# Fiche du cours

**60 h**

**Titre :**

Jav330 - Big Data & Streaming (Java)

**Description :**

Ingérer, transformer et diffuser des flux de données en Java avec Apache Kafka et Kafka Streams (avec aperçu Flink). On couvre la modélisation d'événements, la sérialisation (Avro + Schema Registry), la gestion d'état et des fenêtres (tumbling/hopping/session), les garanties de livraison (at-least/exactly-once – notions), les tests d'intégration (Testcontainers), ainsi que l'observabilité (logs/metrics) et l'emballage Docker vers un sink (DB/Parquet).

**Objectifs :**

- Produire/consommer avec Kafka (topics, partitions, clés, ordering, rebalancing).*
- Écrire des pipelines en Kafka Streams (topologie, state stores, fenêtrage).*
- Gérer les schémas avec Avro et Schema Registry (compatibilité, évolution).*
- Assurer la sérialisation/désérialisation (Serdes) et la qualité des événements.*
- Tester en conditions réalistes avec Testcontainers (Kafka/Schema Registry/DB).*
- Observer (logs/metrics) et packager (Docker) un pipeline vers DB/Parquet.

**Chapitres :**

1. Kafka – fondamentaux : topics/partitions, réplication, clés, offsets, groupes de consommateurs*
2. Producers & Consumers : idempotence, acks, retries, rééquilibrage, DLT (dead letter topic)*
3. Schémas & Avro : Schema Registry, compatibilité (backward/forward), Serdes*
4. Kafka Streams I : DSL, topologies, KStream/KTable, joins (stream–stream, stream–table)*
5. Kafka Streams II : fenêtres (tumbling/hopping/session), state stores, exactly-once (notions)*
6. Flink (aperçu) : job, operators, state & checkpoints, quand préférer Flink à Kafka Streams*
7. Sinks & stockage : vers base relationnelle (JDBC) ou Parquet (batching, schéma, partitionnement)*
8. Tests d'intégration : Testcontainers (Kafka/Schema Registry/DB), datasets de test, assertions sur flux*
9. Observabilité & Ops : logs structurés, metrics (JMX/Micrometer), alertes simples ; Docker Compose (Kafka + app + DB)*
10. Capstone : pipeline temps réel (Kafka → processing → store) + tests & metrics, README & scripts de run

| **À la fin :** |
| --- |
| Vous saurez construire un pipeline streaming en Java : produire/consommer sur Kafka, transformer avec Kafka Streams (état & fenêtres), valider les schémas (Avro/Registry), tester avec Testcontainers, observer (logs/metrics) et déployer sous Docker vers un sink (DB/Parquet) — un livrable portfolio-ready. |