Talend for Data Integration

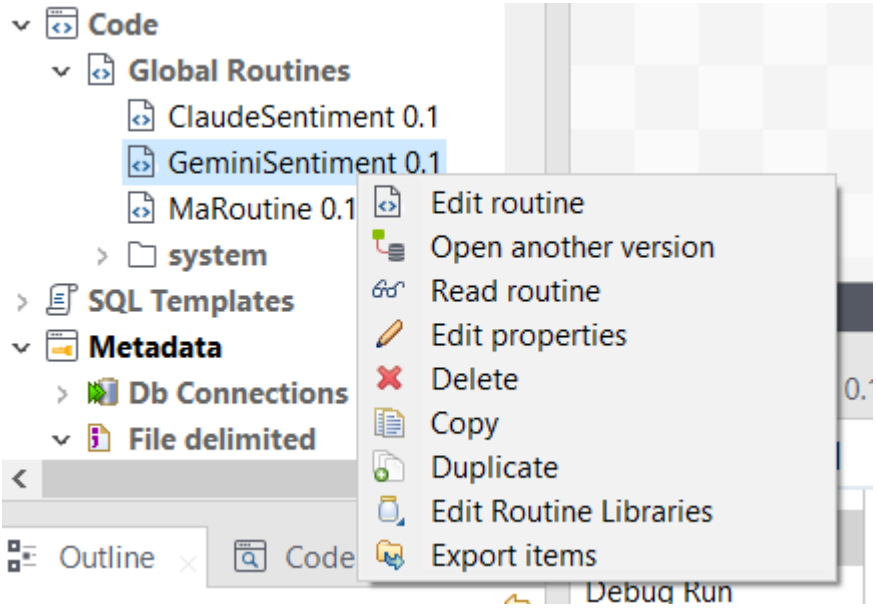
Trainer : Sellami Mokhtar
mokhtar.sellami@data2-ai.com

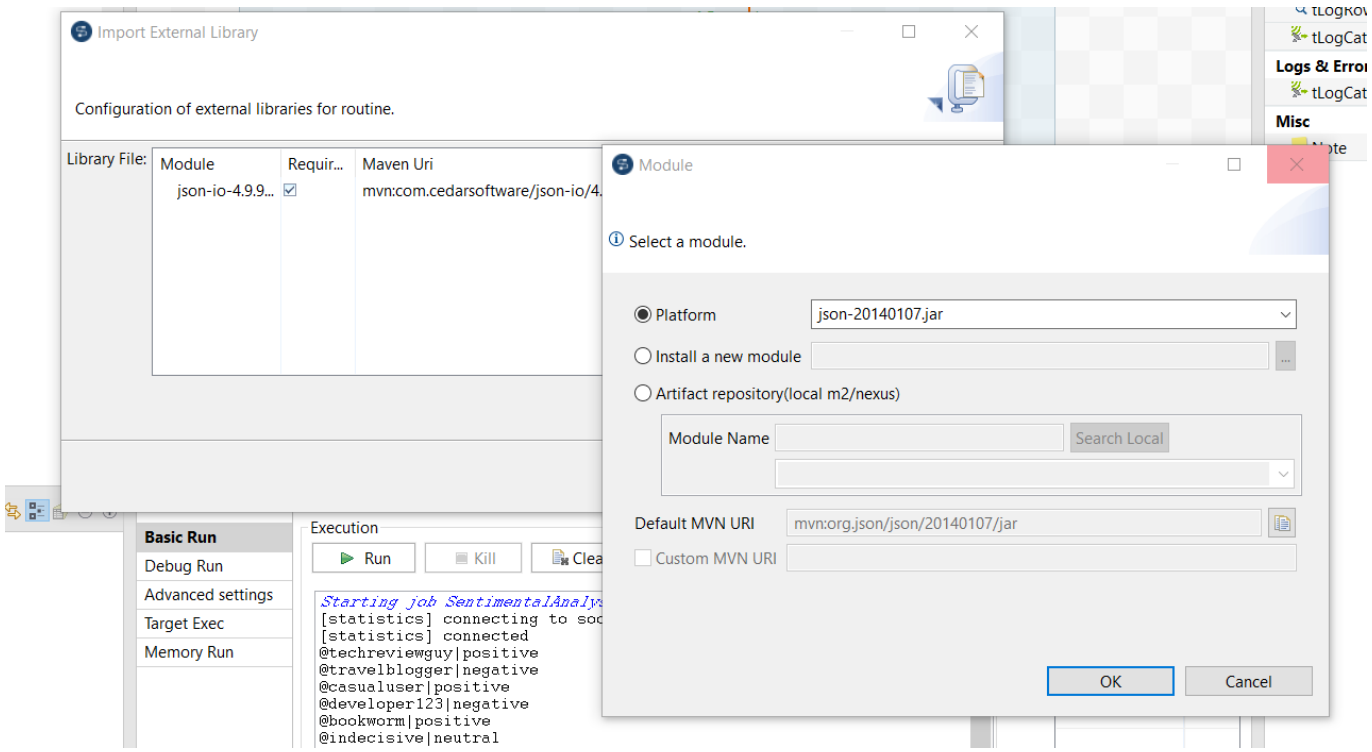
Implémentation de l'analyse de sentiment des tweets dans Talend avec Gemini

Ce guide vous explique étape par étape comment créer une solution Talend utilisant l'API Gemini de Google pour analyser les sentiments exprimés dans des tweets. Chaque étape peut être illustrée par une capture d'écran pour faciliter la compréhension.

☑ Prérequis

1. **Talend Open Studio** ou **Talend Data Integration 8.x**
2. Une **clé API Gemini** (créez-en une sur https://aistudio.google.com/app/apikey)
3. La bibliothèque `json-io-4.9.9-TALEND.jar` ajoutée dans le chemin des bibliothèques de Talend (`Code > Global Routines > Edit Routines Librairies`)





Données:

```
id,tweet_text,user,timestamp
1001,"Just tried the new iPhone 15 Pro Max and I'm absolutely blown away by the
camera quality! Best purchase this year! #Apple #iPhone",@techreviewguy,2025-04-
28T09:15:23
1002,"Flight delayed for the third time today. Terrible customer service and no
information provided. Never flying with this airline again.
#frustrated",@travelblogger,2025-04-28T11:42:56
1003,"It's raining outside today. Might be a good day to stay in and watch
movies.",@casualuser,2025-04-28T13:30:12
1004,"The latest software update completely bricked my device. Hours of work lost
and support isn't responding. This is unacceptable!",@developer123,2025-04-
28T14:22:45
1005,"Just finished reading 'The Psychology of Money'. Highly recommend it for
anyone looking to improve their financial mindset!",@bookworm,2025-04-28T16:05:37
1006,"Can't decide whether to go to the beach or the mountains this weekend. Any
suggestions?",@indecisive,2025-04-28T17:11:09
1007,"Stock market is all over the place today. Some big gains in tech but losses
in energy sector.",@investorpro,2025-04-28T09:45:18
1008,"This coffee shop has the most amazing pastries I've ever tasted! The
atmosphere is so cozy too. Definitely coming back! ☕♡",@foodiefan,2025-04-
28T10:30:22
1009,"Monday meetings are the worst. Two hours of my life I'll never get
back.",@officeworker,2025-04-28T11:15:00
1010,"Just adopted a rescue puppy! She's the sweetest thing ever. Can't wait to
bring her home tomorrow! #rescuedogs #adoption",@dogperson,2025-04-28T18:25:41
```

📌 Étape 1 : Créer la Routine Java `GeminiSentiment`

1. Ouvrez **Talend Studio** et ouvrez votre projet
2. Clic droit sur le dossier **Code > Routines** dans le panneau Repository
3. Sélectionnez **Create routine**
4. Donnez-lui le nom `GeminiSentiment`, puis cliquez sur **Finish**
5. Remplacez le code par défaut par la version corrigée du code Java

```
package routines;

import java.io.IOException;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
import java.time.Duration;

import org.json.JSONArray;
import org.json.JSONObject;

/**
 * GeminiSentiment
 *
 * Analyzes tweet sentiment using Google Gemini API.
 */
public class GeminiSentiment {

    public static final String SENTIMENT_POSITIVE = "positive";
    public static final String SENTIMENT_NEGATIVE = "negative";
    public static final String SENTIMENT_NEUTRAL = "neutral";

    /**
     * Analyzes the sentiment of a tweet using Gemini API
     *
     * @param tweet  The tweet text
     * @param apiKey The Google API Key
     * @return "positive", "negative", or "neutral"
     */
    public static String analyzeSentiment(String tweet, String apiKey) {
        if (tweet == null || tweet.trim().isEmpty()) {
            return SENTIMENT_NEUTRAL;
        }

        try {
            // Request body with correct Gemini v1beta format
            String requestBody = new JSONObject()
                .put("contents", new JSONArray()
                    .put(new JSONObject()
                        .put("parts", new JSONArray()
                            .put(new JSONObject()
                                .put("text", "Analyze the sentiment of this tweet
and respond with exactly one word: 'positive', 'negative', or 'neutral'. The tweet
is:\n\n" + tweet)
                            )
```

```
                        )
                    )
                )
                .put("generationConfig", new JSONObject()
                    .put("temperature", 0)
                    .put("maxOutputTokens", 10)
                )
                .toString();

            HttpClient client = HttpClient.newBuilder()
                .connectTimeout(Duration.ofSeconds(30))
                .build();

            HttpRequest request = HttpRequest.newBuilder()

.uri(URI.create("https://generativelanguage.googleapis.com/v1beta/models/gemini-
1.5-flash:generateContent?key=" + apiKey))
                .header("Content-Type", "application/json")
                .POST(HttpRequest.BodyPublishers.ofString(requestBody))
                .build();

            HttpResponse<String> response = client.send(request,
HttpResponse.BodyHandlers.ofString());

            if (response.statusCode() != 200) {
                System.err.println("API Error: " + response.statusCode() + " - " +
response.body());
                return SENTIMENT_NEUTRAL;
            }

            JSONObject jsonResponse = new JSONObject(response.body());
            String content = jsonResponse
                .getJSONArray("candidates")
                .getJSONObject(0)
                .getJSONObject("content")
                .getJSONArray("parts")
                .getJSONObject(0)
                .getString("text")
                .toLowerCase()
                .trim();

            if (content.contains("positive")) {
                return SENTIMENT_POSITIVE;
            } else if (content.contains("negative")) {
                return SENTIMENT_NEGATIVE;
            } else {
                return SENTIMENT_NEUTRAL;
            }

        } catch (IOException | InterruptedException e) {
            System.err.println("Error calling Gemini API: " + e.getMessage());
            e.printStackTrace();
            return SENTIMENT_NEUTRAL;
        }
```

```
    }

    // Main method for local testing
    public static void main(String[] args) {
        String apiKey = "YOUR_GEMINI_API_KEY"; // Replace with your actual API key

        String[] tweets = {
            "I absolutely love the new product! It's amazing and works perfectly!
#happy",
            "This service is terrible. I've been waiting for hours and nobody is
helping me. #frustrated",
            "The weather today is cloudy with some rain expected later."
        };

        for (String tweet : tweets) {
            System.out.println("Tweet: " + tweet);
            System.out.println("Sentiment: " + analyzeSentiment(tweet, apiKey));
            System.out.println();
        }
    }
}
```

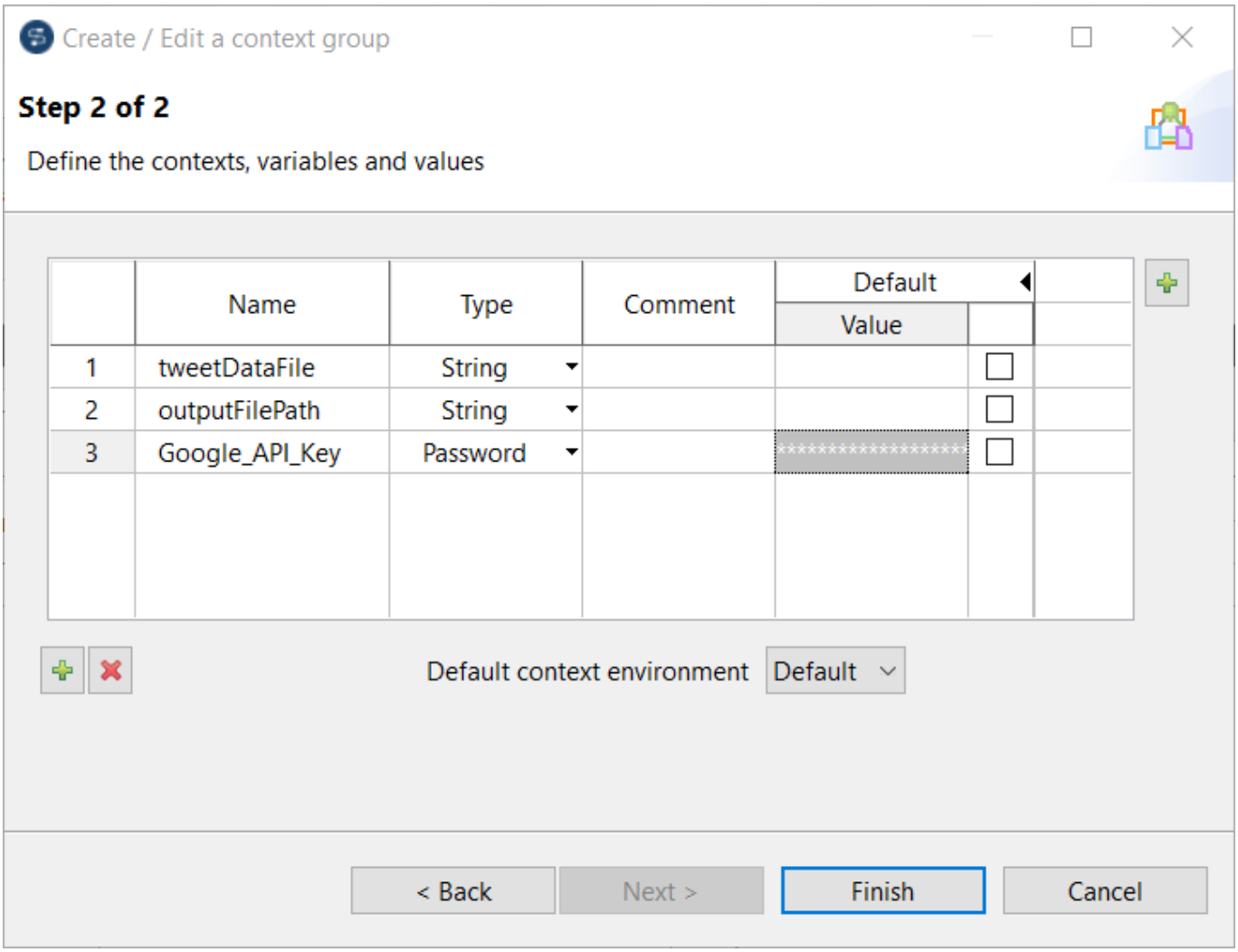
6. Cliquez sur **Save**

💡 Ce que fait la routine :

- Envoie une requête à l'API Gemini de Google
- Analyse la réponse JSON
- Retourne une valeur parmi : `"positive"`, `"negative"` ou `"neutral"`

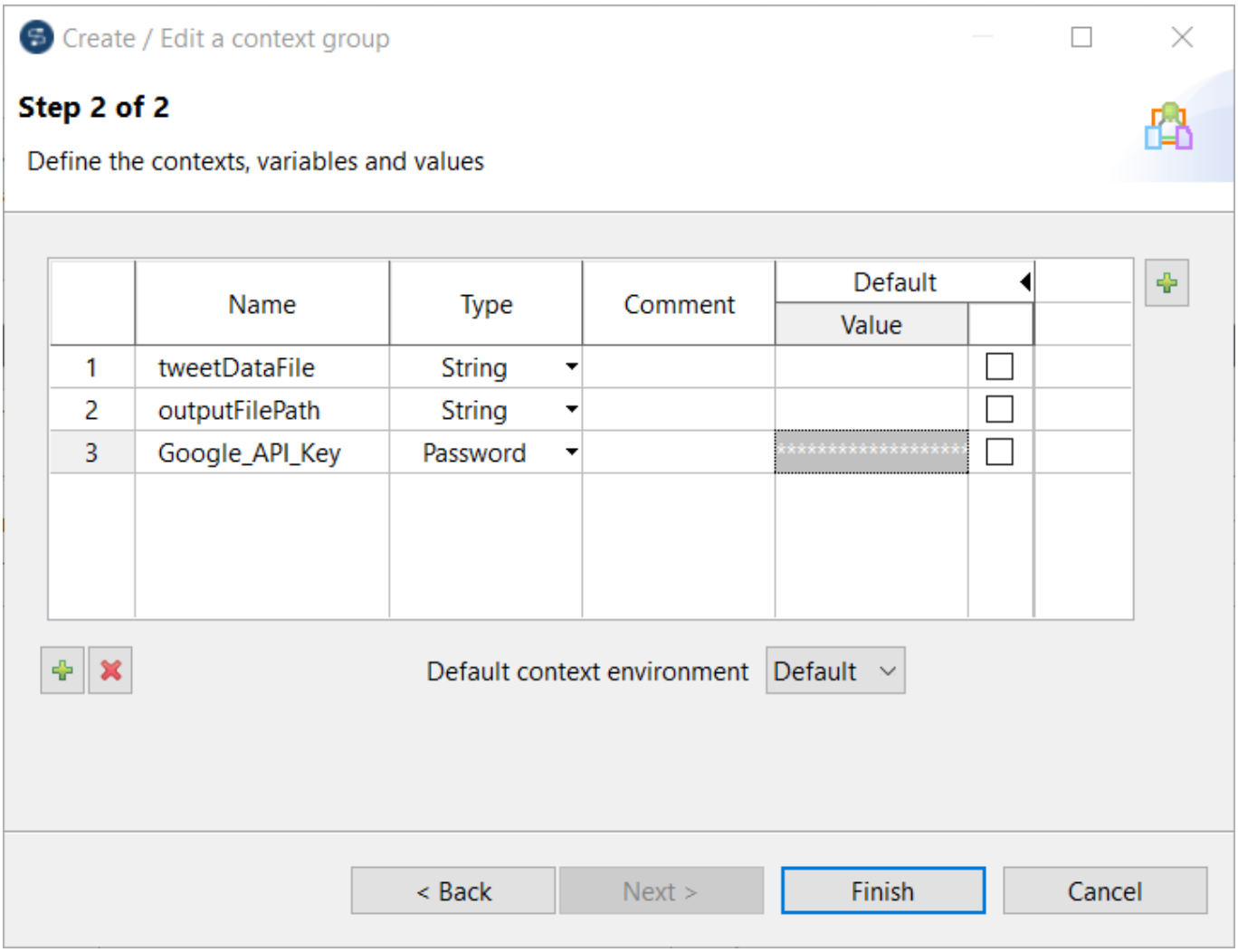
📌 Étape 2 : Définir les variables de contexte

1. Clic droit sur **Contexts** dans le panneau Repository
2. Sélectionnez **Create context group**
3. Nommez-la `SentimentAnalysisConfig`
4. Ajoutez les variables suivantes :
    - `tweetDataFile` (String) : chemin du fichier CSV avec les tweets
    - `outputFilePath` (String) : chemin pour le fichier de sortie
    - `geminiApiKey` (Password) : clé API Gemini



📌 Étape 3 : Créer le Job Talend

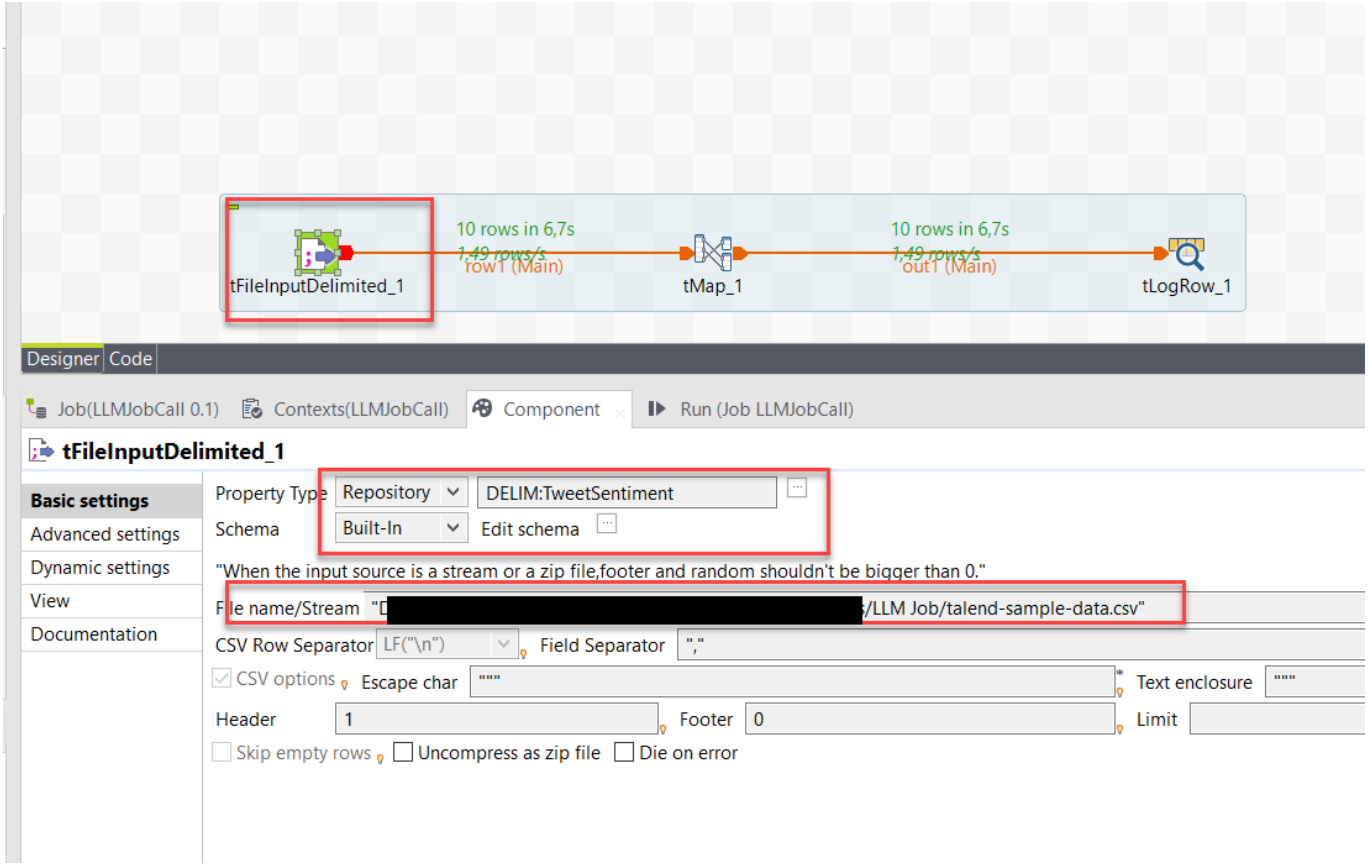
1. Clic droit sur **Job Designs** > **Create job**
2. Donnez-lui le nom `TweetSentimentAnalysis`
3. Glissez ces composants depuis la palette :
   - `tFileInputDelimited`
   - `tMap`
   - `tFileOutputDelimited`



Configuration des composants :

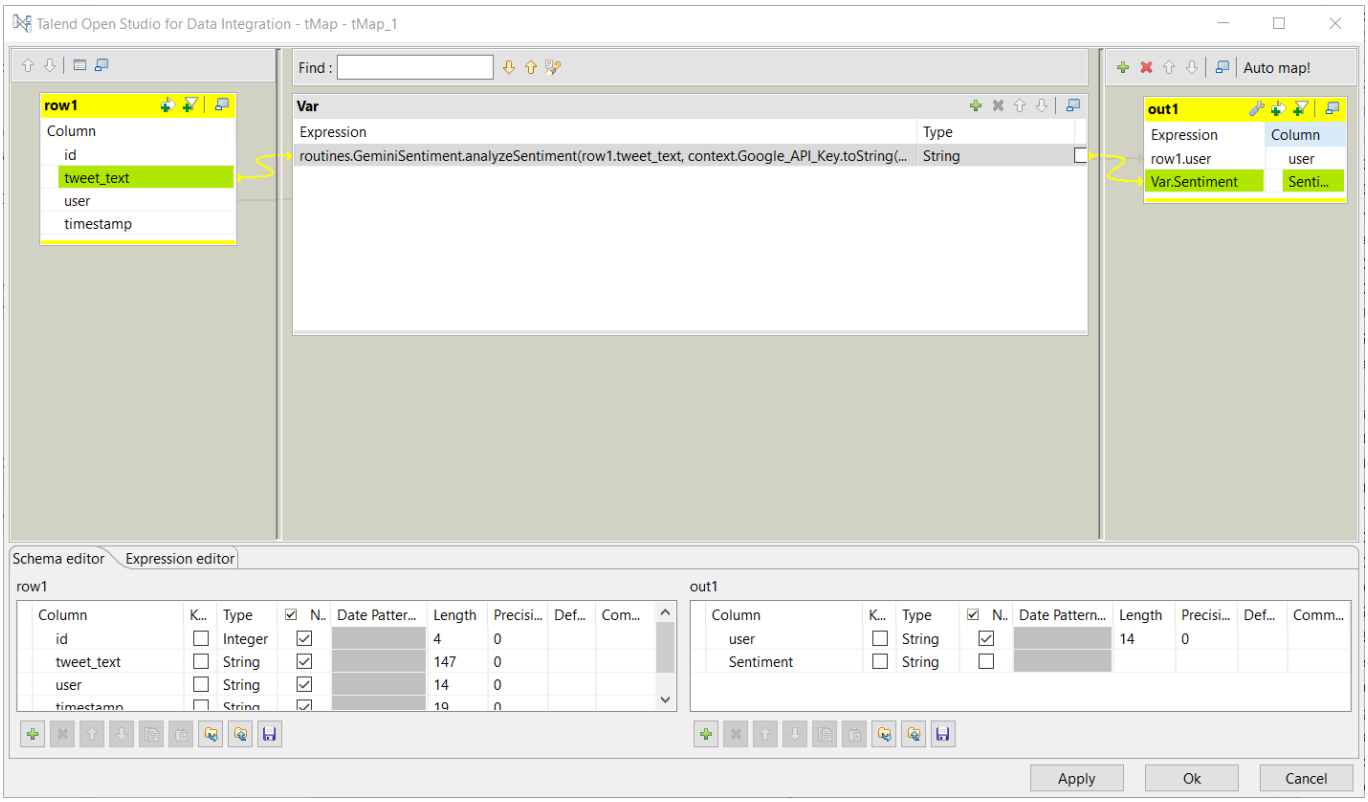
- **tFileInputDelimited**

  - File Name: `context.tweetDataFile`
  - Row Separator: `\n`
  - Field Separator: `,`
  - Header: `1`
  - Schéma :
    - id (String)
    - tweet_text (String)
    - user (String)
    - timestamp (String)



- **tMap**

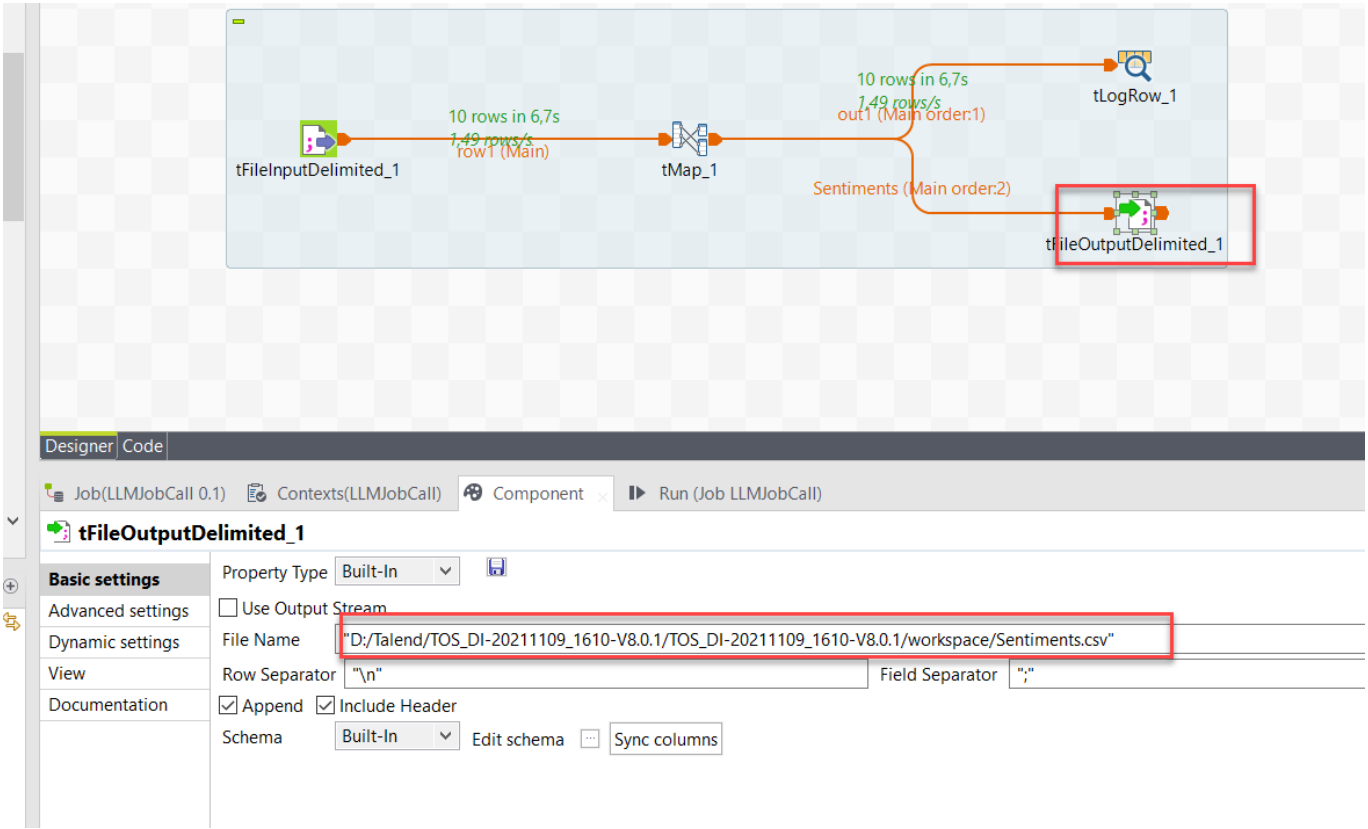
  - Créer une sortie "output"
  - Mapper les colonnes d'entrée vers la sortie
  - Ajouter une colonne `sentiment` :



```
routines.GeminiSentiment.analyzeSentiment(row1.tweet_text, context.geminiApiKey)
```

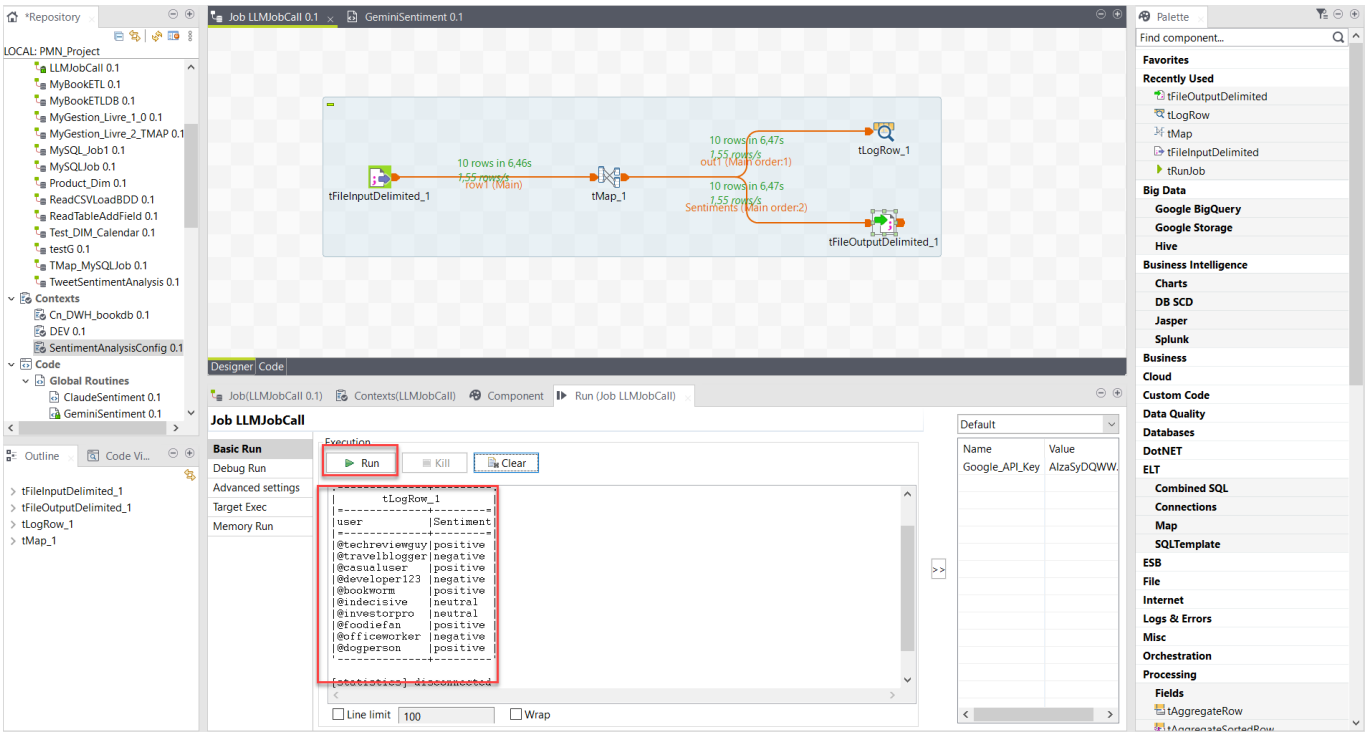
- **tFileOutputDelimited**

  - File Name: `context.outputFilePath`
  - Include Header: cocher
  - Schema : identique à la sortie de `tMap`



📌 Étape 4 : Exécuter le Job

1. Enregistrez le job
2. Clic droit > **Run**
3. Renseignez les variables de contexte
4. Surveillez l'exécution dans la console **Run**



📊 Exemple de résultat

Le fichier CSV de sortie inclura les données du tweet et la colonne `sentiment` :

Tweet	Sentiment
"J'adore le nouveau produit, il est incroyable !"	positive
"Service client horrible, j'attends depuis 2h."	negative
"Il fait gris aujourd'hui, un bon jour pour rester chez soi."	neutral

⚙️ Considérations de performance

- Les appels API prennent du temps → penser au traitement par lots ou en parallèle
- Ajouter une limitation de débit si besoin
- Pour de gros volumes, ajouter des mécanismes de relance ou de récupération

🔐 Bonnes pratiques de sécurité

1. Stocker la clé API comme variable de contexte de type `Password`
2. Chiffrer les variables de contexte en production
3. Respecter la confidentialité des données utilisateurs
4. Ne pas journaliser d'informations sensibles