



# Rapport : Traitement des Transactions avec Spark et MinIO

Lucas CHIPAN  
Lyes BOUMRAH

## 1 Introduction

L'objectif de ce projet est d'intégrer un **pipeline de traitement des données** en temps réel en utilisant :

- **Kafka** pour la réception des transactions.
- **Apache Spark** pour la transformation et le stockage des données.
- **MinIO** comme solution de stockage objet en format **Parquet**.

Ce document détaille l'**implémentation Scala avec Spark Streaming** et l'**envoi des fichiers traités vers MinIO**.

## 2 Architecture du Système d'Information (SI)

Le système est composé de :

- **Kafka** : Diffusion des transactions en temps réel.
- **Spark Streaming** : Transformation des données.
- **MinIO (S3)** : Stockage des fichiers Parquet transformés.

## 3 Implémentation de l'Étape 3 - Traitement et Stockage

### 3.1 Traitement des transactions

1. **Connexion à Kafka** : Récupération des transactions en temps réel.
2. **Transformation des données** :
  - Conversion des montants USD en EUR.
  - Conversion des dates et application du fuseau horaire Europe/Paris.
  - Suppression des transactions erronées et des valeurs nulles.
3. **Stockage sur MinIO** : Sauvegarde des transactions sous format **Parquet compressé (Snappy)**.

### 3.2 Explication du Processus

- **Lecture en continu** des messages Kafka avec Spark Streaming.
- **Transformation des données** pour assurer la cohérence et la qualité des informations.
- **Stockage dans MinIO** sous format **Parquet**, optimisé pour l'analyse de données.

### 3.3 Résultats Attendus

Les transactions avant transformation contiennent des devises différentes et des dates au format texte.

Après transformation :

- Les montants sont convertis en EUR.
- Les dates sont standardisées avec le fuseau horaire Europe/Paris.
- Les transactions erronées sont filtrées.

Les fichiers sont stockés sur **MinIO** dans **bigdata/transactions/** et peuvent être analysés avec Spark.

bigdata			Rewind ↶	Refresh ↻	Upload ↗
Created on: Thu, Feb 06 2025 23:00:09 (GMT+1) Access: PRIVATE 1.5 MiB - 14 Objects					
bigdata / transactions			Create new path ↗		
<input type="checkbox"/>	Name	Last Modified	Size		
<input type="checkbox"/>	_spark_metadata		-		
<input type="checkbox"/>	checkpoint		-		
<input type="checkbox"/>	part-00000-060cad5b-a26b-4f1f-a39d-2b6aab50847b-c000.snappy....	Today, 14:24	466.8 KiB		
<input type="checkbox"/>	part-00000-d476fe00-0664-4103-9db2-ed5b5c3e5d8f-c000.snappy...	Today, 19:31	57.4 KiB		
<input type="checkbox"/>	part-00001-683a215c-33a6-4d82-a9e8-f27524403513-c000.snappy....	Today, 14:24	467.9 KiB		
<input type="checkbox"/>	part-00001-6c3e9602-96cf-43c7-8688-4ff735cd4072-c000.snappy....	Today, 19:31	56.2 KiB		
<input type="checkbox"/>	part-00002-137619af-fba1-4db9-8be3-f37555ab66b2-c000.snappy.p...	Today, 19:31	56.4 KiB		
<input type="checkbox"/>	part-00002-c632b5d9-d1ed-45cc-95c5-88b005052751-c000.snappy...	Today, 14:24	470.9 KiB		

	idTransaction	typeTransaction	montant	...	moyenPaiement	details	utilisateur
0	a7799563-2811-4f0d-b167-b1fa38ac0b5d	achat	279.704895	...	{}	[]	[]
1	ed5b113c-a633-45ec-99d2-ecf6c67ad781	achat	92.232221	...	{}	[]	[]
2	b296a460-4c70-4916-bf4e-18f94c0d71d9	achat	234.337880	...	{}	[]	[]
3	cd871c7a-c53a-476f-b3e4-4deba637e580	remboursement	82.398349	...	{}	[]	[]
4	3772e080-cfba-4813-b5a1-44748eac5d6d	achat	529.384865	...	{}	[]	[]

## 4 Conclusion

- Implémentation du **pipeline Kafka → Spark → MinIO**.
- Stockage des transactions en **Parquet compressé (Snappy)**.
- Possibilité de **relire les données avec Spark**.