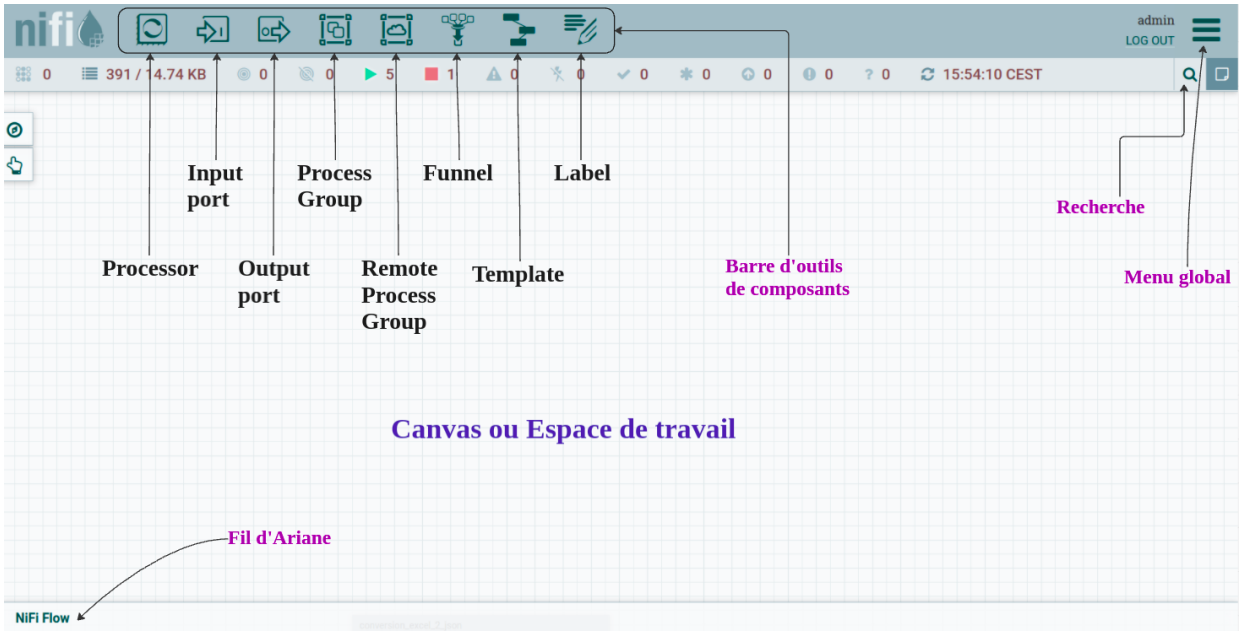
TP1 : Prise en main de l'interface utilisateur Nifi

Objectif. Ce TP vise à vous familiariser avec l'interface utilisateur Nifi et à mettre en place des pipelines pour des flux de données pour exécuter des tâches spécifiques. Dans un premier temps, une présentation succincte de l'interface sera faite à l'aide de visuels et de définitions. Ensuite, une première implémentation d'un pipeline sera proposée. Et enfin, plusieurs cas d'études seront abordés. Il est important de noter que l'objectif n'est pas de couvrir l'ensemble des fonctionnalités offertes par NiFi, en raison de leur nombre considérable, ce qui n'est pas réalisable en pratique dans le cadre de ce cours. Au contraire, l'accent sera mis sur la compréhension globale du fonctionnement de NiFi pour vous permettre d'accomplir des tâches en fonction de besoins spécifiques. Le TP se déroulera en mode standalone, c'est-à-dire, sur une instance Nifi.

Pour approfondir vos connaissances sur les fonctionnalités vues durant ce tp ou sur d'autres fonctionnalités et concepts de Nifi, consultez la documentation officielle : https://nifi.apache.org/docs/nifi-docs/html/user-guide.html

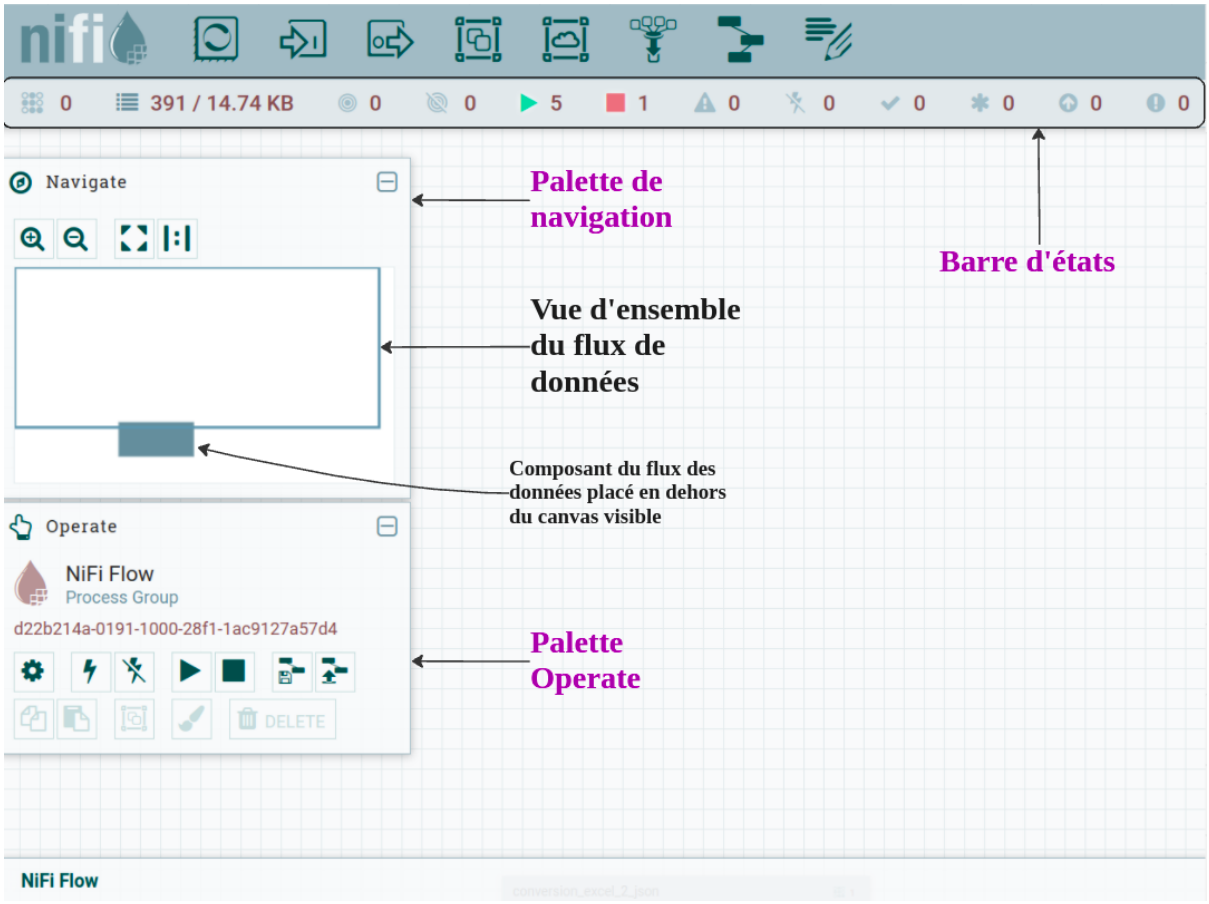
1. Interface Graphique



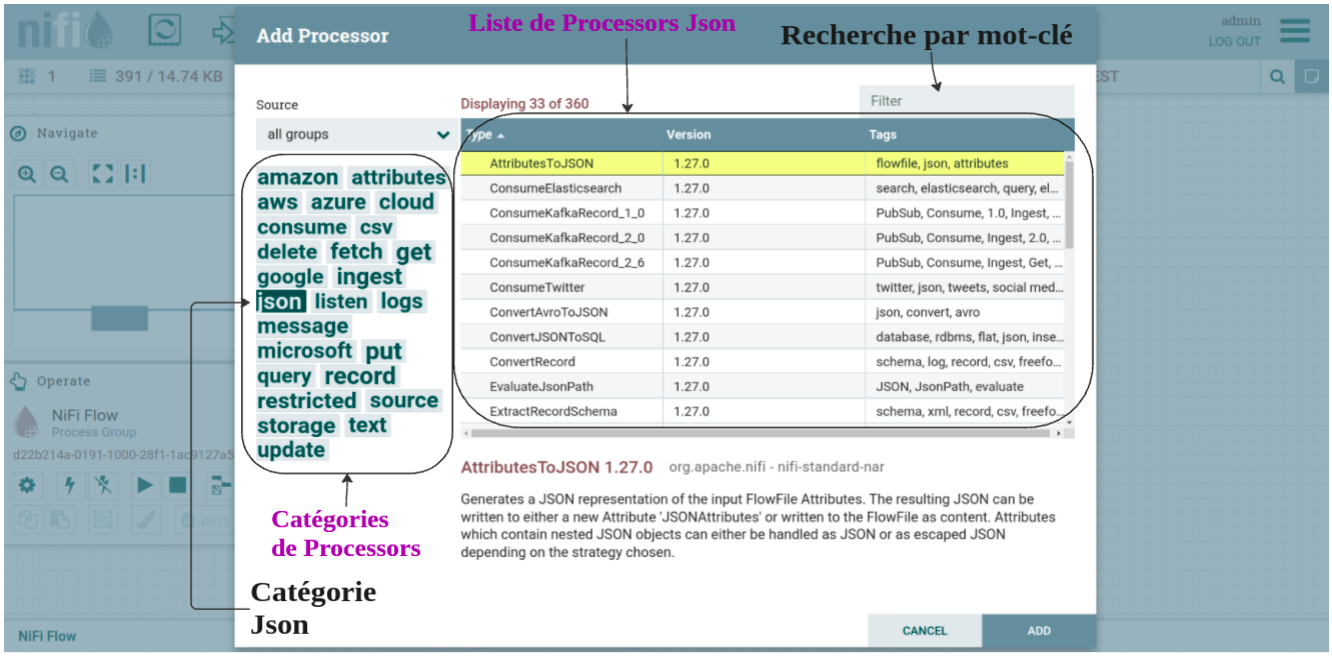
Composant	Description

Processor	Composant de traitement visant à effectuer une tâche spécifique. Il accès aux attributs et aux contenus d'un flux de données. Dans la version 1.27.0 au moins 360 composants sont présents et sont catégorisées en fonction de leur caractéristique au niveau des traitements. Un large éventail de traitements rencontrés en entreprise peuvent être effectués via les Processors Nifi.
Process Group	Composant offrant un mécanisme d'organisation de composants en une structure logique pour rendre le flux de données plus compréhensible à un niveau d'abstraction de haut niveau. Typiquement, il correspond à un ensemble de composants (généralement de type Processor) ainsi que les relations (appelées connections). Ce regroupement permet de créer de nouvelles tâches relativement complexes par rapport aux composants Processor.
Remote Process Groups	Similaire à Process Group sauf qu'au lieu de déplacer les données entre des Process Groups au sein de la même instance Nifi (typiquement un serveur/machine), elles sont transférées entre les Process Groups issus des instances NiFi différents.
Input Port	Point d'entrée de données, généralement placé au sein d'un Process Group (local/distant).
Output Port	Point de sortie de données, généralement placé au sein d'un Process Group (local/distant).
Funnel	Mécanisme de regroupement de plusieurs relations en une seule. Il facilite la gestion de flux de données multi-sources.
Template	Structure du schéma du flux de données (typiquement au format xml) pour permettre la sauvegarde et le partage des opérations effectuées sur le flux de données.

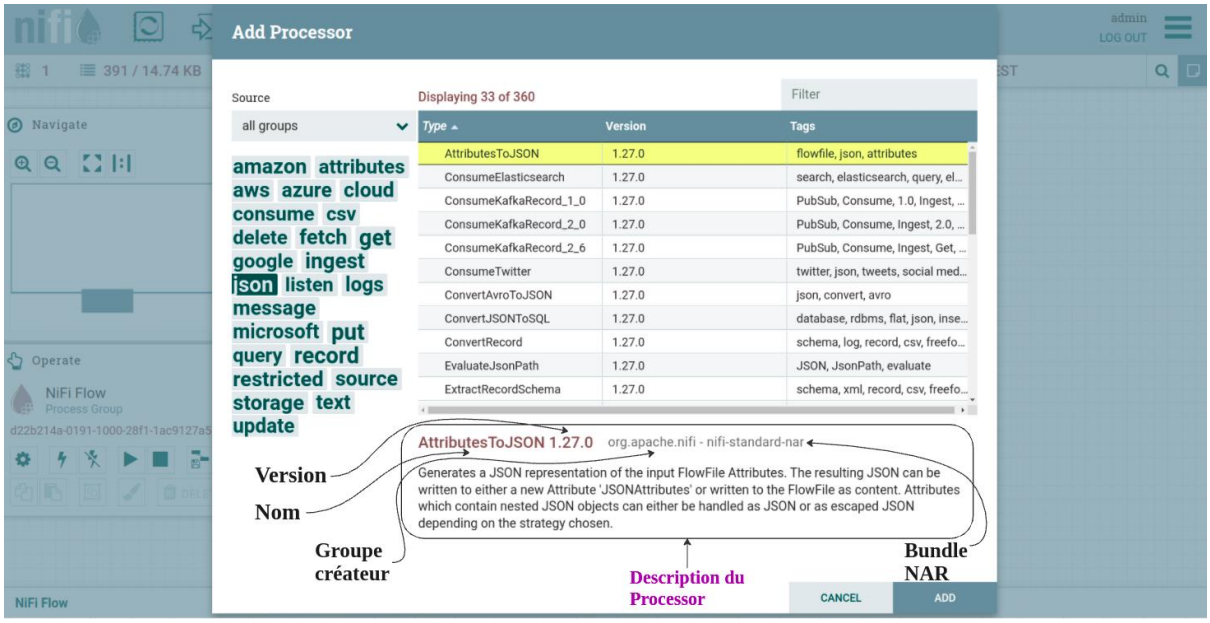
Label	Outil de description de composants présents dans l'espace de travail



Les composants sont créés dans l'espace de travail en les faisant glisser-déposer (drag-and-drop) vers le canvas. Concernant le composant Processor, une fois déposé sur le canvas, une fenêtre s'ouvre pour permettre à l'utilisateur de choisir le type de traitement à effectuer (voir l'image ci-dessous). Le Processor recherché peut être identifié via la fonction de recherche par mot-clé. Il est important de noter que les Processors sont organisés en groupes ou "tags" (amazon, azure, aws, ingest, json, etc.). Par exemple, le groupe "json" propose plusieurs Processors liés au traitement des données JSON, comme EvaluateJsonPath, ExtractRecordSchema, AttributesToJSON, etc. Dans cet exemple, seuls les Processors JSON apparaissent dans la liste grâce à l'application du filtre. Pour retirer ce filtre, il suffit de décocher l'option "json".



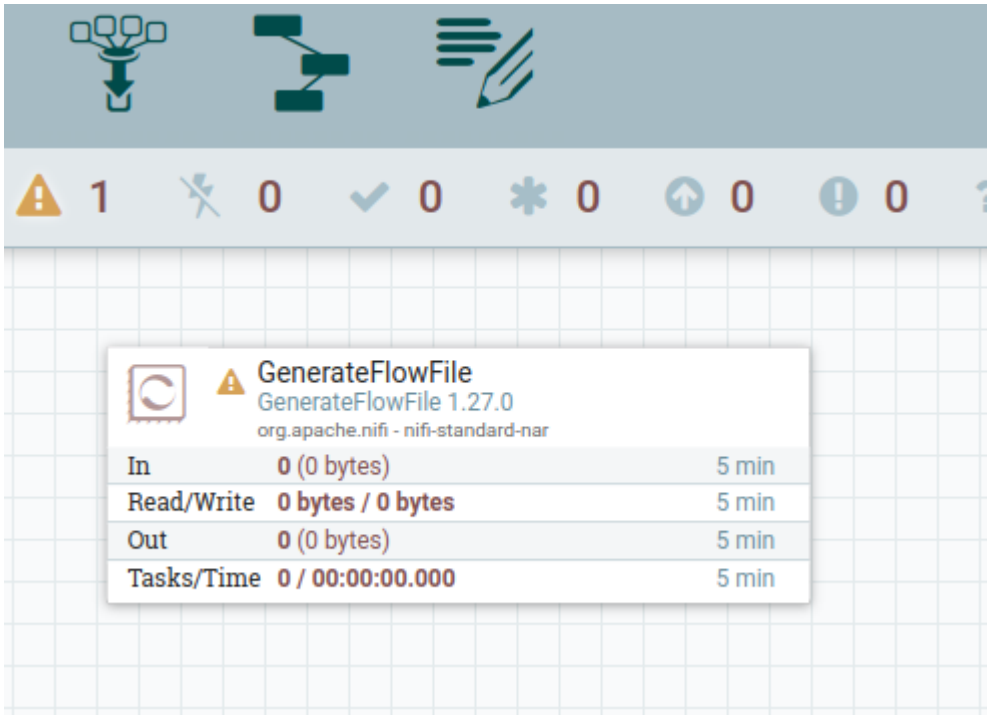
Lorsque l'on clique sur un Processor, des informations à son sujet s'affichent en bas de la fenêtre, notamment son nom, sa version et sa description (voir l'image ci-dessous). On y trouve également le groupe qui a développé le Processor ainsi que le package dans lequel il est inclus. Pour confirmer l'ajout du Processor, il suffit de cliquer sur le bouton "ADD" situé en bas à droite de la fenêtre.



2. Implémentation d’un schéma de flux de données

Dans ce qui suit, nous allons créer un schéma de flux de données dans l'espace de travail, qui générera des fichiers de flux (FlowFile), remplacera une partie de leur contenu en fonction d'une expression régulière, puis les enregistrera avec l'extension .text.

1) Pour générer un fichier de flux, créez le Processor "GenerateFlowFile", comme expliqué dans la section précédente. Son rôle est de produire des données aléatoires ou des données basées sur un schéma défini par l'utilisateur. L'interface du Processor "GenerateFlowFile" s'ouvre comme illustré dans la figure ci-dessous.

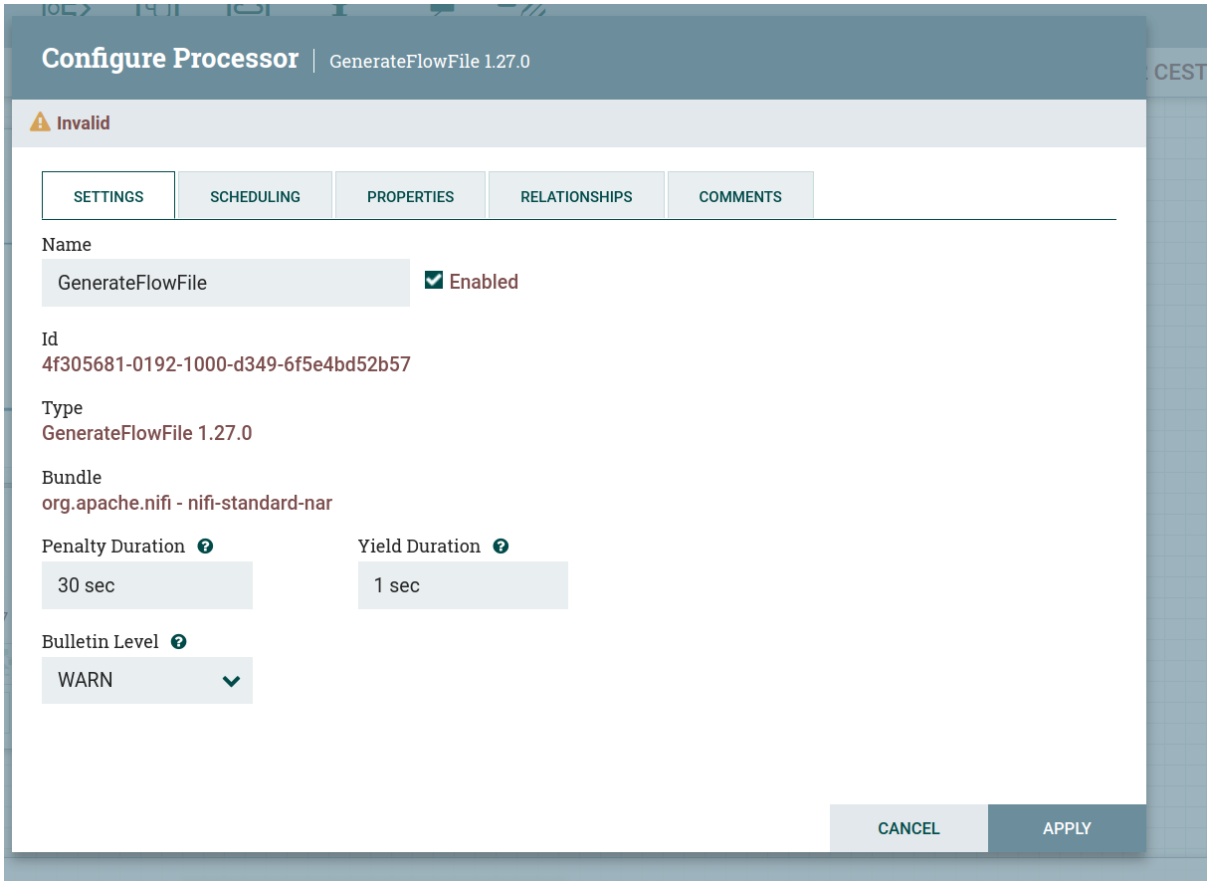


Dans NiFi, chaque processor en cours d'exécution affiche des statistiques clés dans son interface utilisateur, permettant de surveiller le flux de données. Le tableau ci-dessous présente les statistiques disponibles dans cette interface :

Métriques standards d’un Processor	Description
In	La taille (en cardinalité et en octets) des FlowFiles reçus par le processor au cours des 5 dernières minutes.

Read / Write	La quantité de données, en octets, que le Processor a lue et écrite pendant son fonctionnement.
Out	La taille (en cardinalité et en octets) des FlowFiles transférés par le processor au cours des 5 dernières minutes.
Tasks/Time	Le nombre de tâches effectuées dans les 5 dernières minutes. Le temps CPU utilisé par le Processor au cours des 5 dernières minutes. Cela mesure le temps de traitement consommé pour gérer les FlowFiles.

En double-cliquant sur l'interface du Processor, une fenêtre de configuration s'ouvre, permettant de régler certains paramètres en fonction des besoins spécifiques et des contraintes des machines hôtes, notamment en termes de mémoire, CPU et puissance de calcul.



Dans NiFi, lors de la configuration d'un Processor, il existe des champs communs à tous les processors, ainsi que des paramètres spécifiques qui dépendent du processor en question. Par exemple, dans la page de configuration “setting”, "name", "penalty duration", "yield duration" et "bulletin level" sont des champs communs.

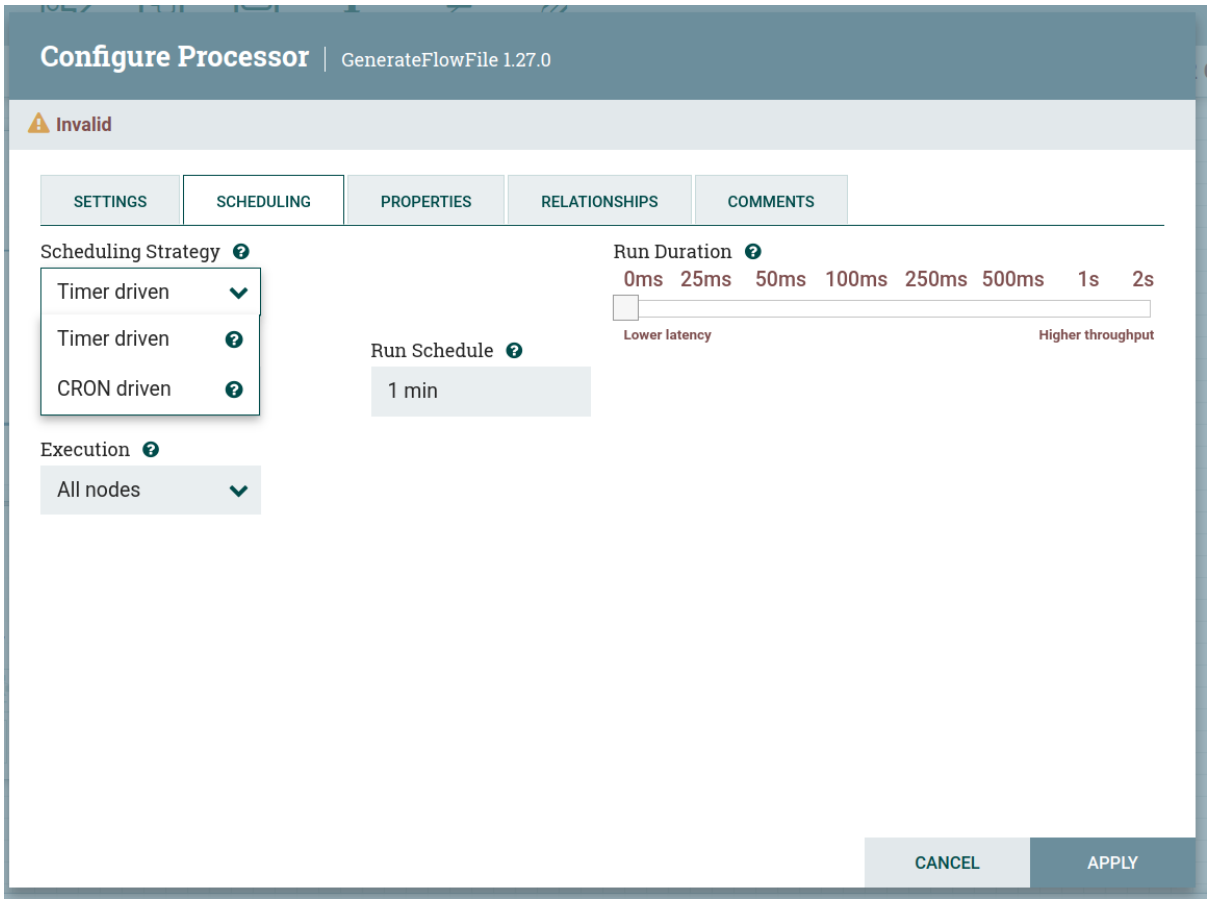
La case à cocher “Enabled” permet d’activer ou de désactiver le fonctionnement du Processor. Les champs "penalty duration" et "yield duration" contrôlent la gestion des erreurs et des tentatives de réexécution par les Processors. Penalty duration définit la durée de temps que traitement d’un Flowfile spécifique est retardé après une erreur avant d'être réessayé, tandis que yield duration détermine la durée de temps l'ensemble du Processor est mis en pause après avoir rencontré un problème, avant de tenter à nouveau de traiter des Flowfiles. Penalty duration est propre à chaque Flowfile, empêchant les réessaies immédiats des fichiers problématiques, tandis que yield duration s'applique à tout le Processor, stoppant temporairement son fonctionnement pour éviter une surcharge des ressources lors de défaillances. Ces deux paramètres aident à gérer la gestion des erreurs et l'efficacité des ressources dans les flux de données.

2) Dans notre cas, pour le moment, nous ne remplaçons que le nom actuel par "Générateur de FlowFiles".

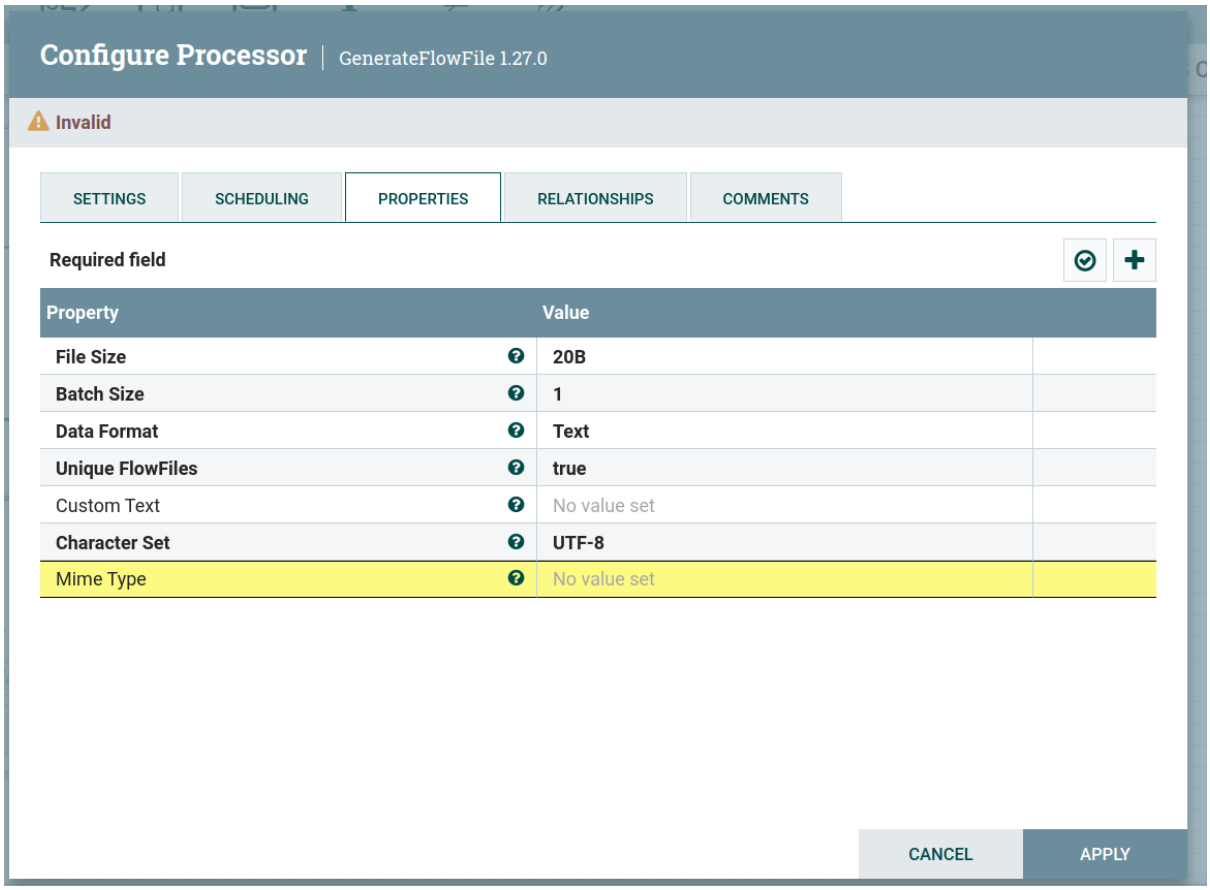
En ce qui concerne l’ordonnancement des tâches contrôlé via la page scheduling, plusieurs champs communs aux Processors sont présents (voir figure ci-dessous) : Scheduling Strategy, Run Schedule et Run Duration.

Champ	Description
Run Schedule	définit la fréquence à laquelle le processor s'exécutera (par exemple, toutes les 1 seconde ou 5 minutes).
Run Duration	spécifie la durée pendant laquelle le processor doit s'exécuter dans un cycle d'exécution unique. Ce paramètre est généralement défini pour les processors qui doivent traiter de grandes quantités de données ou gérer des tâches longues, leur permettant de continuer le traitement pendant un certain temps avant de céder le contrôle à d'autres processors ou tâches. Si ce paramètre est laissé non défini ou réglé à 0, le processor ne s'exécute que pour une seule tâche par cycle d'exécution (traitant un lot ou un fichier) avant de céder, permettant aux autres processors d'exécuter leurs tâches.

Scheduling Strategy	Champ proposant deux options : • Timer Driven : le Processor s'exécute à intervalles réguliers. • CRON Driven : S'exécute selon une expression CRON (par exemple apparition d'un évenement).



La page Properties englobe des paramètres propres au Processor de type GenerateFlowFile. Il est sous-entendu que les champs de cette page varient d'un type de Processor à l'autre.



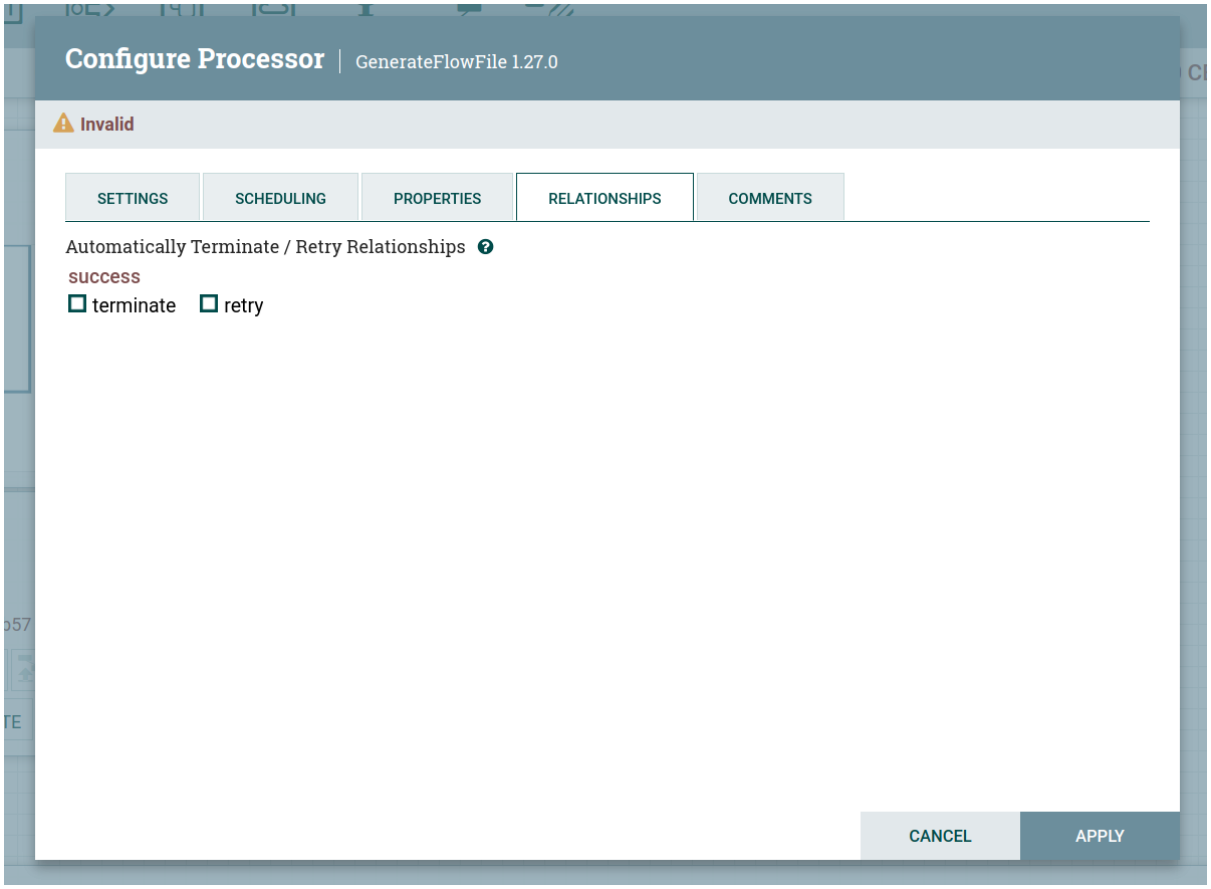
3) Fixer la taille de chacun des fichiers à 20 octets pour avoir un contenu correspondant à environ une demi-ligne. Le champ Unique FlowFiles doit être positionné à true pour que les fichiers générés soient différents. Puis cliquer sur Apply pour enregistrer votre configuration.

Chaque processor dispose d'un ensemble de relations définies, appelées Relationships dans Nifi, pour le transfert de données. Lorsque le traitement d'un FlowFile est terminé, le processor le dirige vers l'une de ces relations afin d'adapter le traitement des FlowFiles en fonction du résultat obtenu.

Une relation représente un chemin logique pour les FlowFiles au sein d'un processor et indique l'état du résultat de traitement de chaque FlowFile, ce qui permet de déterminer leur traitement ultérieur. Les relations les plus courantes sont "Success" et "Failure":

- La relation Success est destinée aux FlowFiles traités sans erreurs, leur permettant de continuer vers les composants NiFi suivants pour un traitement ou un stockage ultérieur. Ce chemin est crucial pour maintenir l'intégrité des données et s'assurer que les données valides poursuivent leur circulation dans le système.
- La relation Failure concerne les FlowFiles ayant rencontré des erreurs durant le traitement. Elle permet de gérer ces erreurs de manière appropriée, notamment en

consignant le problème, en notifiant les utilisateurs ou en redirigeant les FlowFiles vers un composant dédié au dépannage ou à la remédiation. Une gestion efficace des échecs est primordiale pour assurer la robustesse du système et la qualité des données.



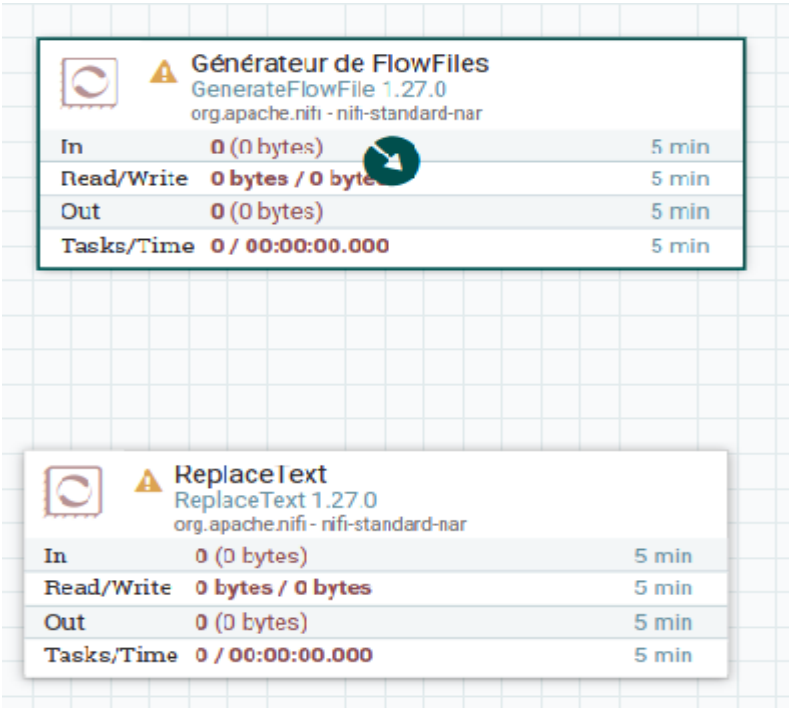
Notre processor GenerateFlowFile possède deux champs qui gèrent la relation de succès dans l'onglet "Relations", à savoir "terminate" et "retry". Si l'objectif est simplement d'arrêter le traitement après la génération des fichiers de flux, vous pouvez suivre la relation de succès de GenerateFlowFile directement via une option de terminaison. Cela signifie que tous les fichiers de flux générés avec succès seront immédiatement supprimés. Cette option n’est pas disponible si un composant en aval est connecté au processor actuel. Tandis que "retry" indique qu’en cas de succès, le fichier de flux est retraité. Les tentatives de réexécution sont généralement utilisées lorsqu'un fichier de flux rencontre une erreur (par exemple, un problème de réseau lors d'une tentative d’un fichier à distance). Dans ces cas, les réessais ont du sens car le processor peut échouer à accomplir sa tâche et tentera de retraiter le fichier de flux. Cependant, dans le cas de GenerateFlowFile, puisqu'il n'y a pas de scénario d'échec prévu (il génère simplement des fichiers de flux), le concept de réessai n’est pas pertinent.

4) Glissez un processor ReplaceText sur le canevas. Configurez-le pour remplacer le texte dans les fichiers de flux générés par GenerateFlowFile dans la page propriétés :
    - Search value : une expression régulière qui se focalise une lettre majuscule ou minuscule.
    - Replacement Value : "Apache Nifi"

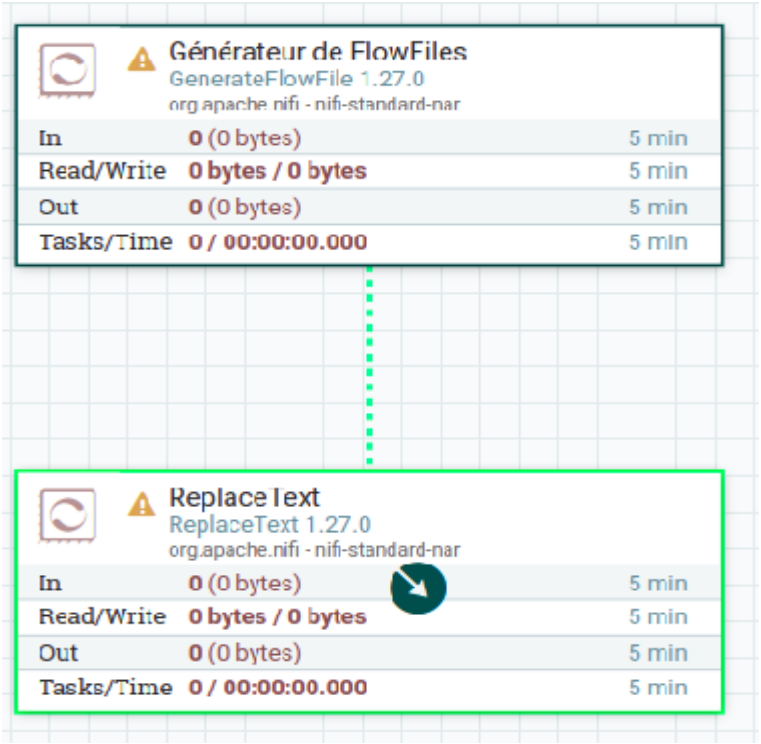
Cela remplacera une lettre quelconque par "Apache Nifi" dans le contenu du fichier de flux.

Après avoir ajouté et configuré les Processors et autres composants dans l'espace de travail, l'étape suivante consiste à les relier entre eux afin que NiFi puisse déterminer le composant qui sera appliqué à chaque FlowFile après leur traitement par un autre composant. Cette opération s'effectue en créant une connexion entre chaque composant. Une connexion est un objet qui permet d'acheminer les données d'un composant à un autre, tout en offrant la possibilité de contrôler le débit et l'ordonnancement des données.

Pour instancier une connexion, positionnez le curseur au centre d'un composant de départ, une icône de connexion () apparaît :

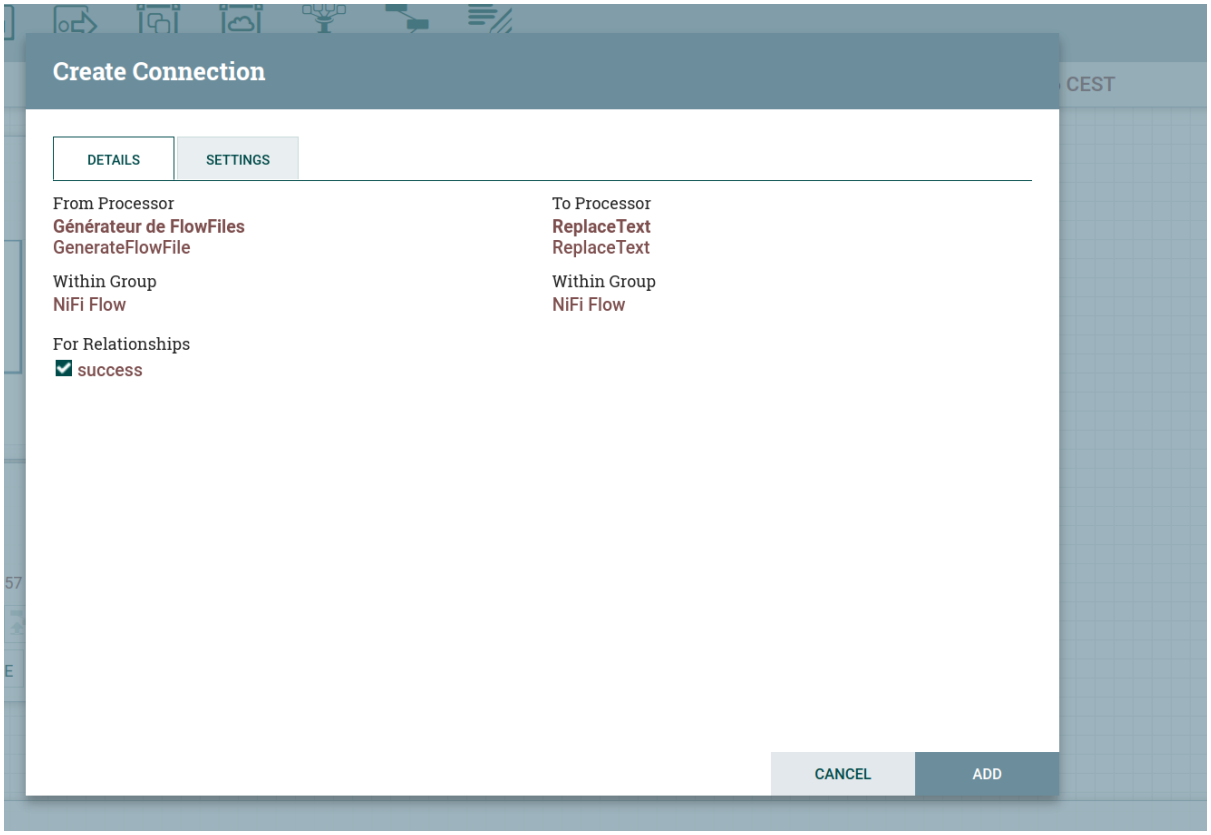


Glissez ensuite cette icône du composant vers un autre composant jusqu'à ce que les contours de ce dernier changent de couleur, généralement en vert. Il est également possible de créer une connexion en boucle sur le même processor. Cette option est utile si l'on souhaite que le composant tente de retravailler les FlowFiles en cas d'échec. Pour créer ce type de connexion, il suffit de faire glisser l'icône de connexion et de la ramener vers le même composant jusqu'à ce que ses contours changent de couleur.



En relâchant la souris, la même fenêtre de dialogue « Create Connection » s'ouvre. Celle-ci comporte deux onglets : « Details » et « Settings », décrits plus en détail ci-dessous.

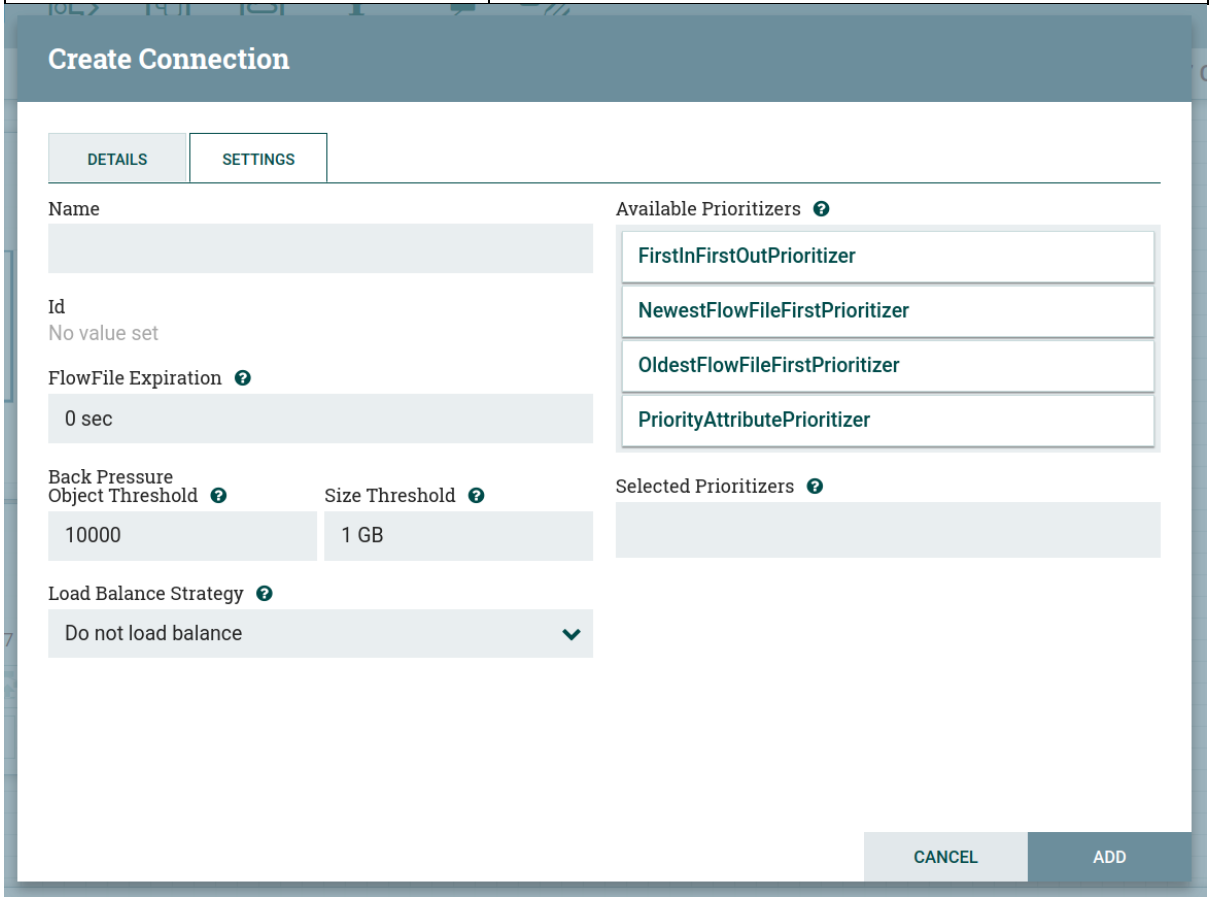
L'interface de configuration propose deux onglets : « Details » et « Settings ». L'onglet « Details » fournit des informations sur l'origine et la destination de la connexion, ainsi que les relations associées. L'utilisateur peut sélectionner les relations souhaitées. Dans notre cas (voir l'image ci-dessus), seule la relation « success » est proposée, car le processor GenerateFlowFile dirige ses résultats uniquement vers cette relation.



Dans l'onglet « Settings » (voir l'image ci-dessous), plusieurs champs sont proposés pour gérer les flux de données transitant par la connexion, notamment FlowExpiration, BackPressure Object Threshold et Size Threshold, les autres champs seront abordés plus tard dans ce module :

Champ	Description
FlowExpiration	Paramètre de configuration qui contrôle la durée pendant laquelle un fichier de flux peut rester dans une file d'attente d'une connexion avant d'être automatiquement supprimé. Cela est utile dans les situations où les données sont très volumineuses ou sensibles au temps (pour éviter de traiter des données obsolètes ou non pertinentes). Si cette durée est fixée à "10 min", cela signifie que les fichiers de flux dans la file d'attente de la connexion depuis plus de 10 minutes seront automatiquement supprimés.
BackPressure Object threshold	Contrôle le nombre maximal de fichiers de flux pouvant être mis en file d'attente dans une connexion entre deux processors avant que NiFi n'applique une contre-pression. Lorsque cette limite est atteinte, les processors en amont arrêteront de générer de nouveaux fichiers de flux jusqu'à ce que de l'espace soit disponible dans la file d'attente.
BackPressure Size threshod	Similaire à BackPressure Object threshold, mais au lieu de limiter le nombre de fichiers de flux, il contrôle

	la taille totale des données (en octets) pouvant être mises en file d'attente dans une connexion.



5) Pour démarrer les processors, choisissez l'une des options suivantes :
   - Effectuez un clic droit sur le processor et sélectionnez "Démarrer".
   - Sélectionnez le processor et cliquez sur "Démarrer" dans la barre d'outils des opérations.

Ces deux options ne s'appliquent qu'à un processor à la fois. Pour sélectionner plusieurs processors, maintenez la touche Shift enfoncée tout en cliquant avec le bouton gauche de la souris et en faisant glisser pour entourer tous les processors concernés. Une fois que vous les avez sélectionnés, relâchez la touche et cliquez sur "Démarrer" dans la barre d'outils des opérations.

3. Cas pratiques

Exercice 1. Lecture d'un fichier depuis une source et écriture vers une destination

L’objectif est de créer un flux NiFi qui lit des fichiers à partir d'un répertoire et les écrit dans un autre répertoire.

Étape 1 : Ajout des processors

1. **GetFile** : Ce processor lit les fichiers depuis un répertoire source.
    a. **Propriétés** :
        i. Input Directory : Spécifiez le chemin du répertoire source (ex : /home/me/nifi/data).
        ii. File Filter: Filtrer les fichiers pdf (expression régulière)
        iii. Keep Source File : Réglez sur true pour conserver le fichier après traitement.
    b. Sheduling
        i. Fixez Run Schedule à 30 secondes pour espacer les lectures de 30 secondes.
2. **PutFile** : Écrit les fichiers dans un répertoire de destination.
    a. **Propriétés** :
        i. Directory : Spécifiez le chemin du répertoire de destination (ex : /home/me/nifi/data_output).

Étape 2 : Configuration des connexions entre processors

- **Connexion** : Dans NiFi, les processors sont connectés entre eux via des connexions. Une connexion est une file d'attente qui permet de passer les données d'un processor à l'autre. Voici comment configurer les connexions entre les processors :
    - **GetFile → PutFile** : Une fois que vous avez ajouté les deux processors (GetFile et PutFile) sur votre canvas, vous devez les connecter :
    - **Relations** :
        - Pour le processor **GetFile**, la relation unique est success , ce qui signifie que lorsque le fichier est lu avec succès, il est envoyé à la prochaine étape (dans ce cas, **PutFile**).
        - Pour **PutFile**, vous pouvez configurer la relation success ou failure (échec), en fonction de la gestion des erreurs. Dans notre cas, si l'écriture échoue, adapter la configuration pour réessayer 5 fois avant de aut-terminer la relation failure. Si succès, optez pour une terminaison automatique de la relation success.

Étape 3 : Exécuter le flux

1. Activez les processors en les sélectionnant et en cliquant sur **Start**.
2. Le fichier devrait être lu depuis le répertoire source et écrit dans le répertoire de destination.

Vérifiez la file de connexion entre **GetFile** et **PutFile** pour voir si des fichiers sont en attente ou ont été traités.

3. Pendant que les processors sont en marche, ajoutez d'autres fichiers dans votre répertoire source puis vérifier s'ils sont copiés dans le répertoire destinataire.
4. Visualisez la liste des flowFiles qui transitent par la connexion. Cliquez droit sur la connexion puis List queue, un tableau de flowFiles s'ouvre. Une ligne référence un flowFile et ses informations (identifiant, nom, taille...). En cliquant sur l'icône œil dans la dernière colonne, vous obtenez le contenu du flowFile.
5. Listez les flowFiles ayant transité par le processor PutFile. Pour cela, vous effectuez un clique-droit sur le processor puis View data provenance. Vous obtenez un tableau de flowFiles. Une ligne référence un flowFile et ses informations (la date de traitement par le processor, le type d'opération appliqué sur le flowFile...). Dans la dernière colonne, vous pouvez visualisez, sous forme de graphe simple, l'ensemble de traitements subis par le flowFile depuis le début du flux de données jusqu'au processor actuel.
6. Dans le processor GetFile, fixez Run Schedule à 0 secondes. Que se passe-t-il au niveau de la connexion.
7. Arrêter les processors puis videz la file d'attente de la connexion (Clique-droit puis Empty queue).
8. Modifiez le schéma du flux de données pour effectuer une tâche de déplacement des fichiers au lieu d'une copie.
9. Essayez de supprimez la connexion entre les deux processors. Que se passe-t-il ?
10. Arrêtez tous les processors.
11. Dans NiFi, il n'existe qu'un seul espace de travail pour tous les flux de données. Avant de poursuivre avec l'exercice suivant, il est nécessaire de regrouper l'ensemble du flux de données dans un Process Group. Pour ce faire, sélectionnez tout le flux, faites un clic droit, puis choisissez l'option Group. Une fenêtre s'ouvrira pour vous permettre de nommer le groupe.

Exercice 2. Découpage de fichiers

Considérez un flux de données qui récupère des fichiers à partir d'un répertoire, les découpe ligne par ligne, puis enregistre chaque ligne dans un fichier distinct dans un dossier de destination :

GetFile → success → **SplitText** → splits →**UpdateAttribute** → success → **PutFile**

Créez un nouveau Process Group et donnez-lui un nom. Double-cliquez ensuite sur celui-ci. Vous êtes maintenant à l'intérieur de l'espace de ce Process Group, où vous allez configurer votre flux de données pour l'exercice 2.

Proposez une implémentation de ce flux dans l'espace de travail de NiFi, en configurant correctement les propriétés des processors et des connexions pour accomplir la tâche décrite. Dans l'onglet properties du processor UpdateAttribute, créez un nouvel attribut (en cliquant sur le bouton + en haut à droite) appelé filename avec la valeur ${filename}-${UUID()}.txt pour assigner à chaque ligne un fichier avec nom unique.

Dans le dossier “surveillé” par GetFile, mettez un fichier ou fichiers texte quelconques contenant au moins deux lignes puis vérifier le contenu du dossier contenant le résultat.

Pour information, les flowFiles possèdent des attributs. Lorsqu'un flowFile passe par un processeur, l'utilisateur peut ajouter de nouveaux attributs (également appelés propriétés) ou modifier ceux existants. Ces attributs sont dits dynamiques, car ils ne sont pas prédéfinis par le système Nifi. Ils sont créés à l'exécution en fonction des besoins spécifiques. Dans notre exemple, l'attribut dynamique filename prend la valeur ${filename}-${UUID()}.txt :

- ${filename} : Cette expression fait référence à l'attribut filename déja existant, qui contient le nom du fichier en cours de traitement.
- UUID() : Fonction de NiFi qui génère un identifiant unique (UUID). Chaque exécution produit un nouvel identifiant.
- Le tiret - est utilisé pour concaténer le nom du fichier et l'UUID.
- .txt : L'extension .txt est ajoutée pour indiquer que le fichier résultant sera au format texte.

L'expression ${filename}-${UUID()}.txt est écrite en NiFi Expression Language, un langage spécifique à Apache NiFi permettant de manipuler les attributs, d'utiliser des fonctions et de créer des expressions dynamiques pour modifier les flux de données.

Quelques caractéristiques du NiFi Expression Language :

- **Accès aux attributs** : Les attributs, tels que ${filename}, sont accessibles via la syntaxe ${} pour en extraire la valeur.
- **Fonctions intégrées** : NiFi propose un ensemble de fonctions telles que UUID(), substring(), replace(), et bien d'autres, permettant de manipuler les valeurs des attributs.

Pour plus d'informations sur les fonctions prédéfinies et la syntaxe, consultez la documentation : Guide du NiFi Expression Language.

Exercice 3. Découpage de fichiers distants

Réalisez la même tâche que dans l'Exercice 2, mais cette fois-ci avec des fichiers CSV disponibles à l'adresse suivante : https://github.com/scikit-learn/scikit-learn/tree/main/sklearn/datasets/data. Ce lien présente la liste des fichiers CSV. Pour télécharger un jeu de données spécifique (par exemple nommé nom_du_jeu_de_donnees.csv), utilisez le lien suivant : https://raw.githubusercontent.com/scikit-learn/scikit-learn/refs/heads/main/sklearn/datasets/data/nom_du_jeu_de_donnees.csv.

Servez-vous du processor **InvokeHTTP** pour télécharger des fichiers distants via le protocole HTTP. Vous pouvez également utiliser le processeur **ExtractText** pour extraire les liens des fichiers CSV en appliquant, par exemple, des expressions régulières.

À noter qu'un type de processeur peut être réutilisé plusieurs fois dans le flux de données.

Si une connexion a une file d'attente dont le seuil de BackPressure Object threshold ou BackPressure size threshold est atteint, vous pouvez augmenter la valeur sur l'onglet settings en double-cliquant sur la connexion.

Exercice Défi. Découpage de fichiers zippés distants

Refaites l'Exercice 3, mais cette fois-ci avec des fichiers zippés disponibles sur https://web.ais.dk/aisdata/. Il est demandé de décompresser chaque archive zip afin d'en extraire le fichier CSV. Vous pouvez utiliser le processor **UnpackContent** pour cela. Cet exercice est à déposer sur Moodle. Pour ce faire, sélectionnez le Process Group de l'exercice, faites un clic droit et choisissez "create template". Nommez ensuite le template avec votre prénom_nom_TP1_defi. Allez ensuite dans le menu principal de l'interface web de NiFi, puis sélectionnez "templates". Enfin, téléchargez le template nouvellement créé (fichier XML) et déposez-le sur Moodle.