**Sommaire**

[**Introduction Générale** 5](#_Toc163569041)

[**Chapitre 1 : Présentation du cadre du projet** 6](#_Toc163569042)

[**Introduction** 7](#_Toc163569043)

[**I.** **Présentation de l’organisme d’accueil** 7](#_Toc163569044)

[**II.** **Présentation du projet** 8](#_Toc163569045)

[**1.** **Produits présentés par NeoLedge** 8](#_Toc163569046)

[**2.** **Contexte du projet** 8](#_Toc163569047)

[**3.** **Étude de l’existant** 9](#_Toc163569048)

[**4.** **Analyse de l’existant** 9](#_Toc163569049)

[**5.** **Solution proposée :** 10](#_Toc163569050)

[**III.** **Méthodologies et formalisme adopté** 10](#_Toc163569051)

[**1.** **Méthodologie de modélisation et de conception** 10](#_Toc163569052)

[**2.** **Méthodologie de travail** 11](#_Toc163569053)

[**IV.** **Environnement de développement** 13](#_Toc163569054)

[**1.** **Environnement matériel** 13](#_Toc163569055)

[**2.** **Environnement logiciel** 14](#_Toc163569056)

[***Conclusion*** 15](#_Toc163569057)

[**Chapitre 2 : Planification du Backlog Produit** 16](#_Toc163569058)

[**Introduction** 17](#_Toc163569059)

[**I.** **Identifications des acteurs :** 17](#_Toc163569060)

[**II.** **Les besoins non fonctionnels** 18](#_Toc163569061)

[**III.** **Les User stories** 18](#_Toc163569062)

[**IV.** **Product Backlog :** 21](#_Toc163569063)

[**Chapitre 3 : Release 1** 25](#_Toc163569064)

[**Introduction** 26](#_Toc163569065)

[**I.** **Organisation des sprints :** 26](#_Toc163569066)

[**II.** **Sprint 1 « Etude de l’environnement »** 26](#_Toc163569067)

[**1.** **Sprint Goal** 26](#_Toc163569068)

[**2.** **Sprint Backlog :** 27](#_Toc163569069)

[**3.** **Implémentation du sprint :** 28](#_Toc163569070)

[**4.** **Sprint Review :** 41](#_Toc163569071)

[**5.** **Sprint Rétrospective :** 41](#_Toc163569072)

[**III.** **Sprint 2 Personnaliser le panneau des propriétés :** 41](#_Toc163569073)

[**Conclusion :** 42](#_Toc163569074)

[Introduction 44](#_Toc163569075)

**Liste des figures**

[Figure 1: Logo de NeoLedge 7](#_Toc163564369)

[Figure 2: Logo de Elise 9](#_Toc163564370)

[Figure 4: Méthode SCRUM 13](#_Toc163564371)

[Figure 5 : Organisation des releases 24](#_Toc163564372)

[Figure 6 : Release 1 26](#_Toc163564373)

[Figure 7 : Installer le .NET Framework 28](#_Toc163564374)

[Figure 8 : Installation Visual Studio 2022 28](#_Toc163564375)

[Figure 9 : Microsoft SQL Server Management Studio 19 29](#_Toc163564376)

[La figure 10 et 11 ci-dessous décrit bien les créations de compte utilisateurs dan 30](#_Toc163564377)

[Figure 10 créer un compte 30](#_Toc163564378)

[Figure 11 Vérification Dans Base de données 30](#_Toc163564379)

[Figure 12 Login User 31](#_Toc163564380)

[Figure 13 Non autorisé à ajouter une tâche 31](#_Toc163564381)

[Figure 14 Ajouter une tache 32](#_Toc163564382)

[Figure 15 Afficher les tache 32](#_Toc163564383)

[Figure 16 Modifier une tache 33](#_Toc163564384)

[Figure 17 Supprimer une tache 33](#_Toc163564385)

[Figure 18 Dashboard ELSA workflow 34](#_Toc163564386)

[Figure 19 Configure HTTP ENDPOINT 34](#_Toc163564387)

[Figure 20 configure HTTP RESPONSE 35](#_Toc163564388)

[Figure 21 Test Workflow Designer 35](#_Toc163564389)

[Figure 22 Configure Elsa Workflow 36](#_Toc163564390)

[Figure 23 créer le premier workflow 36](#_Toc163564391)

[Figure 24 Test Workflow Programmatique 36](#_Toc163564392)

[Figure 25 Exemple de Workflow pour calculer le somme 37](#_Toc163564393)

[Figure 26 Intégration BPMN.JS dans VueJs3 37](#_Toc163564394)

[Figure 27 Exemple de gestion d’un digramme en ligne (bpmn.io) 38](#_Toc163564395)

[Figure 28 Structure d’un diagramme 38](#_Toc163564396)

[Figure 29 Premier Partie de workflow Extraire contenu de diagramme 39](#_Toc163564397)

[Figure 30 Activité pour et obtenir l’ID de l’utilisateur 39](#_Toc163564398)

[Figure 31 Deuxième Partie de workflow Obtenir l’utilisateur 40](#_Toc163564399)

[Figure 32 Test Exécution 40](#_Toc163564400)

[Figure 33 Plan d'amélioration du sprint 1 41](#_Toc163564401)

**Liste des Tableaux**

[Tableau 1 : Product Backlog 23](#_Toc163564364)

[Tableau 2 : Sprint Backlog du sprint 1 27](#_Toc163564365)

# **Introduction Générale**

# **Chapitre 1 : Présentation du cadre du projet**

## **Introduction**

Le contexte général du projet est présenté dans ce chapitre introductif. Nous présenterons

d'abord notre organisme d'accueil. Ensuite, nous expliquerons les problèmes à résoudre et la solution proposée pour donner un aperçu de notre projet. Enfin, nous fournirons une description de la méthode que nous utilisons pour mener à bien le projet.

## **Présentation de l’organisme d’accueil**

Acteur indépendant et à dimension internationale, NeoLedge est une société française en forte croissance, qui s’appuie sur un réseau de partenaires privilégiés en Europe, en Amérique du Nord et en Afrique.

Éditeur spécialisé dans la gestion électronique de documents, NeoLedge compte à son actif des centaines de clients, des dizaines de milliers d’utilisateurs quotidiens et des millions de documents gérés par ses solutions, dans le secteur public comme dans le secteur privé.

Partenaire certifié Gold de Microsoft en ce qui concerne le développement d’applications et les plateformes cloud, NeoLedge accompagne des organisations pendant leur transition numérique. NeoLedge s’appuie depuis ses débuts sur les techniques développées par Microsoft.

La figure 1 représente le logo de NeoLedge.

Figure 1: Logo de NeoLedge

La structure hiérarchique et les différents départements qui composent NeoLedge peuvent être visualisés à l'aide de l'organigramme suivant :

## **Présentation du projet**

### **Produits présentés par NeoLedge**

NeoLedge propose une gamme complète de solutions de gestion électronique de documents (GED) et de gestion de contenus (ECM) pour les entreprises et les administrations.

• ECM Elise : Pour les entreprises de toutes tailles, qui sont confrontées à une gestion complexe des flux d’informations nécessitant de nombreuses interactions avec des tiers, la solution ECM Elise, éditée par NeoLedge, propose des fonctions automatisées pour la capture des flux multicanaux, la gestion des dossiers et l’orchestration des processus métiers.

• DocFactory : Pour les entreprises à la recherche d’une solution performante capable de les aider dans leur transformation numérique et leur passage au zéro papier, DocFactory est une solution numérique tout-En-un offrant des fonctions centralisées de numérisation, conversion, et capture de documents électroniques, ainsi que d’export et de signature électronique.

Contrairement aux solutions propriétaires ne gérant pas le processus de dématérialisation dans sa globalité, DocFactory permet d’industrialiser l’ensemble des opérations de capture des flux de documents jusqu’à leur intégration dans votre système

d’information.

• Illico : Pour les élus et dirigeants de villes de toutes tailles, engagées dans une politique de transition numérique grâce à des outils innovants qui simplifient leur fonctionnement, illico offre aux citoyens et aux équipes de la ville des services de confiance pour la dématérialisation, le traitement des requêtes, la gestion du courrière l’automatisation des processus métiers. Contrairement aux solutions en silos ne gérant pas la continuité des processus entre les citoyens et les agents, “illico powered by Elise” est un service illimité, rapide à mettre en œuvre, conçu pour les villes et les réseaux de villes.

### **Contexte du projet**

Le présent projet intitulé « Workflow BPMN » est réalisé dans le cadre de projet de fin

d’études pour l’obtention du diplôme de licence national en technologie d’information au

sein de l’Institut Supérieur des Études Technologiques de Bizerte.

Ce projet a été réalisé dans la société NeoLedge durant la période s’étalant du 05

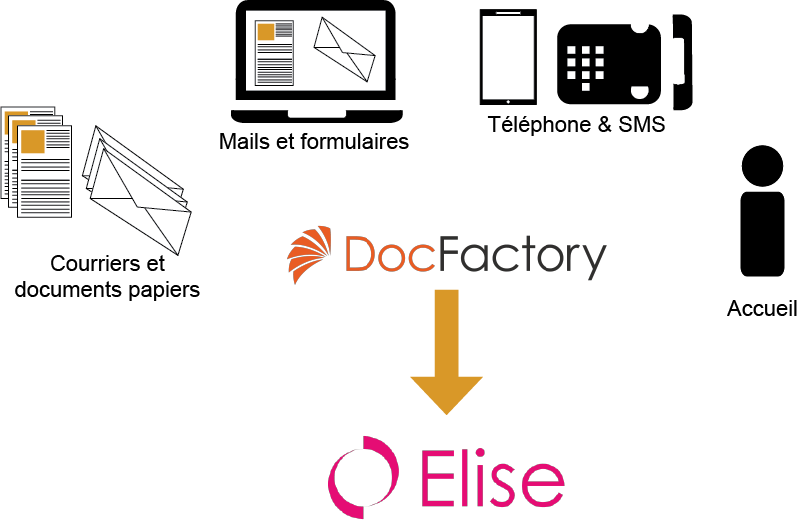
février 2023 jusqu’à 25 mai 2024.

Figure 2: Logo de Elise

### **Étude de l’existant**

Le développement logiciel fait face à des défis, notamment traduire les besoins fonctionnels en code.

Cela demande une compréhension des attentes du client et une expertise technique. Les développeurs doivent analyser des documents détaillés, un processus long et complexe. Ils doivent aussi assurer une grande précision pour que le code corresponde aux exigences, ce qui peut allonger le cycle de développement et augmenter le stress.

Les malentendus dans la conversion des besoins peuvent causer des retards et nécessiter des ajustements coûteux, obligeant à plus de clarifications et à une communication accrue avec les parties prenantes, affectant la productivité.

### **Analyse de l’existant**

Les problèmes constatés au niveau de l'existant comprennent un processus de développement basé sur des spécifications détaillées :

* Développement basé sur des spécifications.
* Manque de visualisation des processus métier.
* Processus d'exécution manuelle.
* Suivi de la performance limitée.

### **Solution proposée :**

Notre objectif consiste à concevoir un moteur de workflow BPMN qui sera intégré à la plateforme de gestion, **Elise**.

L'objectif de cette solution est d'offrir une gamme complète d'outils afin d'optimiser la gestion des processus métier.

Notre solution permettra d'abord une modélisation graphique facile à comprendre des processus, ce qui facilitera leur compréhension et leur communication au sein de l'organisation.

Nous utiliserons la bibliothèque **BPMN.IO**, qui est connue pour sa flexibilité et sa compatibilité avec **Elise**, pour y parvenir.

Ensuite, notre solution comprendra des fonctionnalités avancées pour l'exécution des processus, telles que la gestion des tâches, les flux de travail et les règles de l'entreprise. Pour cette étape, nous utiliserons la bibliothèque puissante Elsa Workflow.

Enfin, notre solution fournira des tableaux de bord personnalisables et des rapports détaillés pour suivre les indicateurs de performance importants des processus.

Les utilisateurs pourront identifier les domaines d'amélioration et d'optimisation de leurs opérations grâce à ces analyses.

Dans l'ensemble, notre moteur de workflow BPMN pour Elise vise à révolutionner la gestion des processus métier en offrant une solution complète, intuitive et puissante, qui simplifie les opérations quotidiennes, améliore l'efficacité opérationnelle et stimule la croissance de l'entreprise.

## **Méthodologies et formalisme adopté**

Au cours de cette section, nous détaillons les techniques utilisées pour avancer dans la réalisation de notre projet.

### **Méthodologie de modélisation et de conception**

Une méthode de modélisation et de conception est un processus qui permet de formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client lors de la réalisation d'un projet informatique. Pour cette raison, nous avons opté pour UML pour la conception de notre projet.

**UML** (Unified Modeling Language) est un langage universel de modélisation orientée objet,

établi comme un standard par l'OMG (Object Management Group) et dans le domaine public, qui offre une notation visuelle pour communiquer et concevoir des architectures logicielles.



### **Méthodologie de travail**

Le choix entre les méthodes classiques et agiles de gestion de projet dépend de la complexité du projet, des exigences du client et du degré de flexibilité requis. Les méthodes classiques conviennent mieux aux projets bien définis avec des calendriers clairs et des risques identifiés. En revanche, les méthodes agiles sont préférables pour les projets plus vastes où les spécifications ne sont pas claires ou les besoins du client évoluent. Agile permet une adaptation continue en fonction des retours du client, réduisant ainsi le risque de recommencer à zéro. En fin de compte, le choix de la méthode doit correspondre aux exigences uniques de chaque projet pour maximiser ses chances de succès.

#### **Choix méthodologique**

Nous nous sommes inspirés du framework Scrum, donc notre choix méthodologique correspond à notre projet. Scrum est largement reconnu comme le framework de gestion de projet le plus couramment utilisé, le mieux documenté et soutenu. Il offre un cadre de travail structuré pour les équipes pluridisciplinaires, avec une attention particulière portée sur l'adaptabilité et la collaboration, permettant de répondre efficacement aux problèmes complexes et évolutifs.

#### **Pourquoi Scrum**

Dans le cadre de notre projet et afin d’assurer le bon déroulement des différentes phases de ce dernier, nous avons opté pour le framework agile Scrum pour la conception et le développement de notre système pour des raisons bien déterminées. En effet, le framework Scrum s’adapte parfaitement à la décomposition de notre projet de fin d’études.

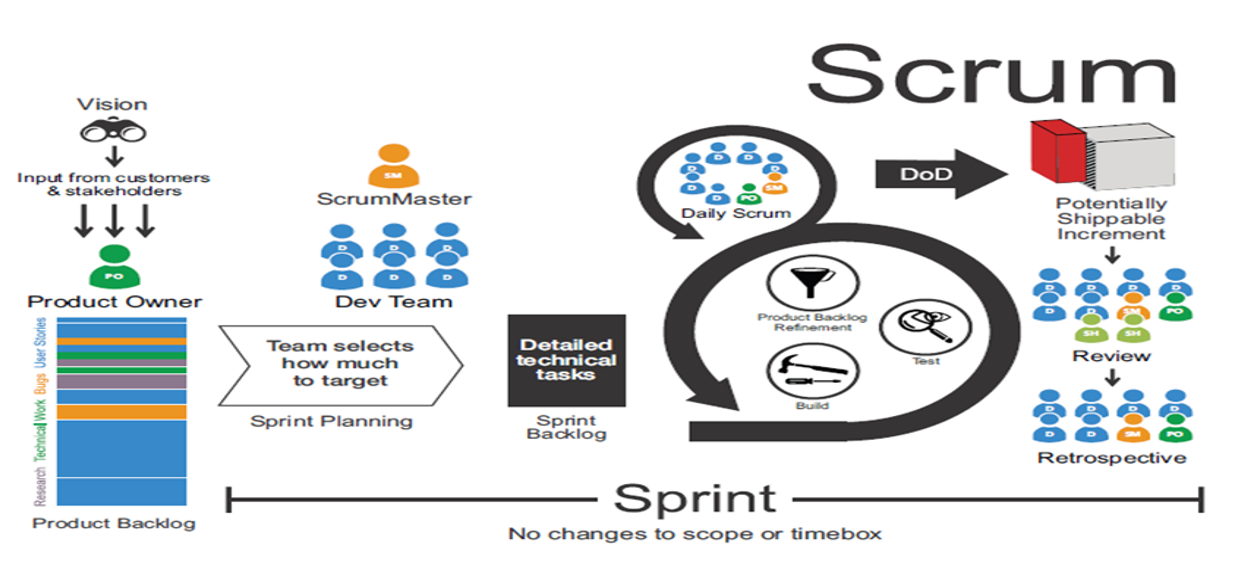
Il se base sur les avantages suivants :

* Plus de souplesse et de réactivité.
* Grande capacité d’adaptation au changement grâce à des itérations courtes.
* Satisfaire au mieux les besoins du client.

**SCRUM** est une méthodologie agile qui consiste à avoir une équipe soudée orientant le projet

au fil de son avancement afin d’atteindre un but. Cette approche est à la fois dynamique et productive, engendre la réalisation des fonctionnalités par itération en incluant la participation du client. Chaque itération peut durer de deux à quatre semaines, à la fin de chaque sprint un produit fonctionnel doit être livré. En effet, Scrum définit trois rôles qui sont :

* **Product Owner** (Le gestionnaire de produit) : Le responsable du produit de l’équipe projet client et il représente les utilisateurs finaux. C’est lui qui va définir et prioriser la liste des fonctionnalités du produit et choisir la date et le contenu de chaque sprint sur la base des valeurs (charges) qui lui sont communiquées par l’équipe.
* **Scrum Master** (Le maître de Scrum) : Véritable facilitateur sur le projet, il veille à ce que chacun puisse travailler au maximum de ses capacités en éliminant les obstacles et en protégeant l’équipe des perturbations extérieures.
* **L’équipe de développement** (L’équipe de projet) : L’équipe de réalisation contient au minimum deux développeurs. Elle regroupe tous les rôles habituellement nécessaires à un projet, à savoir l’architecte, le concepteur, le développeur, le testeur, etc.

Figure 3: Méthode SCRUM

## **Environnement de développement**

Avant de commencer à mettre en œuvre notre projet, nous décrivons l'environnement et les outils de travail que nous utiliserons. Nous définirons d'abord l'environnement matériel, puis l'environnement logiciel. Enfin, nous présenterons les divers frameworks et langages que nous utiliserons pour ce projet.

### **Environnement matériel**

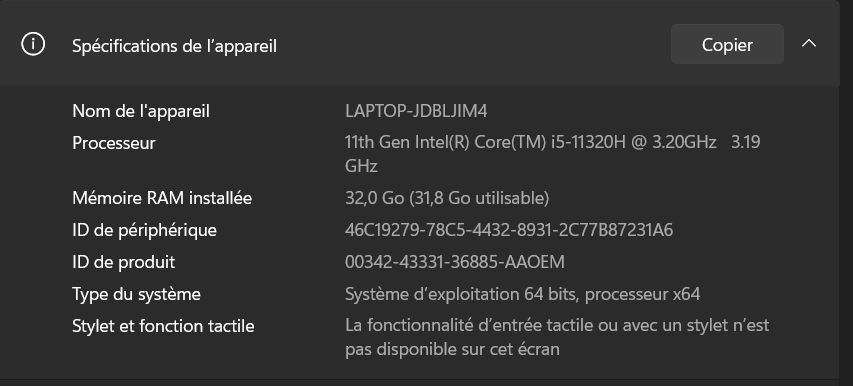
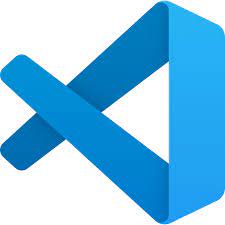


Figure 4 : Caractéristiques du PC de Talel.

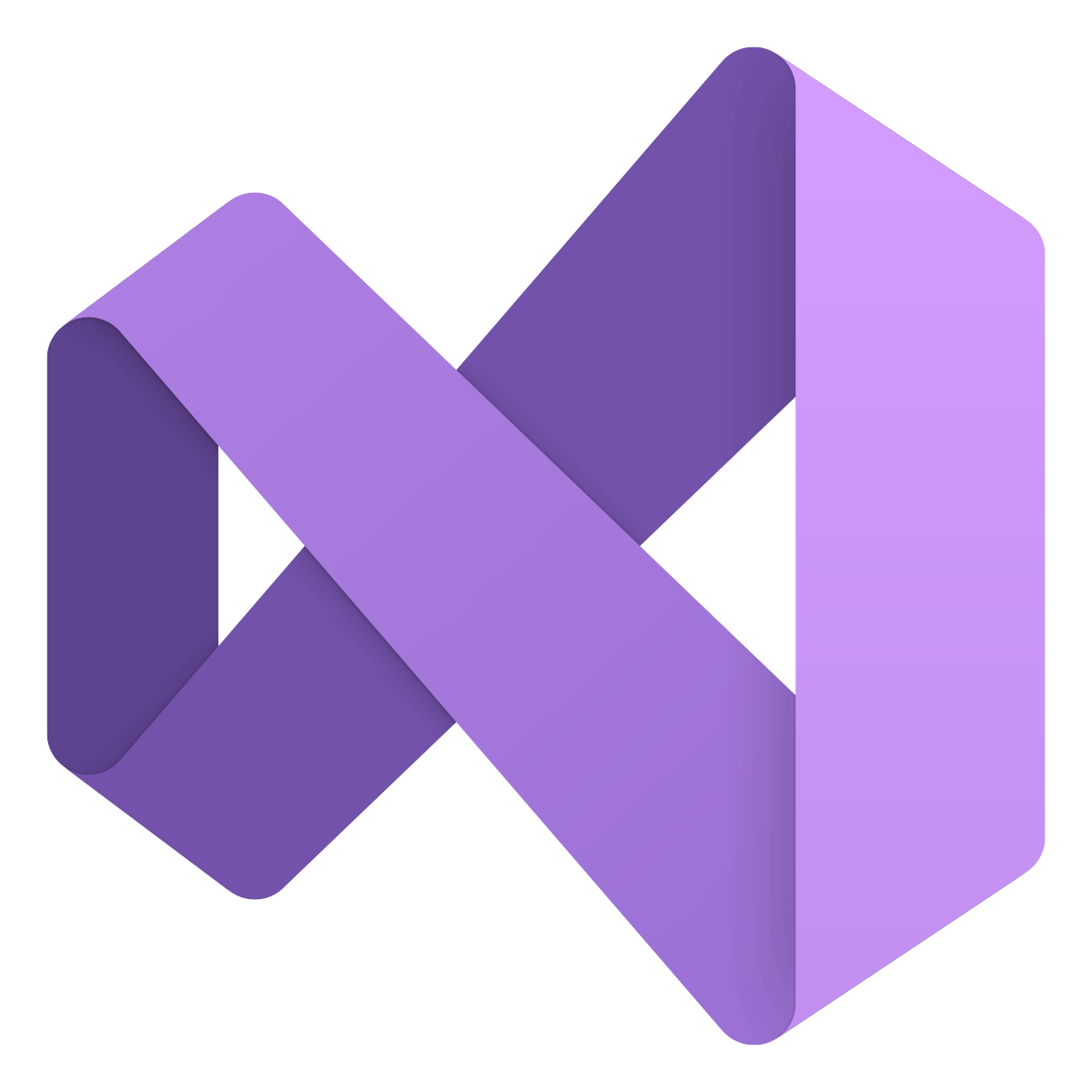
### **Environnement logiciel**

Au cours de cette section, nous dresserons une liste des outils utilisés pour étudier et mettre en place notre application tout au long du projet.

*  **Visual Studio Code logo:**

Un éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOs. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégrer. []

* **Visual Studio logo :**



Visual Studio est un environnement de développement intégré (IDE) extensible, complet et gratuit pour créer des applications modernes pour Android, iOS, Windows, ainsi que des applications web et des services cloud.

## ***Conclusion***

Au cours de ce chapitre, qui est une étape essentielle pour établir les principes de base de notre projet. Nous avons présenté l'organisme d'accueil du projet ainsi que les attentes. En effet, afin de mieux comprendre les fonctionnalités de notre solution et de révéler les limites de la solution existante, nous avons mené une étude de l'existant. De plus, nous avons déterminé le cadre du travail et la méthodologie à utiliser pour ce projet. Le chapitre suivant explique la conception de l'application en détaillant ses spécifications, ses acteurs et le backlog de produits contenant ses fonctionnalités.

# **Chapitre 2 : Planification du Backlog Produit**

## **Introduction**

La planification joue un rôle essentiel dans la réalisation d’un projet et sert de fondement à

chaque application. Ce chapitre se concentre sur la planification du backlog produit, qui vise à clarifier les besoins à satisfaire de manière précise.

## **Identifications des acteurs :**

Dans notre projet, les utilisateurs sont des entités externes au système de modélisation qui

interagissent directement avec des personnes ou des appareils. Chaque utilisateur a un ensemble d'actions qui correspondent aux fonctions dont il a besoin. Notre projet a deux acteurs différents, ayant chacun un rôle correspondant.

L'acteur **Utilisateur** est un employé de l'entreprise. Il possède les caractéristiques suivantes :

* En utilisant une gamme de symboles standardisés pour représenter le flux de travail, ajouter des activités spécifiques à un processus métier.
* établir des règles métiers pour automatiser certaines décisions et optimiser l'efficacité opérationnelle.
* Relier les activités du processus aux fonctionnalités existantes de la solution Elise.
* Suivre l'état d'avancement des processus en cours d'exécution pour surveiller les performances et de distinguer chaque activité réussie or échouée.
* Générer des rapports personnalisés sur les performances des processus en analysant des indicateurs clés tels que les temps de traitement, les taux de réussite et les écarts par rapport aux objectifs fixés.
* Changer la couleur de chaque activité pour une meilleure visualisation et organisation du processus métier.
* Importer un processus dans l'application ou le sauvegarder une fois créé (local Storage) ou modifié.
* Zoomer (zoom in) ou Dézoomer (zoom out) sur le diagramme de processus.
* Sauvegarder les processus dans la base de données.
* Ajouter des commentaires pour chaque activité afin de faciliter la collaboration et la traçabilité des actions.

L’acteur **Admin** : Il a comme fonctionnalités :

* Gestion des autorisations des utilisateurs pour accéder, modifier ou exécuter divers processus afin de protéger les données.

## **Les besoins non fonctionnels**

Notre application doit contenir certaines contraintes indispensables pour son bon

fonctionnement.

* **Maintenabilité :** le code de l’application développé doit être lisible et compréhensible afin d’assurer son état extensible et évolutif par les autres développeurs de l’ERP.
* **La convivialité :** les interfaces de l’application doivent être simples faciles et homogènes avec les interfaces de l’ERP de la société.
* **Fiabilité :** l’application doit être fonctionnelle en minimisant les erreurs d'extraction de données, ce qui peut entraîner des retards et des pertes de temps.
* **La performance :** notre application doit être capable de lire rapidement les informations des factures scannées et d'extraire avec précision les données.

## **Les User stories**

Les user stories représentent **les descriptions d’exigences** pour toutes fonctionnalités ou

“tâches” nécessaires au fonctionnement du produit ou du service en développement. Ils ont

été écrits par le **Product Owner** (Mr Walid AZOUZI) pour nous faciliter l'exécution et la compréhension des tâches.

* **Qui** : la fonctionnalité est faite pour qui
* **Quoi** : qu’est-ce que l’utilisateur veut faire
* **Pourquoi** : pourquoi il veut faire ceci Elle est confirmée par des critères d’acceptation rédigés au même moment que l’histoire. En tant que, je veux pour atteindre

En tant que « **Qui** », je veux « **Quoi** », afin de « **Pourquoi** »

**Les User Stories de l’utilisateur**

En tant qu'utilisateur, je veux une gamme de symboles standardisés pour représenter le flux de travail et pouvoir ajouter des activités.

En tant qu'utilisateur, je

veux établir des règles métiers pour automatiser certaines décisions et optimiser l'efficacité opérationnelle.

En tant qu'utilisateur, je

veux relier les activités du processus aux fonctionnalités existantes de la solution Elise.

En tant qu'utilisateur, je

Veux suivre l'état d'avancement des processus en cours d'exécution distinguer chaque activité réussie or échouée.

En tant qu'utilisateur, je

veux changer la couleur de chaque activité pour une meilleure visualisation et organisation du processus métier.

En tant qu'utilisateur, je

veux Générer des rapports personnalisés sur les performances des processus en analysant des indicateurs clés.

En tant qu'utilisateur, je veux sauvegarder les processus dans la base de données.

En tant qu'utilisateur, je veux revenir au diagramme initial

En tant qu'utilisateur, je

Veux Importer un processus dans l'application ou le sauvegarder une fois créé (local Storage) ou modifié.

**Les User Stories de l’utilisateur**

En tant qu'utilisateur, je veux voir les activités qui ont une erreur ou une propriété manquante.

En tant qu'utilisateur, je veux ajouter des commentaires pour chaque activité afin de faciliter la collaboration et la traçabilité des actions.

En tant qu'utilisateur, je veux Zoomer ou Dézoomer sur le diagramme de processus

En tant qu'utilisateur, je peux voir les raccourcis clavier.

**Les User Stories de l’administrateur**

En tant qu'utilisateur, je veux gérer les autorisations des utilisateurs pour accéder, modifier ou exécuter divers processus afin de protéger et de protéger les données.

## **Product Backlog :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nom du Sprint** | **Num** | **User Story** | **Description** | **Priorité** | **Dod** |
| **Sprint 1- Etude de l'environnement** | TS 1 | Installer l'environnement du travail | En tant que Scrum team, nous voulons initialiser l'environnement de développement, afin de pouvoir commencer à travailler sur le projet et livrer efficacement les histoires utilisateur.//installer | Haut | **L'équipe a installé et configuré les logiciels et outils nécessaires, y compris mais sans s'y limiter l'IDE, le système de contrôle de version, les outils de construction et les bibliothèques de développement.** |
| TS2 | Formation sur la solution Elise | En tant que membre de l’équipe SCRUM, je souhaite comprendre pleinement la fonctionnalité du système Elise. | Haut |  |
| TS 3 | Maîtriser le développement avec les Framework ASP.NET, Elsa Workflow et BPMN.js | En tant qu'équipe Scrum, nous voulons maîtriser l'utilisation d'Elsa Workflow et développer une application Vue.js avec BPMN.js. Notre équipe va acquérir les connaissances et les compétences nécessaires pour développer une application en utilisant le Framework ASP.NET, | Haut | **L'équipe a acquis une connaissance approfondie d’ELSA Workflow, BPMN.js et ASP.NET** |
| **Sprint 2 – Personnaliser le panneau des propriétés** | TS 4 | Intégrer un panneau de propriétés personnalisé avec une activité dans un diagramme BPMN | En tant que Scrum team, nous voulons étudier différentes méthode pour ajouter propriété panel pour chaque activité | Haut | **Lorsque l'utilisateur clique sur une activité, afficher le panneau des propriétés correspondant en conséquence** |
| US 5 | Création un Descripteur | En tant que Scrum Team, nous souhaitons créer un nouveau descripteur propre à la société Neoledge, que nous choisirons de nommer "neo", pour gérer les propriétés personnalisées dans le processus. | Moyenne | **Ajouter descripteur spécifique pour l'entreprise** |
| US 6 | Ajouter des Commentaires | En tant qu'utilisateur, je souhaite ajouter des commentaires pour chaque activité afin de faciliter la collaboration et la traçabilité des actions | Haut | **l'utilisateur peut ajouter des commentaires** |
| US 7 | Supprimer des Commentaires | En tant qu'utilisateur, je souhaite supprimer des commentaires pour chaque activité, | Haut | **l'utilisateur peut supprimer des commentaires** |
| US 8 | Ajouter des propriétés spécifiques pour chaque activité | En tant qu'utilisateur, je veux ajouter des propriétés dans mon activité inclure la gestion des tâches, des flux de travail, des règles métier et des événements. | Haut | **l'utilisateur peut ajouter des propriétés** |
| US 9 | Supprimer des propriétés spécifiques pour chaque activité | En tant qu'utilisateur, je veux supprimer des propriétés dans mon activité | Moyenne | **l'utilisateur peut supprimer des propriétés** |
| US 10 | Modifier des propriétés spécifiques pour chaque activité | En tant qu'utilisateur, je veux modifier des propriétés dans mon activité | Moyenne | **l'utilisateur peut modifier des propriétés** |
| **Sprint 3 - Gestion des processus (digramme BPMN)** | US 11 | Télécharger diagramme | En tant qu'utilisateur, je veux télécharger diagramme (fichier XML, fichier SVG) | Haut | **l'utilisateur peut télécharger diagramme XML ou bien SVG** |
| US 12 | Importer des diagrammes existants | En tant qu'utilisateur, je veux importer un diagramme | Haut | **l'utilisateur peut importer un diagramme** |
| US 13 | Réinitialisation digramme | En tant qu'utilisateur, je veux revenir au diagramme initial | Moyenne | **l'utilisateur peut revenir diagramme initial** |
| US 14 | Changer Couleur des activités | En tant qu'utilisateur, je veux changer couleur des activités du diagramme | Haut | **l'utilisateur peut changer couleur** |
| US 15 | Voir les raccourcis clavier | En tant qu'utilisateur, je peux voir les raccourcis clavier. | Faible | **l'utilisateur peut voir les raccourcis clavier.** |
| US 16 | Ajouter un module pour afficher les erreurs (Linter). | En tant que développeur, ajouter des erreurs pour chaque activité. | Haut | **Développeur peut ajouter des erreurs** |
| US 17 | Consulter les erreurs de chaque activité | En tant qu'utilisateur, je veux voir les activités qui ont une erreur ou une propriété manquante. | Haut | **l'utilisateur peut voir les erreurs** |
| US 18 | Zoomer ou Dézoomer au diagramme | En tant qu'utilisateur, je veux zoomer ou Dézoomer au diagramme | Moyenne | **l'utilisateur peut zoomer ou Dézoomer** |
| **Sprint 4 - Analyse et Extraire Processus BPMN** | US 19 | Sauvegarder Processus | En tant qu'utilisateur, je veux sauvegarder les processus dans la base de données | Moyenne | **Le processus est enregistré dans la base de données et dans un dossier appelé "fichiersBpmn" dans le back end.** |
| US 20 | Etude de la Solution optimale pour extraire des données de processus dans le back end | En tant qu'équipe Scrum, nous voulons étudier différentes techniques afin de déterminer la solution optimale pour extraire les données des diagrammes. | Haut | **Les informations du processus sont enregistrées au format JSON dans le corps de la requête et sont reçues par modèle dans le back end.** |
| **Sprint 5 – Exécution Processus Workflow** | US 21 | Intégrer des fonctionnalités d'exécution des processus par Elsa Workflow | En tant que développeur, exécuter des processus avec des workflows | Haut | **Exécuter les processus du diagramme** |
| US 22 | connecter avec api ELISE |  | Haut |  |
| US 23 | Donner des informations sur les activités effectuées ou non. | En tant qu'utilisateur, je veux suivre l'état d'avancement des processus en cours d'exécution distinguer chaque activité réussie or échouée. | Haut | **L'utilisateur peut savoir si une activité est correctement exécutée ou non grâce à une icône dans l'activité.** |
| **Sprint 6 - Tableaux de bord et rapports** | US 24 | Génération Tableaux de Bord | En tant qu'utilisateur, je veux sauvegarder les processus dans la base de données | Moyenne | **Le processus est enregistré dans la base de données et dans un dossier appelé "fichiersBpmn" dans le back end.** |
|  |  |  |  |  |

Tableau 1 : Product Backlog

Les sprints que nous avons identifiés ont été répartis en trois releases qui se présentent comme suit :

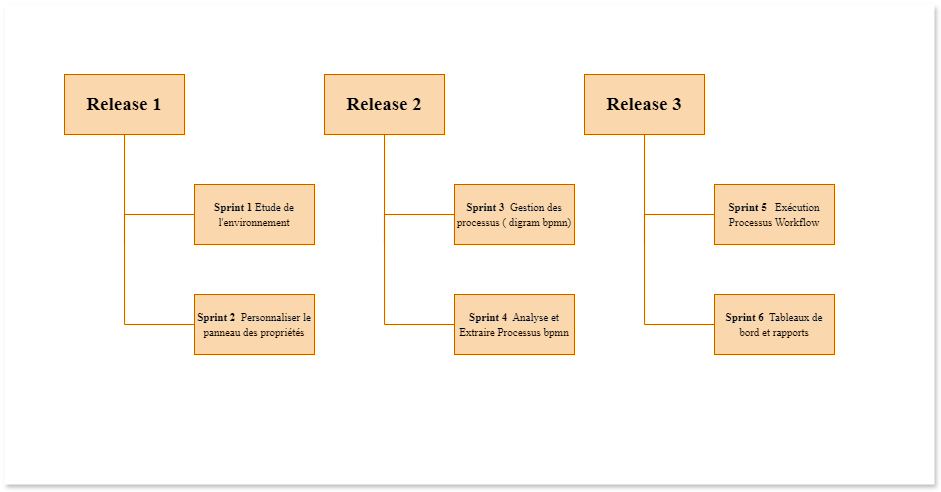


Figure 5 : Organisation des releases

**Conclusion**

Au cours de ce chapitre nous avons présenté les différents acteurs qui vont interagir avec notre application. Ensuite, nous avons cité les User stories, regroupées dans le Product Backlog qui décrit les fonctionnalités de chaque acteur et la répartition de ces User Stories.

Dans le chapitre suivant, nous avons présenté le premier release de notre application.

# **Chapitre 3 : Release 1**

## **Introduction**

Dans ce chapitre, nous allons nous concentrer sur les sprints du premier release

de notre projet. Nous avons présenté en détail différents éléments essentiels tels que

le diagramme des classes et les diagrammes des séquences. Enfin, les interfaces graphiques de chaque sprint

Ces éléments permettront d’assurer une compréhension approfondie de la fonctionnalité

de chaque sprint et de guider la réalisation de la première release.

## **Organisation des sprints :**

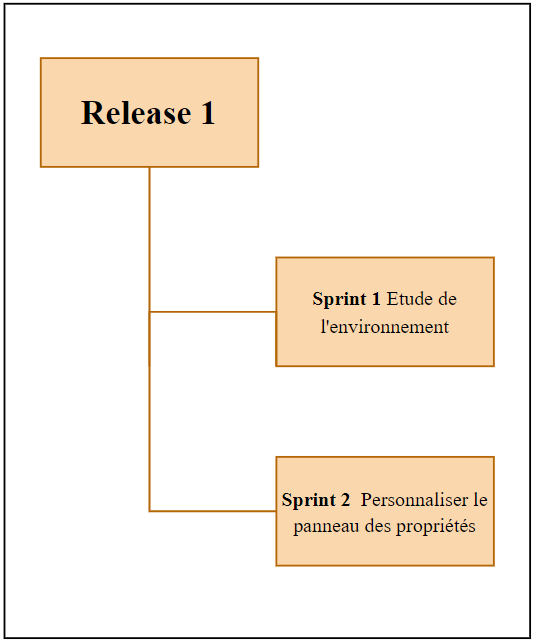


Figure 6 : Release 1

## **Sprint 1 « Etude de l’environnement »**

### **Sprint Goal**

Ce sprint vise à faire une étude approfondie de l'environnement de travail et de tester les

différentes solutions envisageables afin de trouver l’outil le plus adapté pour garantir la

réussite de notre projet.

### **Sprint Backlog :**

Le tableau ci-dessous présente le Backlog de notre premier sprint :

|  |  |  |  |
| --- | --- | --- | --- |
| **User Stories** | | **Les taches** | **Période** |
| **Sprint 1** | **Installer l'environnement du travail** | Installer Visual Studio 2022 | 1j |
| Installer le .NET Framework |
| Microsoft SQL Server Management Studio 19 |
| **Formation sur la solution Elise** | suivre une formation préparée par la société sur Elise. | 2j |
| **Maîtriser le développement avec les Framework ASP.NET, Elsa Workflow et BPMN.js** | Suivre une formation sur YouTube qui explique les notions de base du .NET CORE 6. | 11j |
| Développer une application Gérer taches avec authentification et JWT pour maitriser les concepts de .NET et tester avec Postman. |
| Suivre une formation sur YouTube qui explique les notions de base du Elsa Workflow |
| Intégration Elsa Workflow dans, NET avec deux manières   * Designer Workflow pour une meilleure visualisation * Programmatique Workflows |
| Développer des mini workflows pour tester et comprendre mieux comment fonctionne Elsa Workflow Designer et Programmatique. |
| Intégration BPMN.js dans Vue.js 3 Prime Vue. |
|
|
| Développer une application qui offre la possibilité de créer un diagramme BPMN en utilisant Vue.js, puis de le transmettre à une API.NET afin de pouvoir l'exécuter dans des processus de production. Au cours de cette procédure, l'identifiant de l'utilisateur a été intégré au diagramme. Dans les flux de travail d'Elsa, on extrait le diagramme afin d'obtenir cet identifiant, puis on sollicite une autre API pour récupérer l'utilisateur correspondant à cet identifiant. |

Tableau 2 : Sprint Backlog du sprint 1

### **Implémentation du sprint :**

Cette partie contient des captures écrans pour nos tâches réalisées dans ce sprint :

1. **Installer l'environnement du travail :**

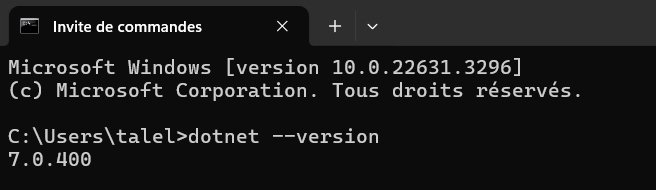


Figure 7 : Installer le .NET Framework

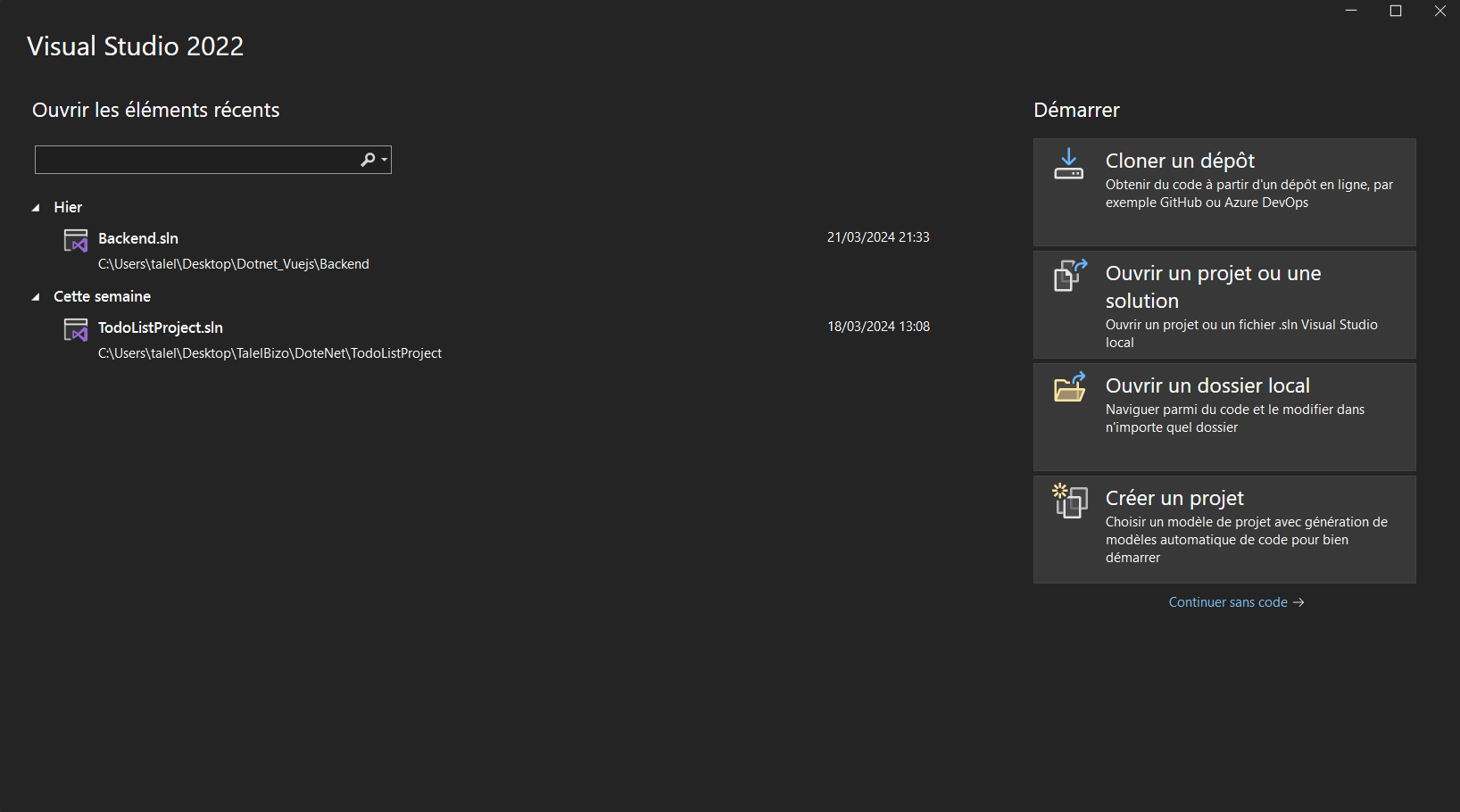


Figure 8 : Installation Visual Studio 2022

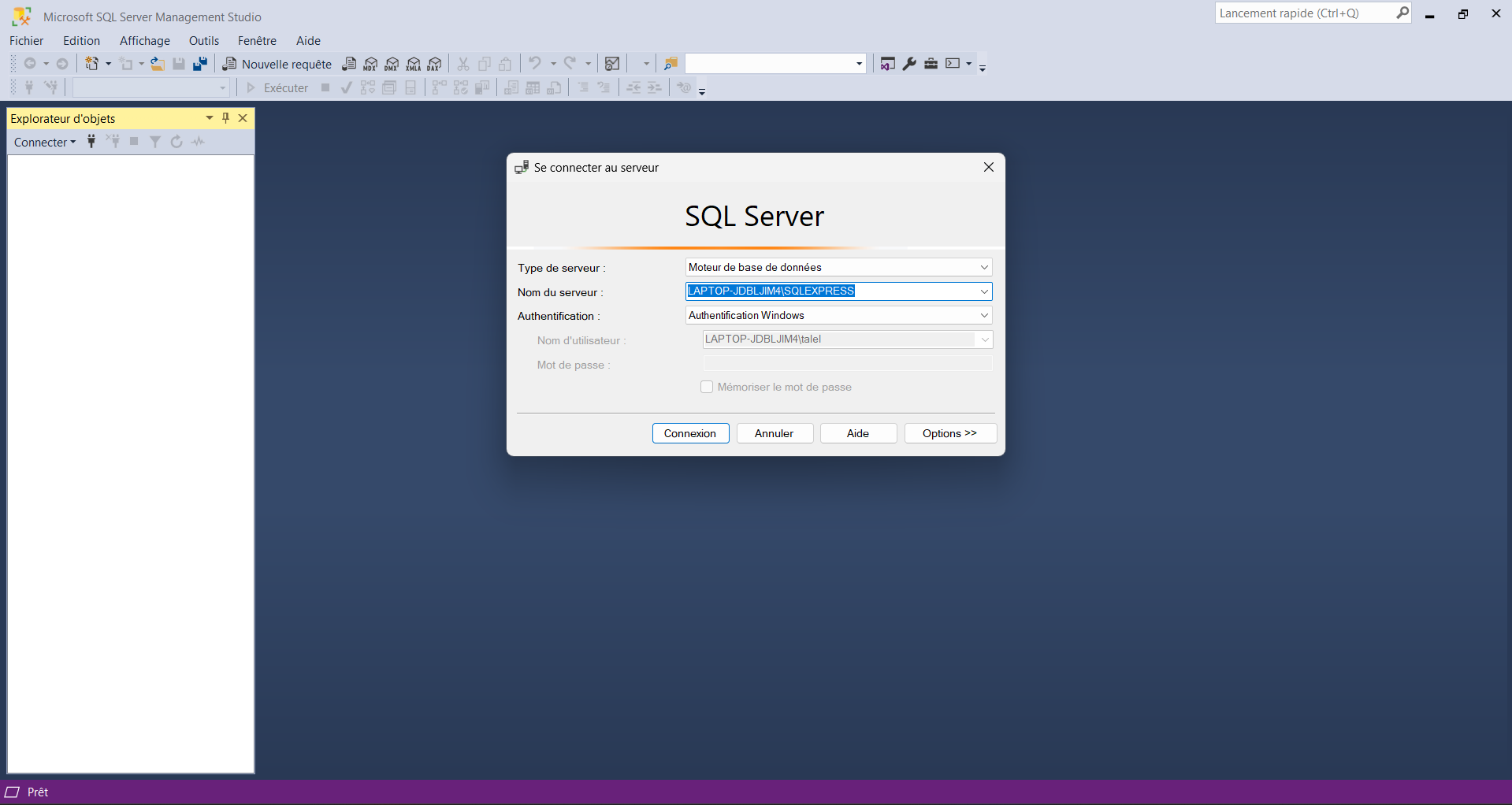


Figure 9 : Microsoft SQL Server Management Studio 19

1. **Formation sur la solution Elise:**
2. **Maîtriser le développement avec les Framework ASP.NET, Elsa Workflow et BPMN.js:**

Dans cette partie, nous avons reçu une formation approfondie sur.NET et le fonctionnement

des Elsa Workflow dans YouTube, en plus de la création de plusieurs mini-applications pour mieux comprendre les concepts d'Elsa et.NET. Nous avons aussi examiné comment BPMN.js peut être intégré à Vue.js, en collaborant avec ces technologies. En outre, nous avons effectué une intégration manuelle des diagrammes BPMN avec Elsa Workflow pour une meilleure compréhension des mécanismes. Grâce à cette expérience, nos connaissances et compétences dans le domaine ont été renforcées, tout en nous donnant une perspective pratique sur l'utilisation de ces technologies dans des projets réels.

* **Développer une application Gérer les taches avec authentification et JWT pour**

**maitriser les concepts de .NET et tester avec Postman.**

La figure 10 et 10 ci-dessous décrit bien les créations de compte utilisateurs dan

Base de données.

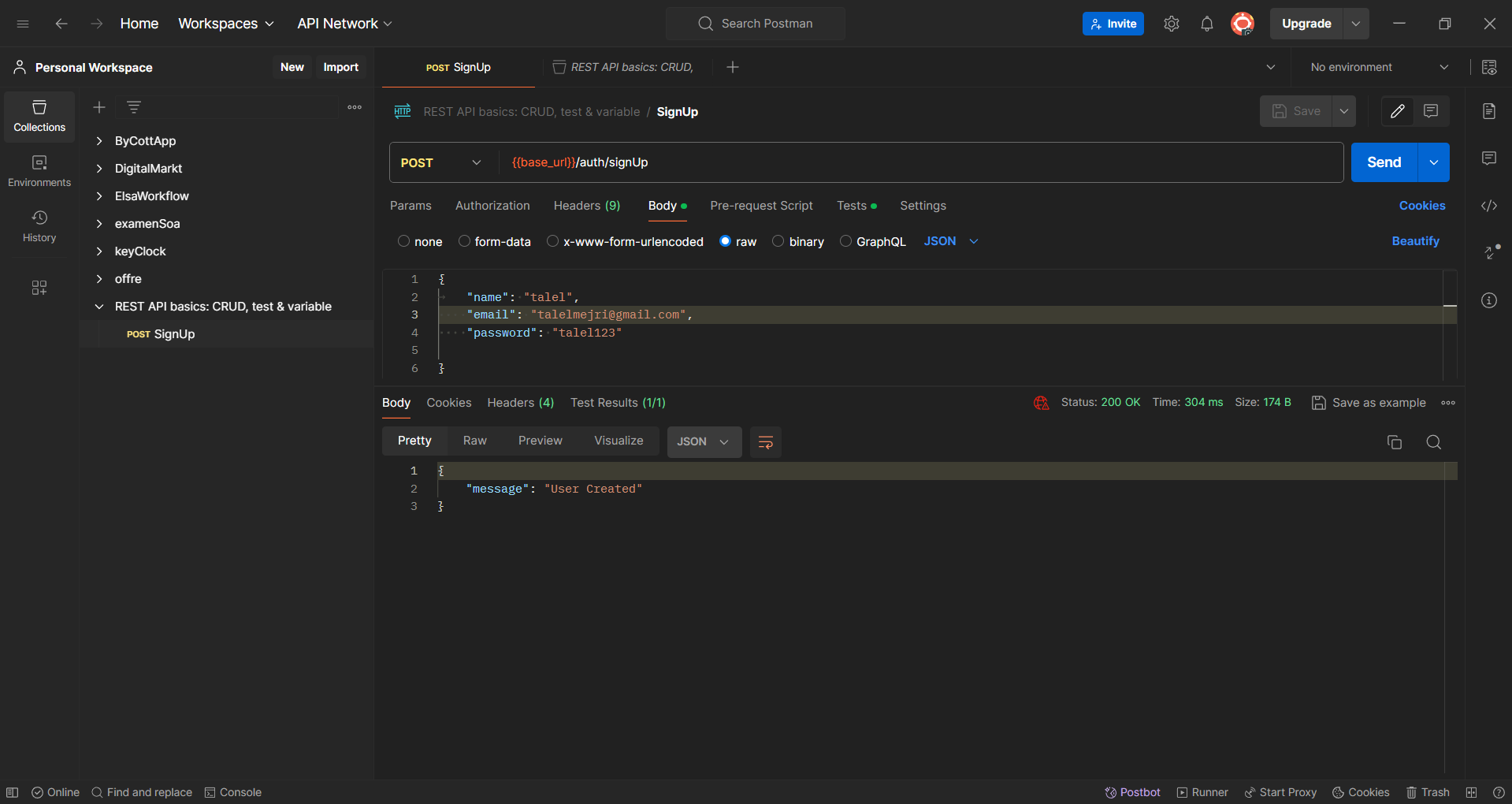


Figure 11 créer un compte

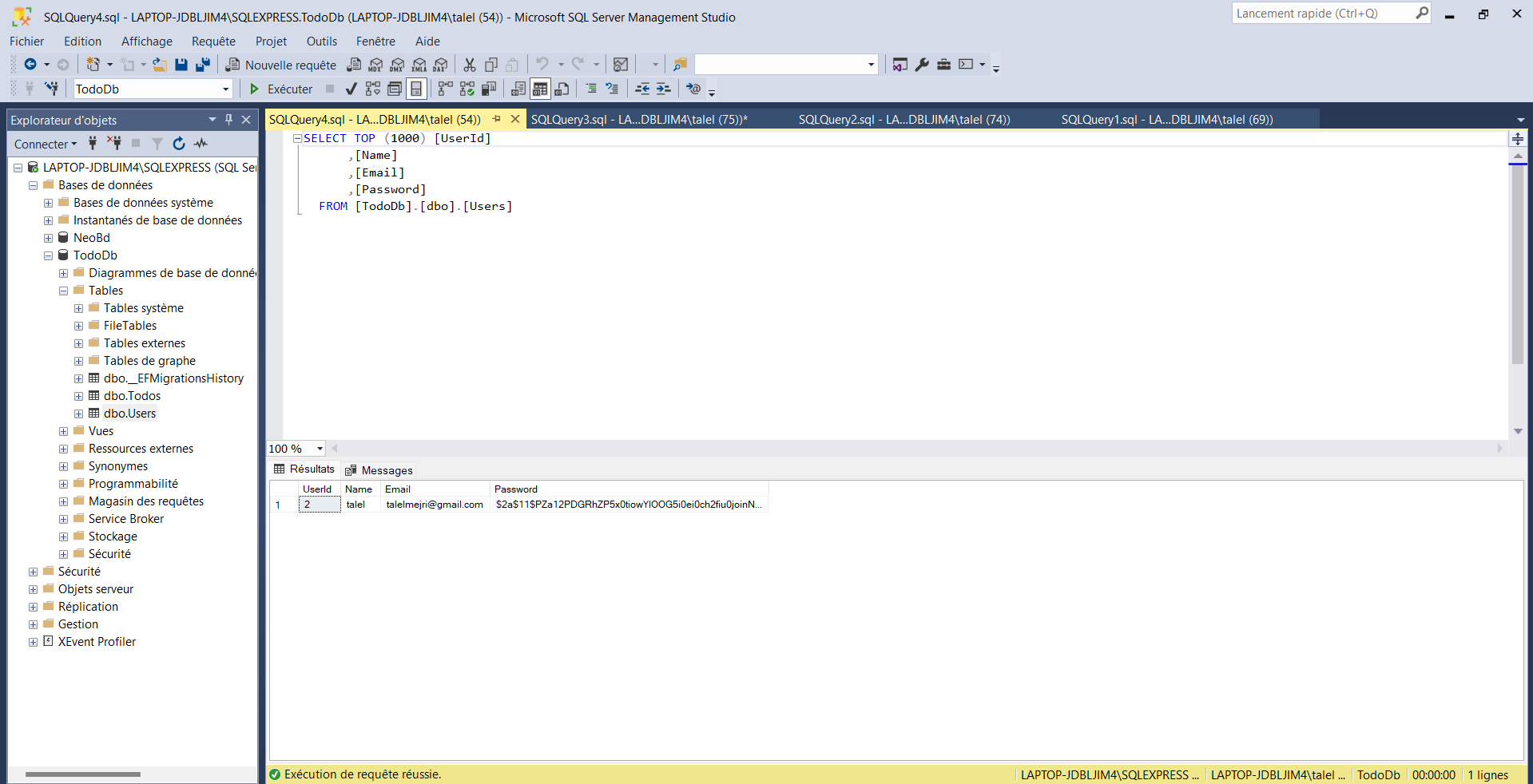


Figure 12 Vérification Dans Base de données

* Authentifier un utilisateur pour voir s'il est dans la base de données et ensuite lui

donner un jeton JWT.

La figure 12 montre clairement ce que j'explique.

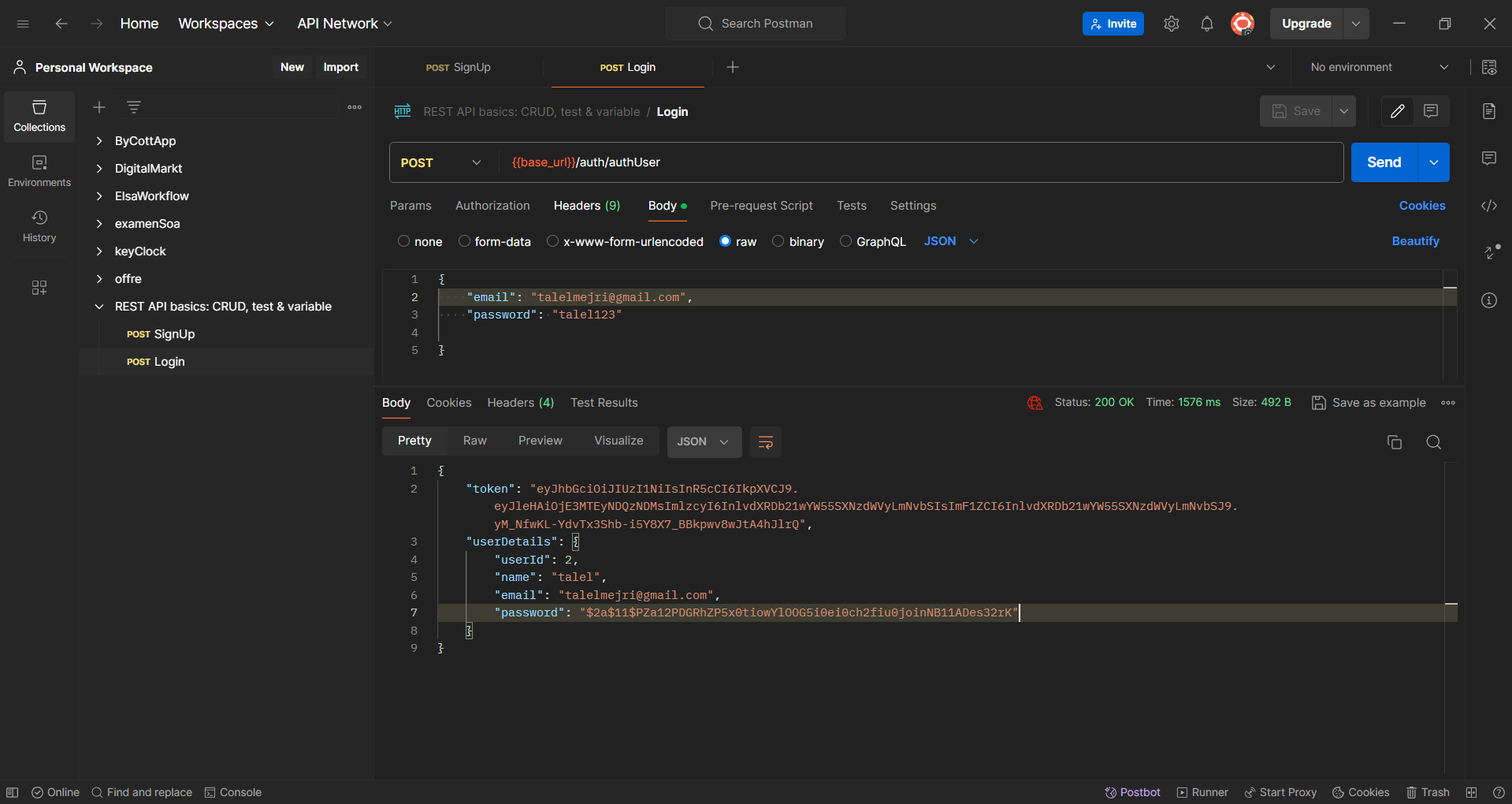


Figure 13 Login User

* Si le jeton n'est pas passé dans les en-têtes (Authorization), mon application ne peut

pas gérer les tâches à faire (todos).

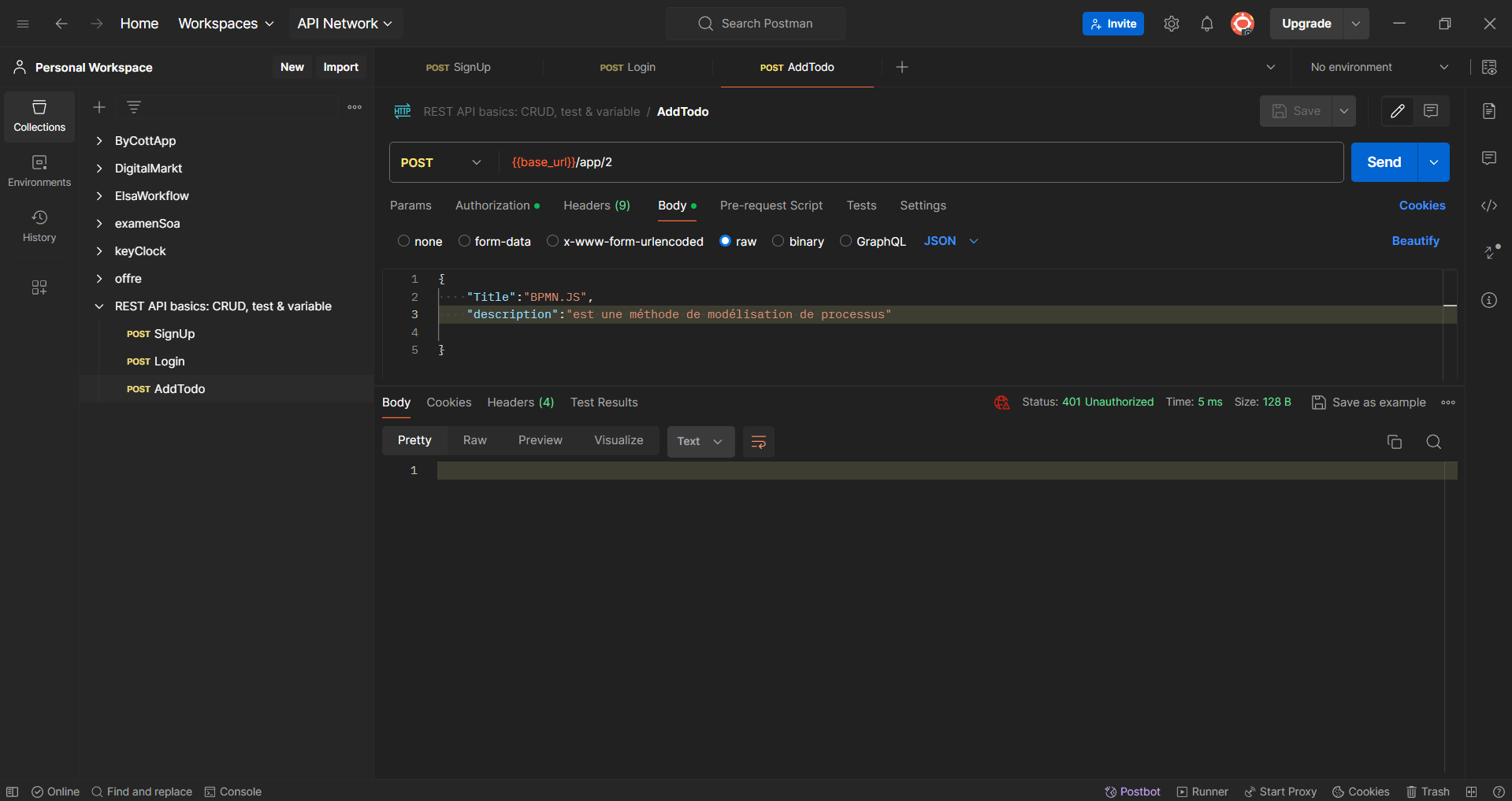


Figure 14 Non autorisé à ajouter une tâche

* Vous pouvez effectuer des opérations CRUD sur les todos lorsqu'un jeton est passé

dans les en-têtes. Les figures ci-dessus expliquent comment ajouter, afficher, modifier et supprimer.

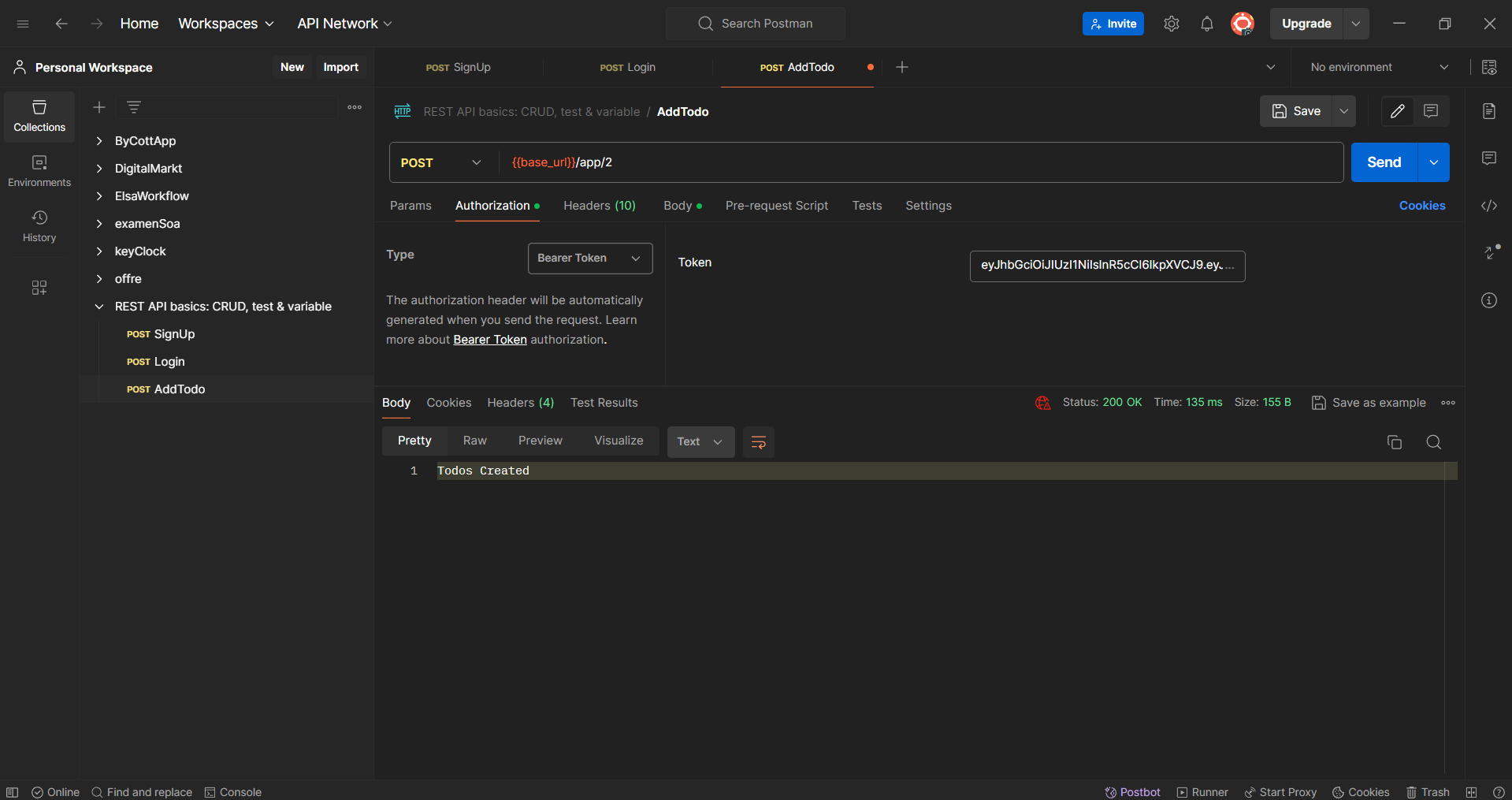
****

Figure 15 Ajouter une tache

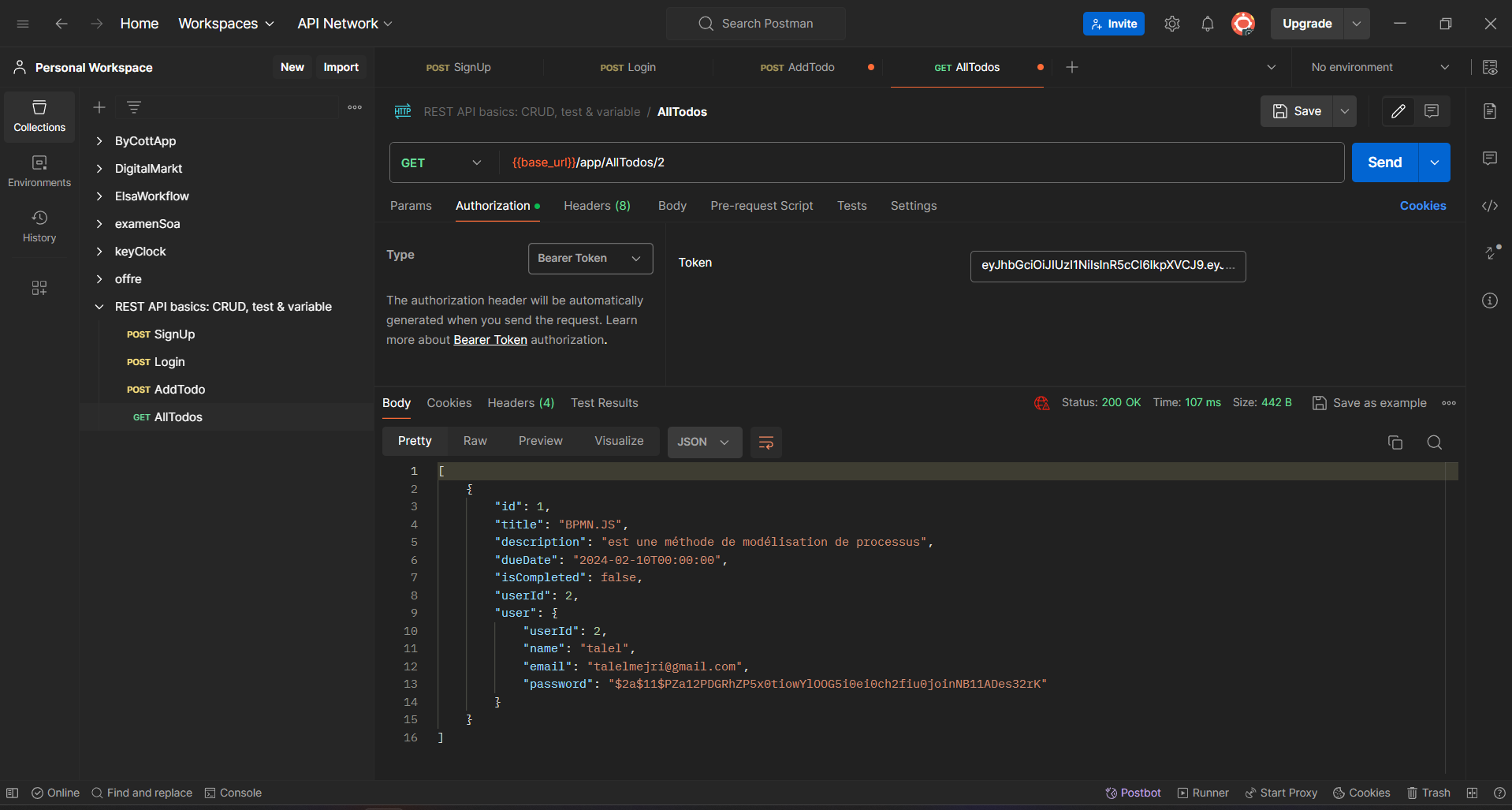
****

Figure 16 Afficher les tache

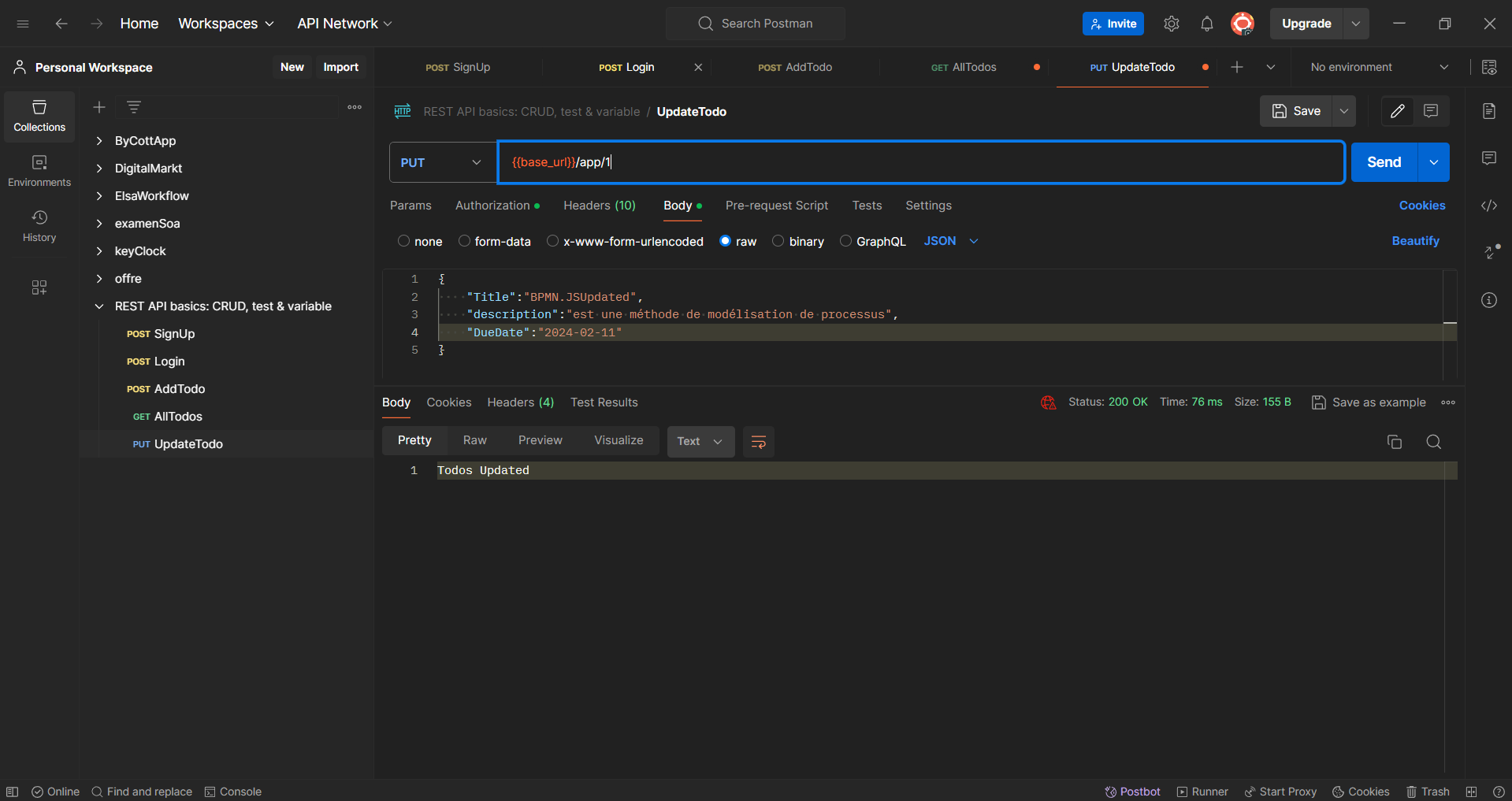
****

Figure 17 Modifier une tache

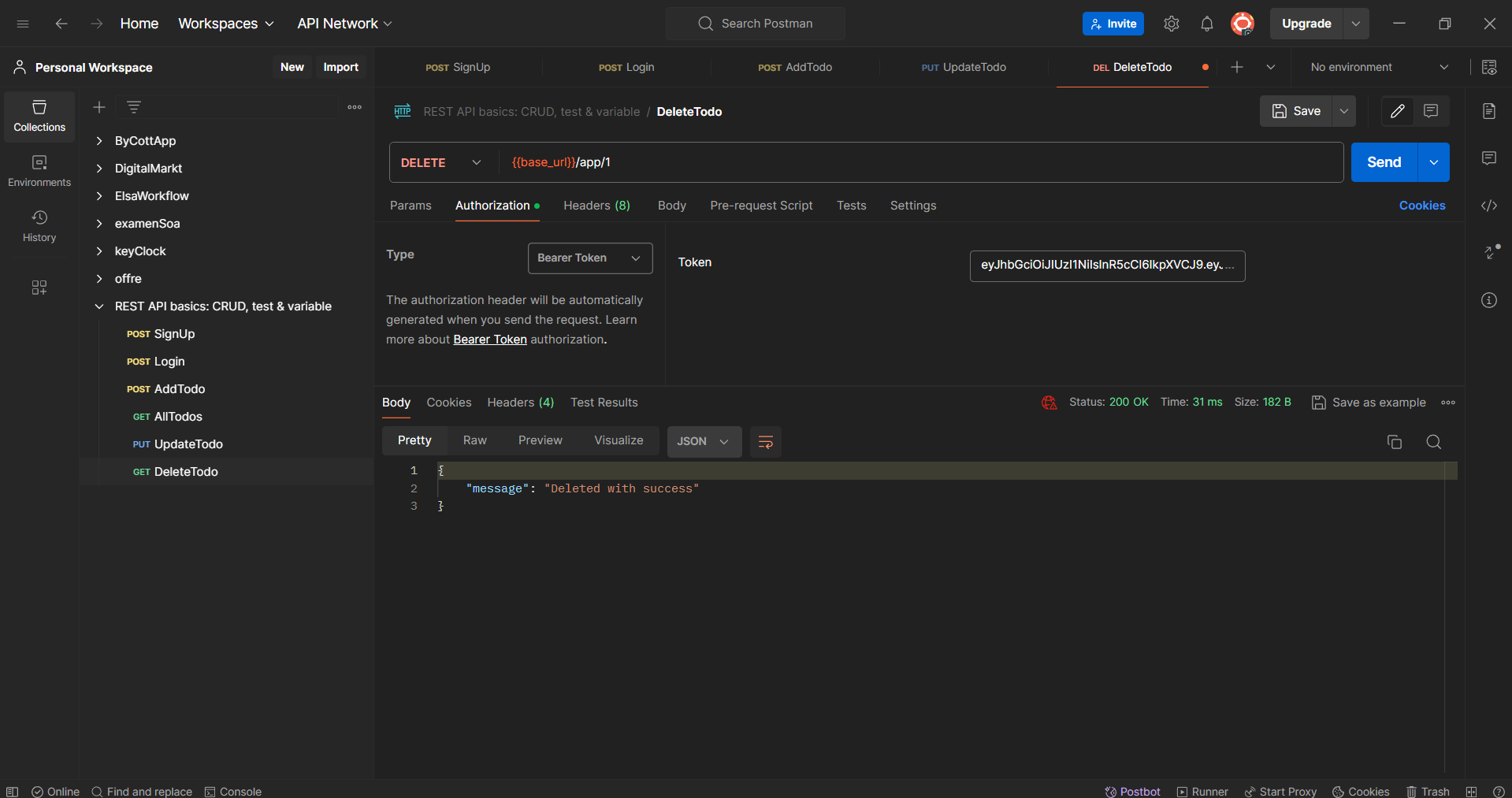
****

Figure 18 Supprimer une tache

* Des captures d’écran du **ELSA** **designer** workflow pour une meilleure visualisation :

La figure 18 ci-dessous décrit bien la visualisation de workflow

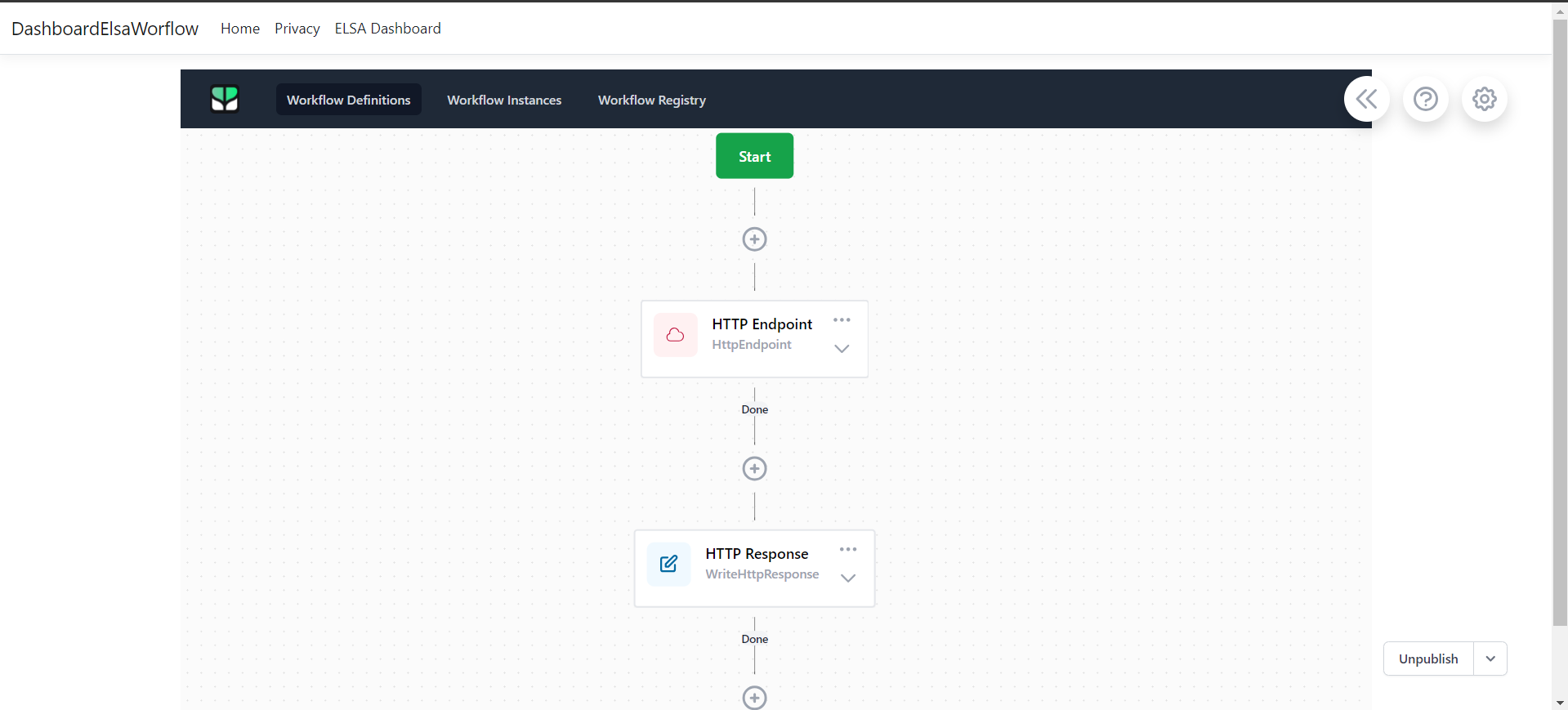


Figure 19 Dashboard ELSA workflow

* Dans le Dashboard, je peux choisir le type de requête et l'endpoint.

La figure ci-dessous décrit bien ce que j'explique.

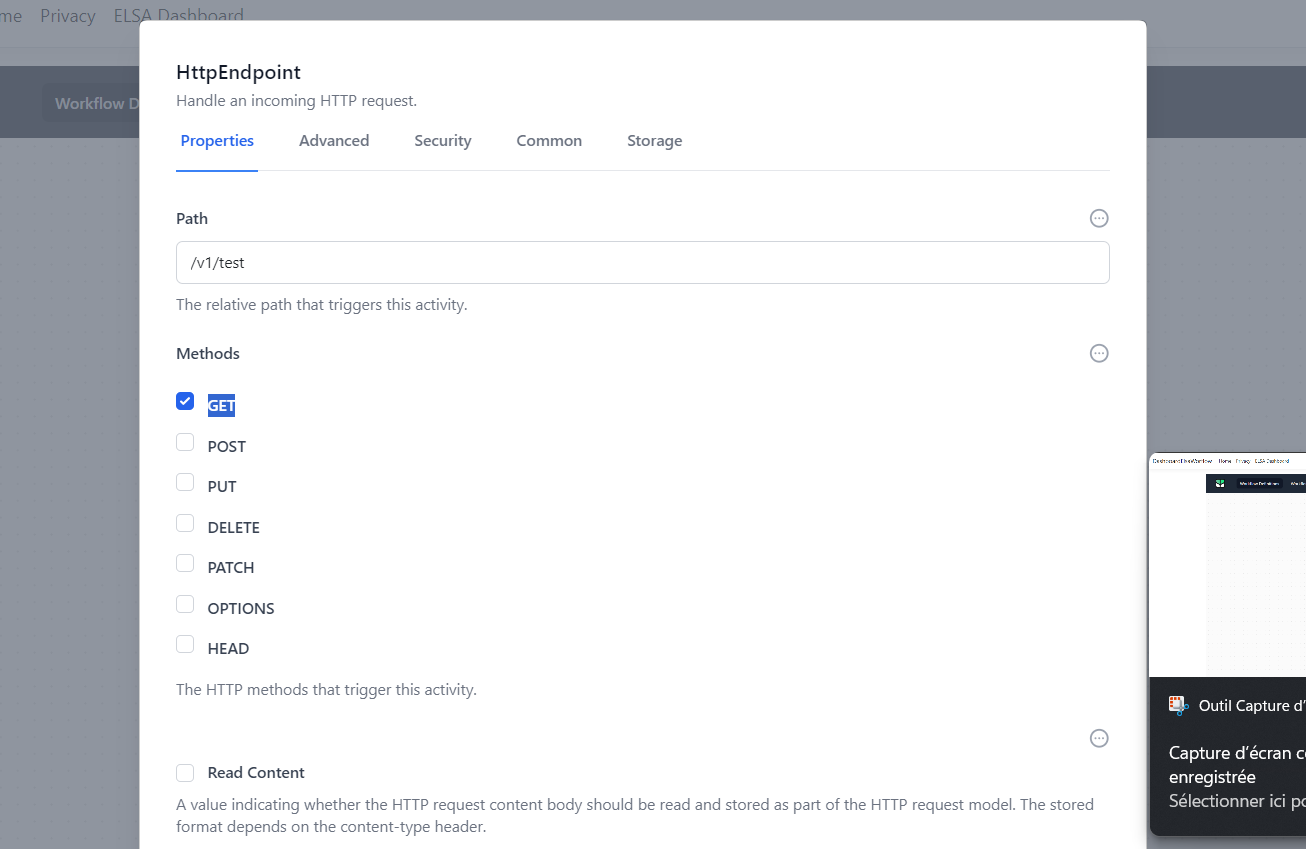


Figure 20 Configure HTTP ENDPOINT

* Je peux également configurer la réponse à la requête.

La figure ci-dessous décrit bien ce que j'explique.

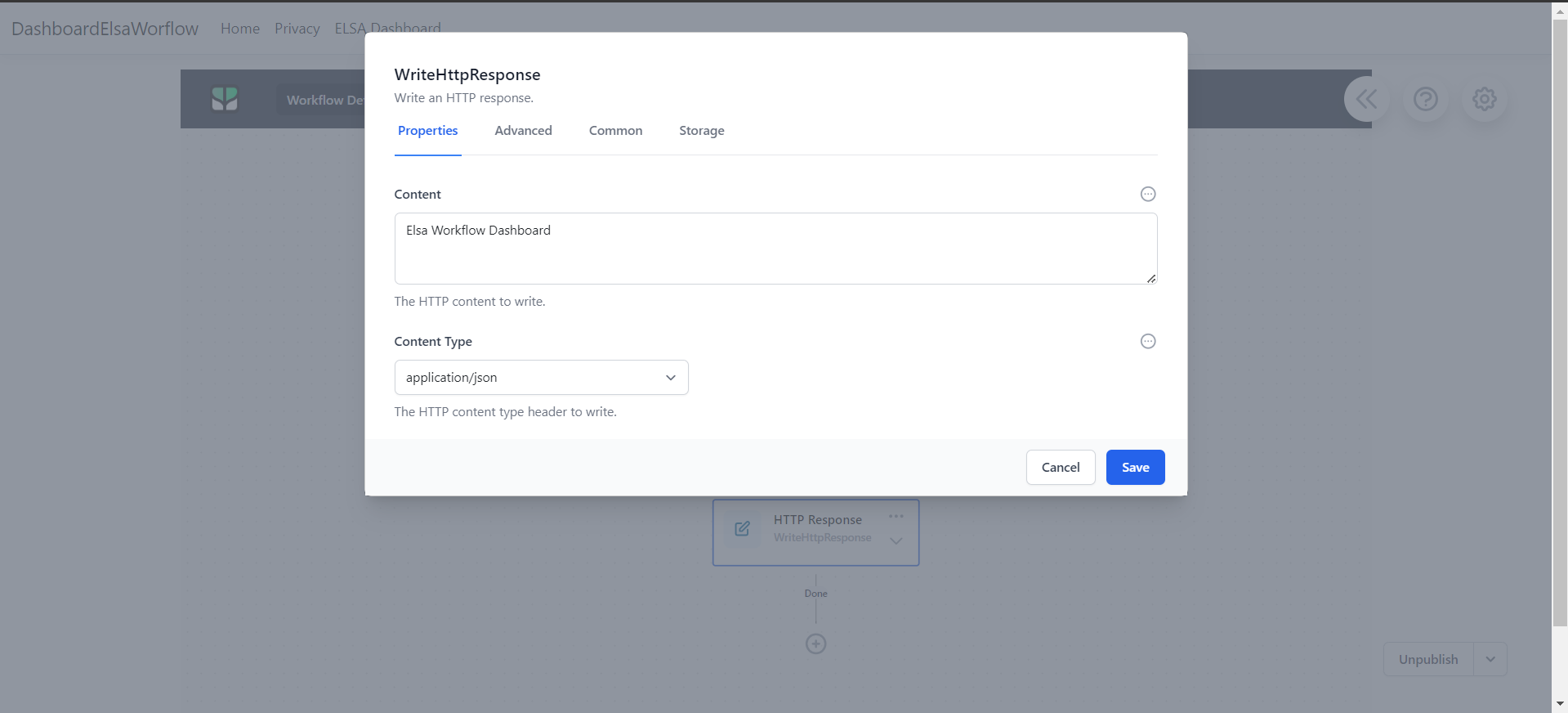


Figure 21 configure HTTP RESPONSE

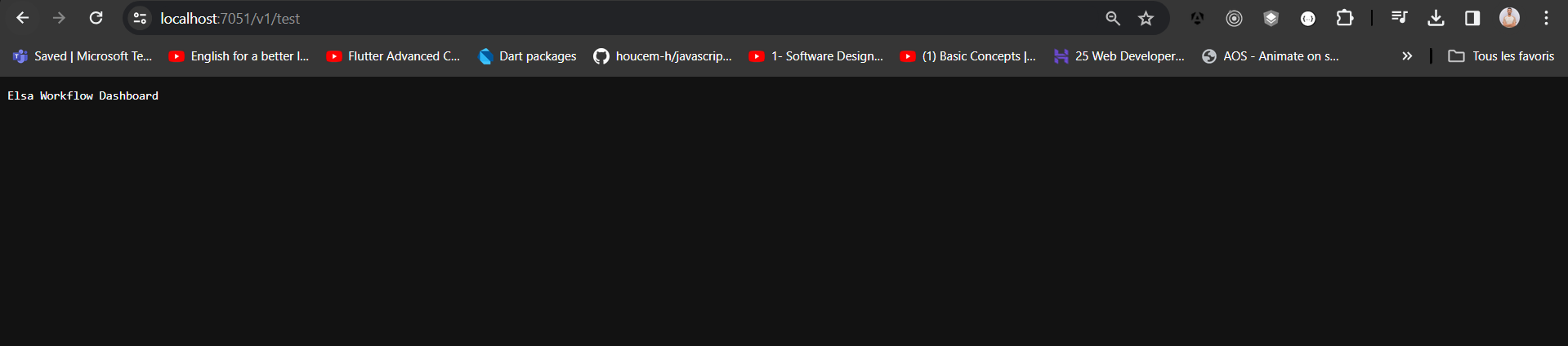
La capture d'écran ci-dessus des tests effectués sont utilisées pour évaluer le fonctionnement du Workflow.

Figure 22 Test Workflow Designer

* **Des Captures d’écran du ELSA Workflow Programmatique**

Je vais montrer un exemple qui ressemble à celui du concepteur mais qui a été créé par

programme. La configuration dans le fichier ‘’program.cs ‘’ est la première étape, puis la gestion du flux de travail via les endpoints et les réponses, suivie des tests sur Postman, comme dans l'exemple.



Figure 23 Configure Elsa Workflow

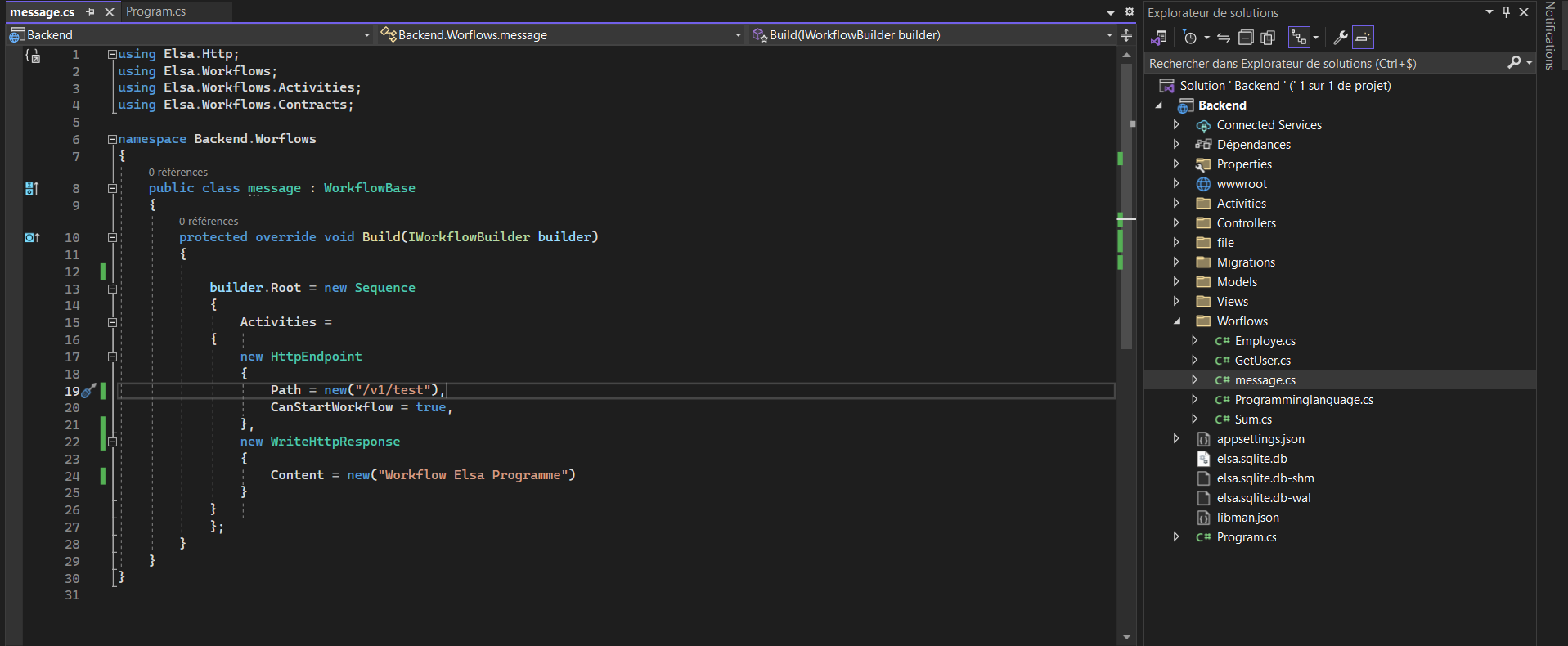


Figure 24 créer le premier workflow

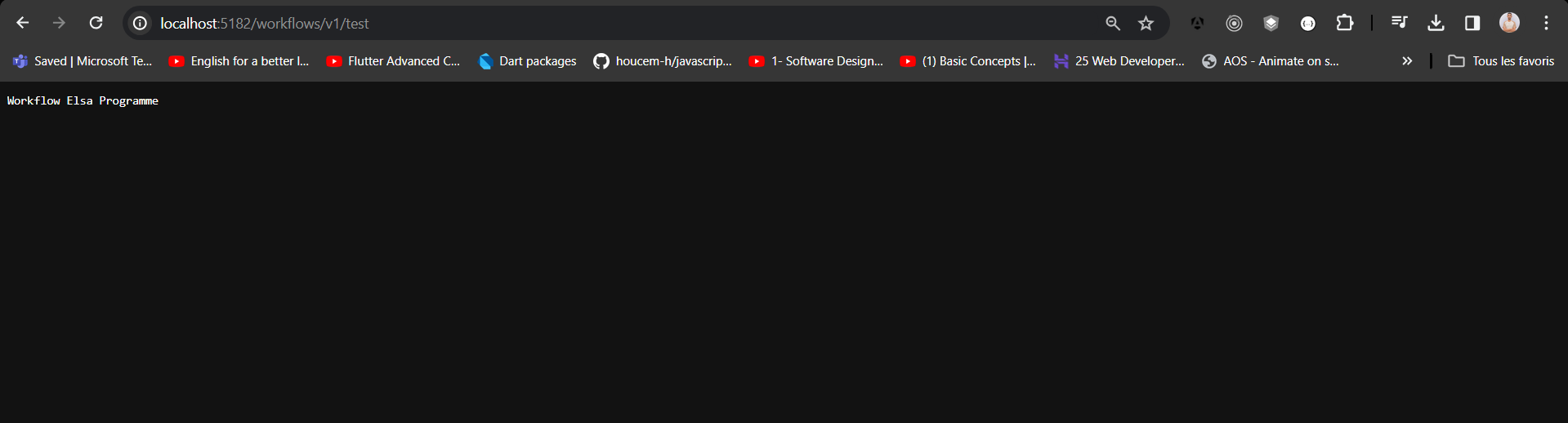


Figure 25 Test Workflow Programmatique

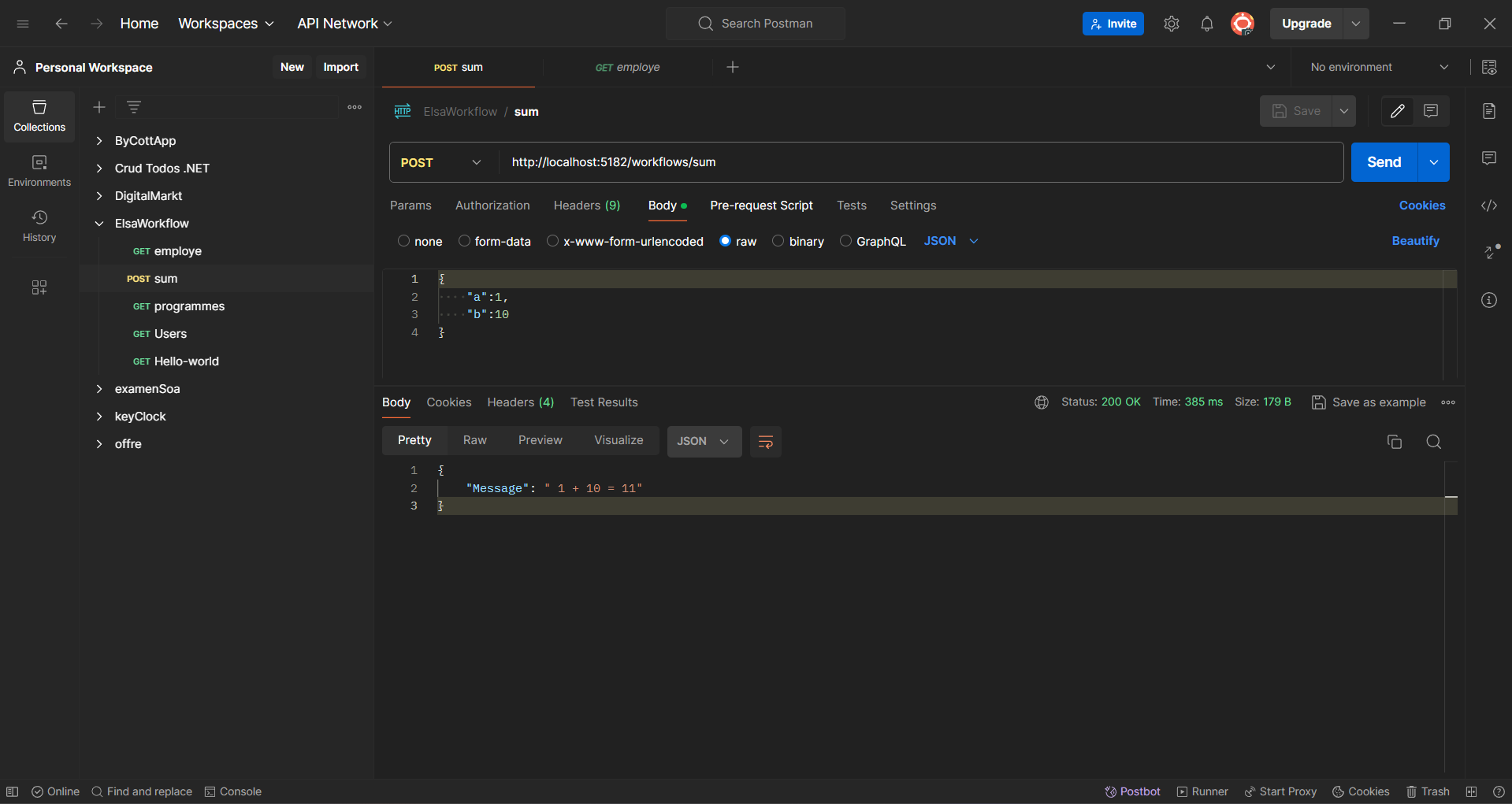


Figure 26 Exemple de Workflow pour calculer le somme

* **Intégration de BPMN.JS dans Vue.Js 3 (Prime Vue) :**

Le succès de l'intégration de BPMN avec Vue.js3 est clairement illustré dans la figure

ci-dessus.

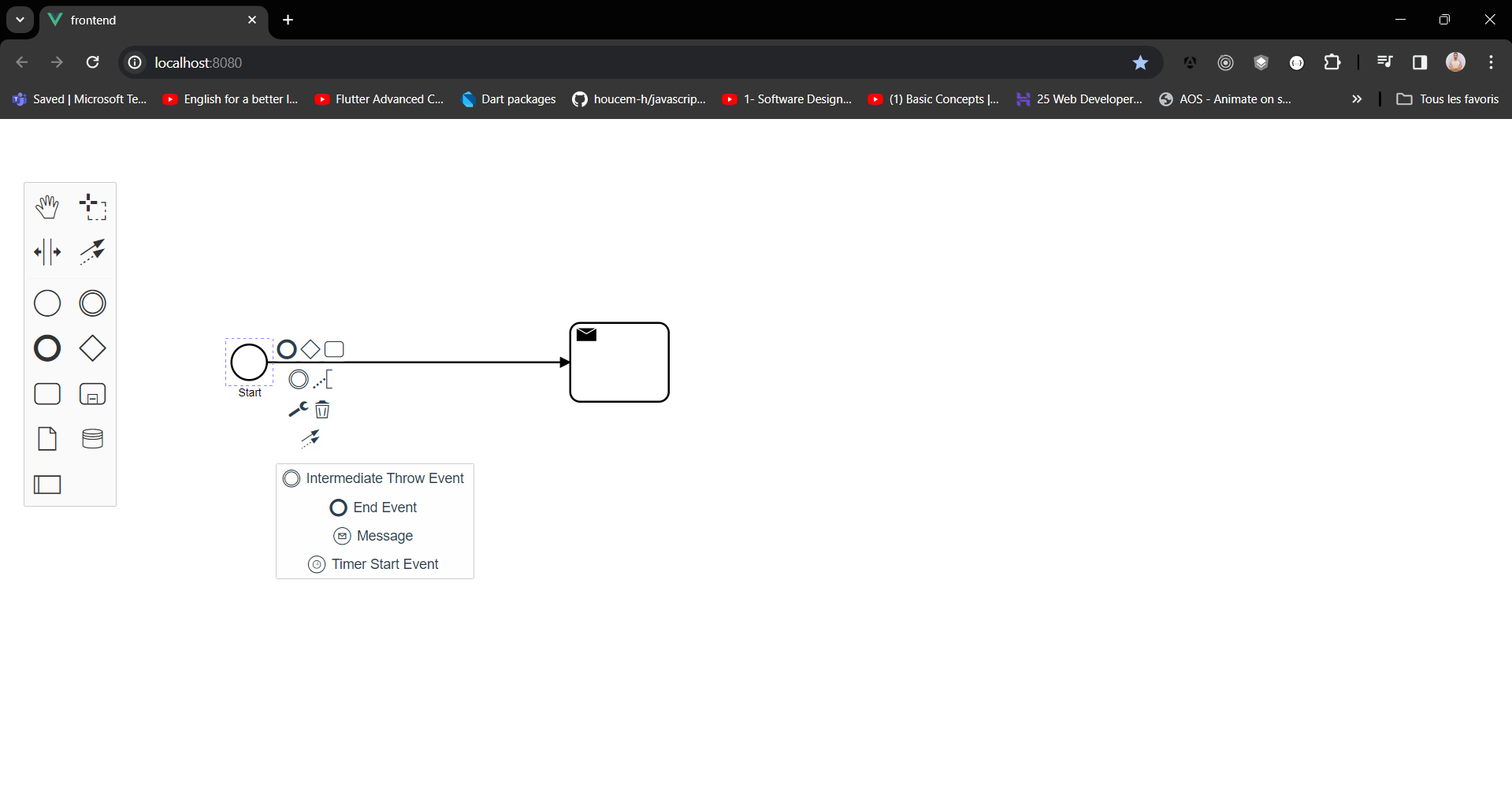


Figure 27 Intégration BPMN.JS dans VueJs3

* Essai pour comprendre le méthode pour exécuter un processus donné par BPMN dans elsa

workflow .

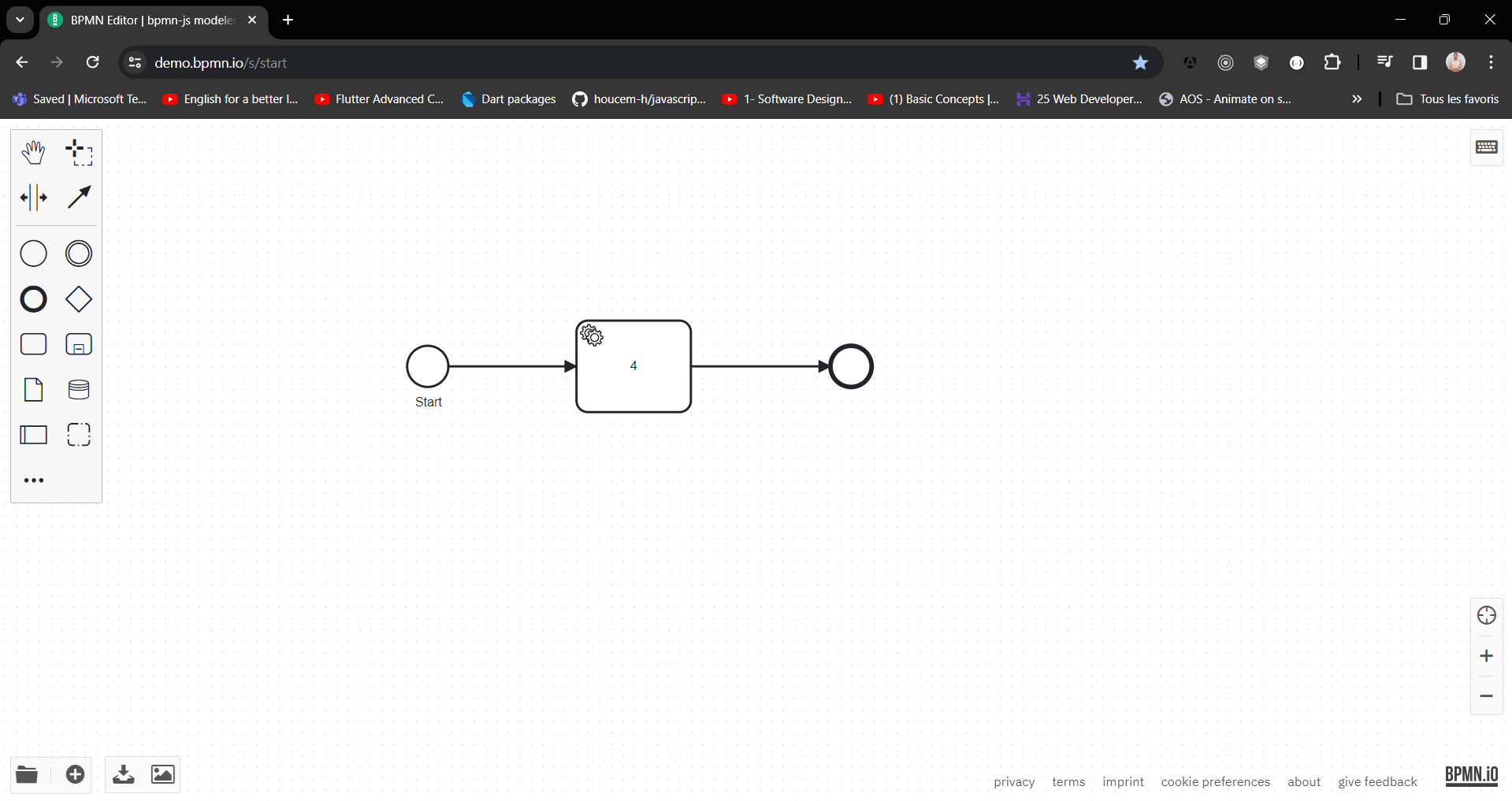


Figure 28 Exemple de gestion d’un digramme en ligne (bpmn.io)

* Exploiter les diagrammes BPMN permet de comprendre leur structure et le type de

code XML associé.de plus, il est important de connaitre la partie de code dont nous avons besoin dans notre travail, c’est-à-dire la partie de balise <processus>.

Pour comprendre sa structure, la figure ci-dessus est un exemple simple.

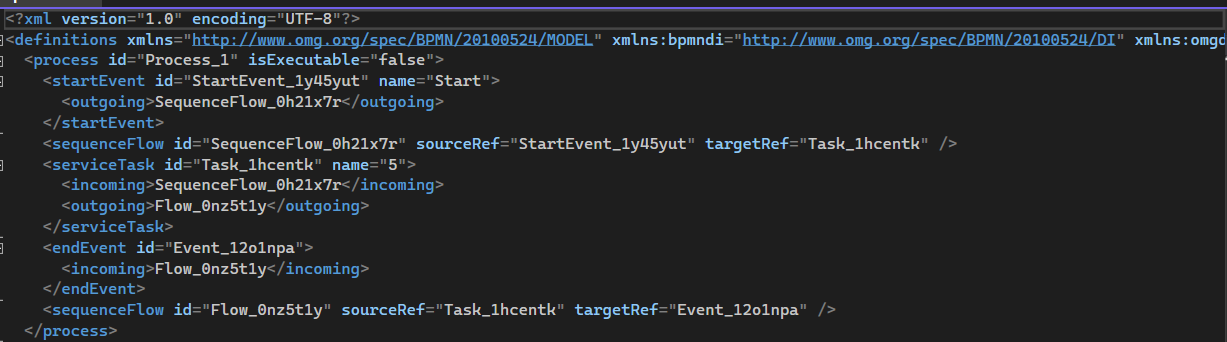


Figure 29 Structure d’un diagramme

* Workflow avec deux activités :
* Extraire le diagramme et obtenir l’ID de l’utilisateur
* Obtenir l’utilisateur par son ID et passer dans le diagramme



Figure 30 Premier Partie de workflow Extraire contenu de diagramme

* L'activité "readpbmn" permet d'obtenir l'identifiant de l'utilisateur passé dans le nom

de l'activité "serverTask".



Figure 31 Activité pour et obtenir l’ID de l’utilisateur

* Je souhaite transférer l'identifiant de l'utilisateur à une autre activité afin de récupérer

leurs informations avec cet identifiant (nom, prénom).

La figure 31 et 33 ci-dessous décrit cette démarche

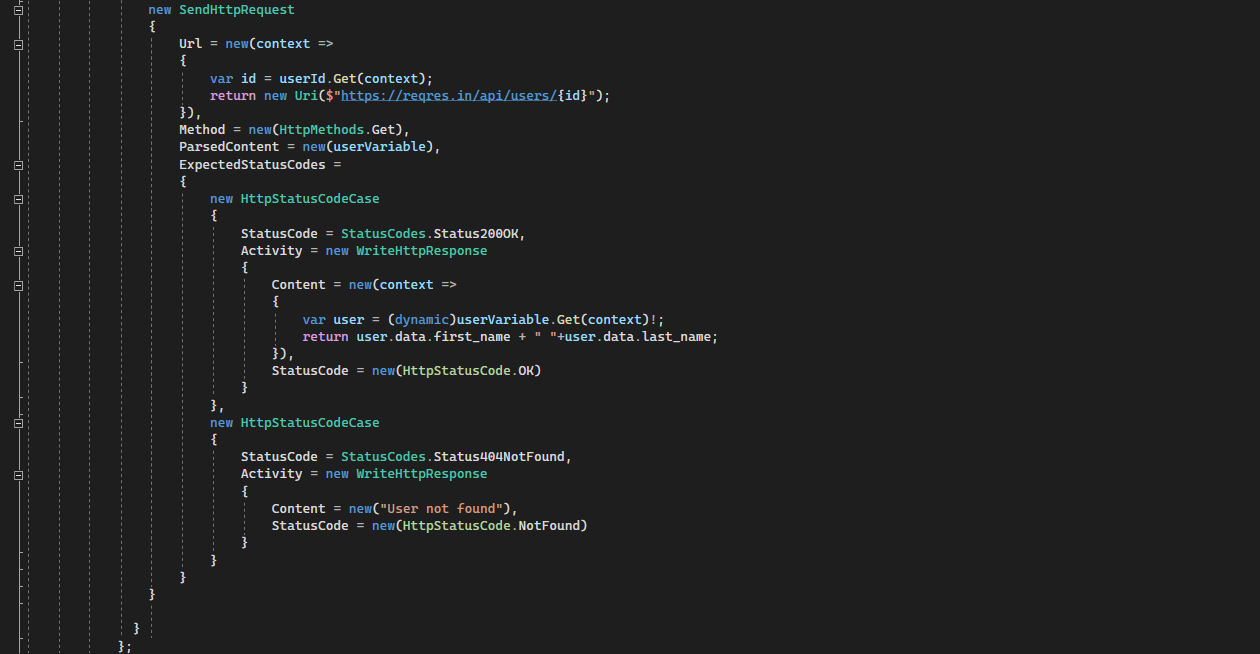


Figure 32 Deuxième Partie de workflow Obtenir l’utilisateur

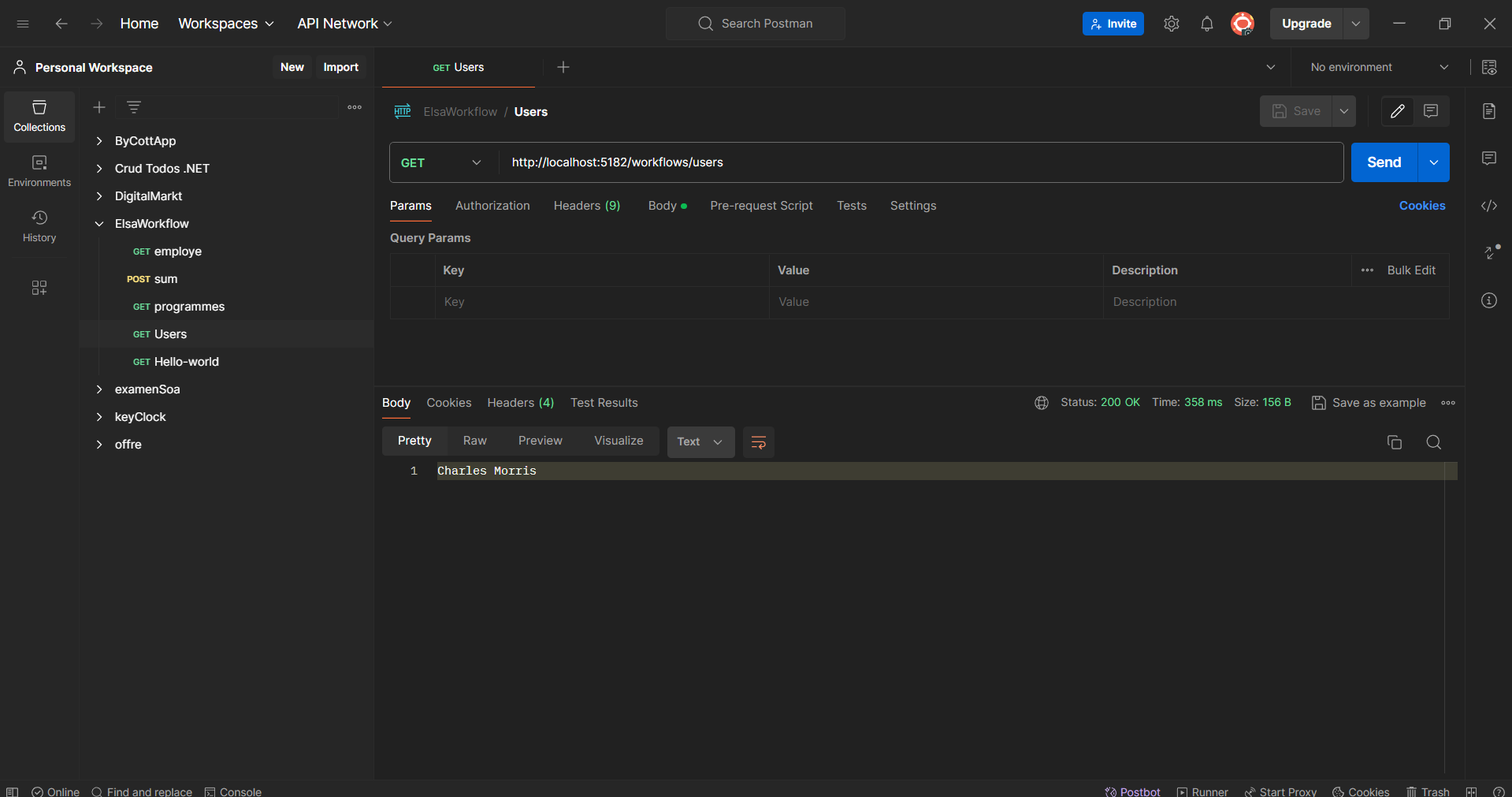


Figure 33 Test Exécution

### **Sprint Review :**

Après avoir terminé ce sprint notre environnement de travail devient prêt afin de pouvoir réaliser les sprints suivants avec succès.

### **Sprint Rétrospective :**

|  |  |
| --- | --- |
| **Ce qui a bien fonctionné** | **Ce qui n’est pas bien passé** |
| - Nous avons installé l'’environnement avec succès. | - Manque de ressource et documentation  - Difficulté niveau de BPMN.Js  - Difficulté niveau du Extraire diagramme |
| - Les formations sur .NET et ELSA Workflow, Intégration BPMN.js des mini applications pour comprendre les environnements. |

Figure 34 Plan d'amélioration du sprint 1

## **Sprint 2 Personnaliser le panneau des propriétés :**

….

## **Conclusion :**

Cette release a été dédié à l'acquisition de compétences sur les nouveaux outils et

frameworks de travail, ainsi que ajouter notre panel de propriétés.

**Chapitre 4 : Release 2**

## **Introduction**

Une fois que nous avons achevé le premier release de notre application, nous sommes prêts à entreprendre les tâches nécessaires pour créer le deuxième release, qui comportera le Product backlog, les différents diagrammes et les interfaces graphiques.

## **Organisation des sprints :**