé	Intégré dans Talend / Webservice personnalisé
Talend Open Studio	Orchestration et transformation	Intégration CSV transporteurs (Job toutes les 2h), appels API Salesforce
Mulesoft (optionnel)	Alternative intégration REST	Orchestration si besoin de bus de données unifié
Webhook / Platform Event	Notification entrée / sortie	Option push retour de statut livraison
Apex Webservice	Point d'entrée Salesforce	Endpoint REST pour recevoir CSV ou trigger Talend

Solution intégrée :

```

flowchart TD
    subgraph "Temps Réel (API)"
        A[Webhook] --> B[Platform Event]
        B --> C[Flow]
        C --> D[Livraison__c]
    end
    subgraph "Batch (CSV)"
        E[FTP] --> F[Talend]
        F --> G[Bulk API]
        G --> H[Livraison__c]
    end
    D & H --> I[(Base de Données salesforce)]
    G --> J[CSV_Import_Log__c]

```

Outils critiques :

Outil	Usage Spécifique	Avantage Clé
Talend	Transformation CSV → Bulk API	Traite 500k lignes/heure
Bulk API v2	Import massif initial	10k enregistrements/requête
Platform Events	Notifications temps réel	Découplage des systèmes

Outil	Usage Spécifique	Avantage Clé
Data Loader CLI	Import historique automatisé	Scriptable, reprise sur erreur

- **Webhooks :**

"Les webhooks sont privilégiés pour LTP France et Transport Luxe Europe pour des notifications temps réel. Configuration requise : endpoint Apex REST + Remote Site Settings."

- **Talend** : utilisé comme outil principal d'automatisation et de mapping complexe des imports, avec intégration Salesforce via les connecteurs natifs.

"Utilisation de l'API Bulk V2 pour les imports CSV, avec journalisation dans CSV_Import_Log__c. Alternative : Mulesoft en V2 si besoin d'orchestration avancée."

4. Ordre logique de chargement (Import initial)

L'ordre d'importation respecte les dépendances entre objets Salesforce :

Étape	Objet	Pourquoi
1	Product2	Produit de base
2	Pricebook2	Catalogues de prix
3	PricebookEntry	Lien Produit ↔ Catalogue (besoin des 2 IDs)
4	Account	Parents des Opportunités et Contacts
5	Contact	Lookup vers Account
6	AccountContactRelation	Junction entre les deux
7	Opportunity	A besoin de l'Account
8	OpportunityLineItem	A besoin de l'Opportunity + PricebookEntry
9	Livraison__c	Lookup vers Opportunity
10	Transporter_Config__c	Paramétrage (indépendant)

5. Méthodologie par type de flux

5.1. Import initial (massif)

Étapes	Détails techniques
Format fichiers	CSV UTF-8, séparateur ;, entête de colonne

Outil	Salesforce Data Loader (batch 50k lignes max)
Fichiers de mapping	<code>.sdl</code> (Data Loader) ou <code>.json</code> (Talend)
Recommandations	Segmenter par région ou type, vérifier les dépendances (lookup IDs), désactiver les règles métiers temporaires. Activer la gestion des erreurs et journaux pour vérification

5.2. Import CSV par lot (toutes les 2h pour Rapid International Transport)

Processus général :

```

[FTP Server]
  ↓
[Talend Job]
  ↓
[Appel Apex Webservice REST POST]
  ↓
[Salesforce : Livraison__c]
  ↳ Log → CSV_Import_Log__c

```

Détails techniques :

Élément	Description
Identifiant	<code>Tracking_Number__c</code> (champ externe sur <code>Livraison__c</code>)
Validation	Parsing avec gestion d'erreur ligne à ligne (<code>try/catch</code>)
Log technique	Objet personnalisé <code>CSV_Import_Log__c</code> avec détail fichier et erreurs
Sécurité	FLS activé, champs comme <code>CSV_Imported__c</code> en lecture seule
Accès utilisateur	Profils "Support" → lecture seule uniquement

5.3 Suivi temps réel (API REST pour LTP France et Transport Luxe Europe)

```

// Classe Apex avec reprise sur erreur
public with sharing class DeliveryAPI {
    @Future(callout=true)
    public static void refreshStatus(String trackingNum) {
        try {
            HttpRequest req = new HttpRequest();
            req.setEndpoint('callout:TransportFrance_NC/track?ref='+trackingNum);
            req.setMethod('GET');

```

```

        HttpResponse res = new Http().send(req);
        // [...]
    } catch(Exception e) {
        EventBus.publish(new API_Error_Event__e(
            Message__c = 'Erreur : ' + e.getMessage(),
            Tracking__c = trackingNum
        ));
    }
}

```

Processus de consultation :

- Bouton Lightning ou Process Builder déclenche un appel à :
 - `TransporterCalloutService.fetchDeliveryStatus(String trackingNumber)`
- La méthode appelle l'endpoint REST :

```
GET https://api.transport-france.fr/track?ref=XYZ
```

- Réponse traitée dans Apex (JSON parsing)

Authentification :

- Via **Named Credentials**
- Ou clé API dans en-tête HTTP (`Authorization: Bearer`)

6. Intégration automatisée avec Talend

Architecture du Job :

```

flowchart TB
    tSftpInput --> tMap
    tMap --> tFilterRow
    tFilterRow --> tSalesforceOutputBulk
    tFilterRow --> tLogRow_Erreurs
    tSalesforceOutputBulk --> tSalesforceOutputBulkExec
    tSalesforceOutputBulkExec --> tFileOutputJSON_Logs

```

Composants clés :

- `tSalesforceOutputBulkExec` : Appel Bulk API asynchrone
- `tMap` : Transformation des champs

- `tFileOutputJSON_Logs` : Export des erreurs vers rapport
- Activer l'option *upsert* avec `Tracking_Number__c` comme External ID.

7. Bonnes pratiques d'import

Bonnes Pratiques

Sécurité :

- Chiffrement des `Named Credentials`
- `WITH SECURITY_ENFORCED` dans les requêtes SOQL
- Audit Trail activé


Performance :

- Index sur `Tracking_Number__c`
- Mode parallèle dans Bulk API
- Limite de 100 appels API/min par transporteur

Domaine	Bonnes pratiques
Qualité des données	Encodage UTF-8, formats de date normalisés, types validés
External Ids / Mapping	Toujours mapper les objets enfants via <code>External Id</code>
Performance	Utiliser BULK API v2, activer <code>parallel mode</code> dans Data Loader
Sécurité	Masquer les règles de validation / triggers lourds pendant l'import
Logs	Logging dans Talend
Rollback /Test	Tester sur sandbox, sandbox-only pour <code>Hard Delete</code>

8. Exemples pratiques

Exemple de mapping CSV (Livraison__c)

Colonne CSV	Champ Salesforce	Type	 Exemple de log (CSV_Import_Log__c)	
<code>tracking_id</code>	Tracking_Number__c	External ID		
<code>status</code>	Status__c	Picklist	Champ	Exemple
<code>livraison_date</code>	Delivery_Date__c	Date	Date_Import__c	<code>2025-06-27 14:00</code>
<code>zone</code>	Zone__c	Picklist	Status__c	<code>Succès</code>

Colonne CSV	Champ Salesforce	Type	Champ	Exemple
transporteur_ref	Transporter_Config__c	Lookup	Fichier_Nom__c	livraisons_270625.c
			Message_Erreur__c	Transporter unknow ou vide

9. Tests & Validation

Scénarios obligatoires :

Cas	Outil de Test	Résultat Attendu
Import CSV 500k lignes	Talend Sandbox	Durée < 15 min
Appel API timeout	HttpCalloutMock	Erreur journalisée
Tracking Number dupliqué	Workbench	Merge automatique

Couverture minimale :

- 90% pour les classes Apex d'intégration
- 100% des handlers d'erreurs

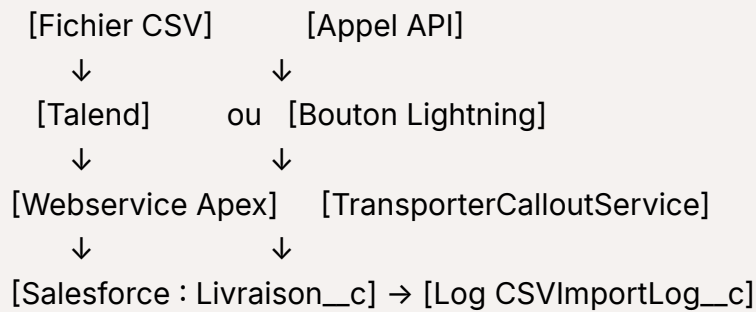
10. Suivi, tableau de bord et audit

- Création d'un **dashboard "Import Monitor"** :
 - Nombre d'import par jour
 - Taux de rejet
 - Dernier import réussi
- **Alertes email** en cas d'échec :
 - Utilisation de **Flow** ou **Apex Exception Email**
- Contrôles ponctuels avec Workbench :

11. Sécurité, gouvernance et conformité

Mesure	Détail
FLS/CRUD	Vérification dans Apex (isAccessible() , isUpdateable())
Champ External ID	Tracking_Number__c (Unique, obligatoire)
Logs	Objets personnalisés + CSV
Accès restreint	Aucun accès utilisateur direct au fichier CSV ou champ CSV_Imported__c
Custom Metadata	Centralisation des paramètres transporteurs et API

12. Schéma d'import global (texte)



13. Conclusion

Cette stratégie d'importation assure à LTP :

- Une **robustesse** des traitements avec journalisation complète
- Une **scalabilité** : support de millions d'enregistrements
- Une **sécurité stricte** (champ externe, règles de partage)
- Une **souplesse** : intégration REST/CSV/Webhook, outils low-code ou ETL
- Une **traçabilité** claire pour l'audit et le support

Elle garantit une **interopérabilité totale avec les transporteurs** et s'aligne avec les bonnes pratiques Salesforce (gouvernance API, séparation logique, logique idempotente).

Roadmap V2 :

- Migration vers **Mulesoft** pour l'orchestration centralisée
- Intégration de **webhooks entrants** pour les mises à jour push
- Portail client enrichi avec historique des livraisons