

EisenFlow : Gestion Intelligente d'Emails

Projet DevOps 1 - Documentation Technique

Alex BRINDUSOIU

Arthur CHAUEAU

Swetha SARAVANAN

Franck ZHENG

Février 2026

Table des matières

1	Présentation Générale	2
1.1	1. Qui sommes-nous?	2
1.2	2. Quoi ? Pourquoi ?	2
1.3	3. À propos du projet	2
1.3.1	Public visé	2
1.4	4. Analyse de Concurrence	3
2	Ingénierie et Fonctionnalités	4
2.1	5. Éléments de gestion de projet et Architecture technique	4
2.2	6. Diagramme de classes	4
2.2.1	6.1 But des features	4
2.2.2	6.2 Scénarios	6
2.2.3	6.3 Wireframe et screenshots	11
2.3	7. Résumé des fonctionnalités	12
3	8. Annexe API REST	12
3.1	Documentation de l'API Mails	12
3.1.1	1. Actions Globales (Collection)	12
3.1.2	2. Actions Spécifiques (Ressource Unique)	13
3.1.3	3. Actions de Délégation	13
3.2	Documentation des Contrôleurs (Application Web)	14
3.2.1	1. Vues Principales (Navigation)	14
3.2.2	2. Actions sur les Mails	14
3.2.3	Délégation	15
3.2.4	3. Événements et Planification	15
3.2.5	4. Base de Connaissances (Knowledge)	15
3.2.6	5. Gestion des Tags (Tags & API)	16

1 Présentation Générale

1.1 1. Qui sommes-nous?

Nous sommes une équipe de quatre étudiants en **Master 1 MIAGE** (Méthodes Informatiques Appliquées à la Gestion des Entreprises), parcours **APPRENTISSAGE**, promotion **2025** à l'**Université Paris Nanterre**.

Ce projet a été réalisé dans le cadre de notre cursus **DevOps**.

1.2 2. Quoi ? Pourquoi ?

L'objectif principal est de résoudre la "surcharge cognitive" liée au flux incessant d'emails en appliquant la méthode de productivité **Eisenhower** directement à la source.

Plutôt que de traiter les messages de manière chronologique, l'application force une hiérarchisation basée sur la valeur (importance) et l'urgence. L'innovation majeure réside dans l'utilisation d'une **IA locale** : contrairement aux solutions cloud classiques, le traitement sémantique s'effectue sur la machine de l'utilisateur, garantissant une **confidentialité absolue** des données sensibles et une indépendance vis-à-vis des connexions internet pour la partie analyse.

1.3 3. À propos du projet

Eisenflow est une solution intelligente de gestion du flux d'emails. Contrairement à une boîte de messagerie classique où l'utilisateur subit un flux chronologique et indifférencié, notre application :

- **Importe et Synchronise** : Récupération sécurisée de vos emails (via IMAP/Gmail) et création d'un cache local pour une navigation ultra-rapide et déconnectée.
- **Analyse** : Évaluation sémantique locale par IA du contenu de chaque message pour en déduire le contexte, le niveau d'urgence et l'importance.
- **Optimise et Catégorise** : Classement automatique (ou manuel) de vos emails selon la **Matrice d'Eisenhower** (DO, PLAN, DELEGATE, DELETE), avec la possibilité de créer des sous-tags personnalisés (ex: *DO · RH*, *DO · Technique*).
- **Délégation Assistée par IA** : Identification automatique du meilleur collaborateur dans votre équipe pour traiter une tâche spécifique, couplée à la pré-rédaction intelligente d'un brouillon de transfert.
- **Extraction de Connaissances** : Transformation des fils d'emails complexes en notes structurées persistantes (*Knowledge Base*) et génération de rapports ou mémos au format PDF.
- **Synchronisation Bidirectionnelle** : Exportation de vos décisions locales (tags et catégories) directement sur votre compte Gmail sous forme de labels, en un clic.
- **Gestion Intelligente de l'Intégrité** : Mécanisme de suppression en cascade garantissant que la modification ou la suppression d'un tag personnalisé réaffecte proprement les emails associés, sans aucune perte de données.

1.3.1 Public visé

- **Étudiants** : Pour jongler entre les mails administratifs de l'université, les rendus de projets urgents et les travaux de groupe. L'application permet d'isoler instantanément ce qui requiert une action immédiate (DO) des simples newsletters étudiantes (DELETE), réduisant ainsi le stress lié aux deadlines.

- **Développeurs** : Pour protéger les précieuses périodes de “Deep Work”. L’IA aide à séparer les alertes serveurs ou rapports de bugs critiques (DO) du reste du bruit numérique (notifications de pull requests, newsletters tech), tout en facilitant la délégation de tickets d’assistance au reste de l’équipe technique.
 - **Enseignants** : Pour gérer efficacement le flux constant et chronophage de questions d’étudiants, de directives administratives et de réunions pédagogiques. L’outil aide à prioriser les demandes urgentes, à planifier les corrections (PLAN) et à extraire rapidement les informations clés en mémos PDF pour les cours.
-

1.4 4. Analyse de Concurrence

Parmi les concurrents de notre application, on retrouve plusieurs applications largement utilisées qui proposent des fonctionnalités similaires. Par exemple, les boîtes mails classiques comme Gmail, Outlook ou AppleMail, qui permettent la gestion centralisée des mails, d’afficher des messages, de créer des dossiers ou encore de créer des règles manuelles. Contrairement à notre application, Gmail vise à faciliter la communication et l’organisation de l’information pour les utilisateurs mais elle ne permet pas le tri de mails automatique (pour l’instant).

Un autre type d’applications parmi les boîtes mails sont celles utilisant l’IA comme SuperHuman. L’IA peut résumer un mail et écrire une réponse. SuperHuman a donc quelques ressemblances par rapport à notre application (utilisation de l’IA ainsi que réponse qui ressemble à notre DELEGATE automatique) mais contrairement à nous, SuperHuman n’utilise pas une IA locale mais envoie les mails sur des serveurs en ligne. Si l’utilisateur souhaite de la confidentialité, ça serait une problématique.

Il y a aussi les applications de gestion de projet comme Jira qui proposent des tableaux Kanban, permettent de suivre les tâches et sont donc très utiles pour le travail en équipe. Notre application permet de suivre les tâches lues dans les mails, contrairement à Jira qui nécessite de mettre manuellement chaque tâche.

Note sur la confidentialité : Contrairement à SuperHuman, EisenFlow privilégie une **IA locale** pour protéger la vie privée des utilisateurs.

2 Ingénierie et Fonctionnalités

2.1 5. Éléments de gestion de projet et Architecture technique

Dans le cadre de ce projet, nous avons mis en place une organisation d'équipe stricte (réunions hebdomadaires, répartition claire des rôles : Backend, UI, QA) et une approche DevOps robuste :

- **Backend & IA Locale** : Le cœur de l'application repose sur **Java 21** et **Spring Boot 3.x**. L'extraction et la relève sécurisée des mails se font via **Jakarta Mail**. L'IA, garantissant 100% de confidentialité, est propulsée localement par **Ollama** (modèle *TinyLlama*).
 - **Collaboration & DevOps** : Nous utilisons la méthodologie Agile (Kanban) avec **GitHub** (Issues, PRs). L'intégration continue (CI/CD) est gérée par GitHub Actions pour automatiser la compilation et les tests. Le suivi strict de la qualité du code et des vulnérabilités est assuré par **SonarCloud**, **Jacoco**, et **JUnit 5**.
 - **Architecture & Patterns** : Afin de garantir un découplage strict et une flexibilité maximale (par exemple pour changer de modèle d'IA facilement), nous avons implémenté une architecture **MVC** soutenue par les design patterns **Strategy**, **Facade**, et **Registry**.
-

2.2 6. Diagramme de classes

2.2.1 6.1 But des features

F1 : Tri Intelligent (IA) * **Objectif** : Automatiser la priorisation selon la matrice d'Eisenhower. * **Implémentation** : Analyse sémantique via IA locale (Ollama) prenant en compte le Persona de l'utilisateur. Utilisation du Pattern *Strategy*.

F2 : Traitement et fusion de fichier markdown * **Objectif** : Centraliser et assembler dynamiquement plusieurs fichiers Markdown fragmentés en un document unique et structuré pour générer le livrable final. * **Implémentation** : Un script parcourt les dossiers cibles, nettoie les métadonnées (frontmatter) et concatène les fichiers .md dans un ordre défini avant de passer le résultat au compilateur Pandoc.

F3 : Tags Personnalisés * **Objectif** : Adapter l'outil aux processus spécifiques de l'utilisateur. * **Implémentation** : Création/suppression de tags sur-mesure validés par contrôles Regex et sauvegardés en JSON via la librairie *Jackson*.

F4 : Délégation Assistée * **Objectif** : Faciliter le transfert des tâches à l'expert adéquat. * **Implémentation** : Routage hybride (mots-clés + IA locale) pour générer un brouillon automatique. Utilisation du Pattern *Adapter*.

F5 : Préparation Réunions * **Objectif** : Fournir un contexte immédiat avant un rendez-vous. * **Implémentation** : Synthèse de l'historique des échanges par l'IA et génération d'un mémo au format PDF à l'aide de flux binaires et de la librairie *OpenPDF*.

F6 : Agenda & Rendez-vous * **Objectif** : Extraire et planifier automatiquement les événements. * **Implémentation** : Détection de dates et lieux via IA locale (NER - *Named Entity Recognition*) et parsing de dates par Regex pour proposer une insertion dans l'agenda.

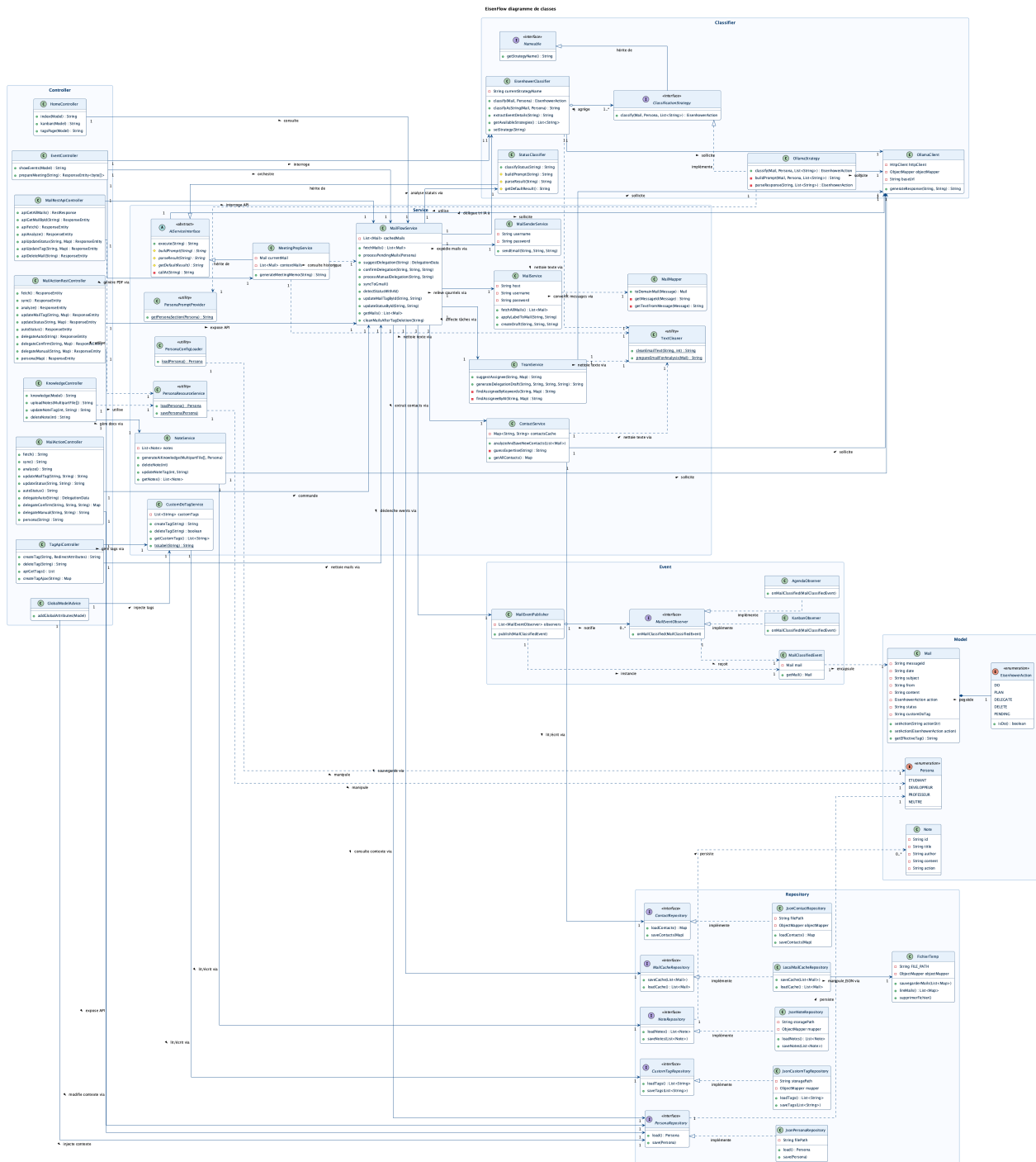
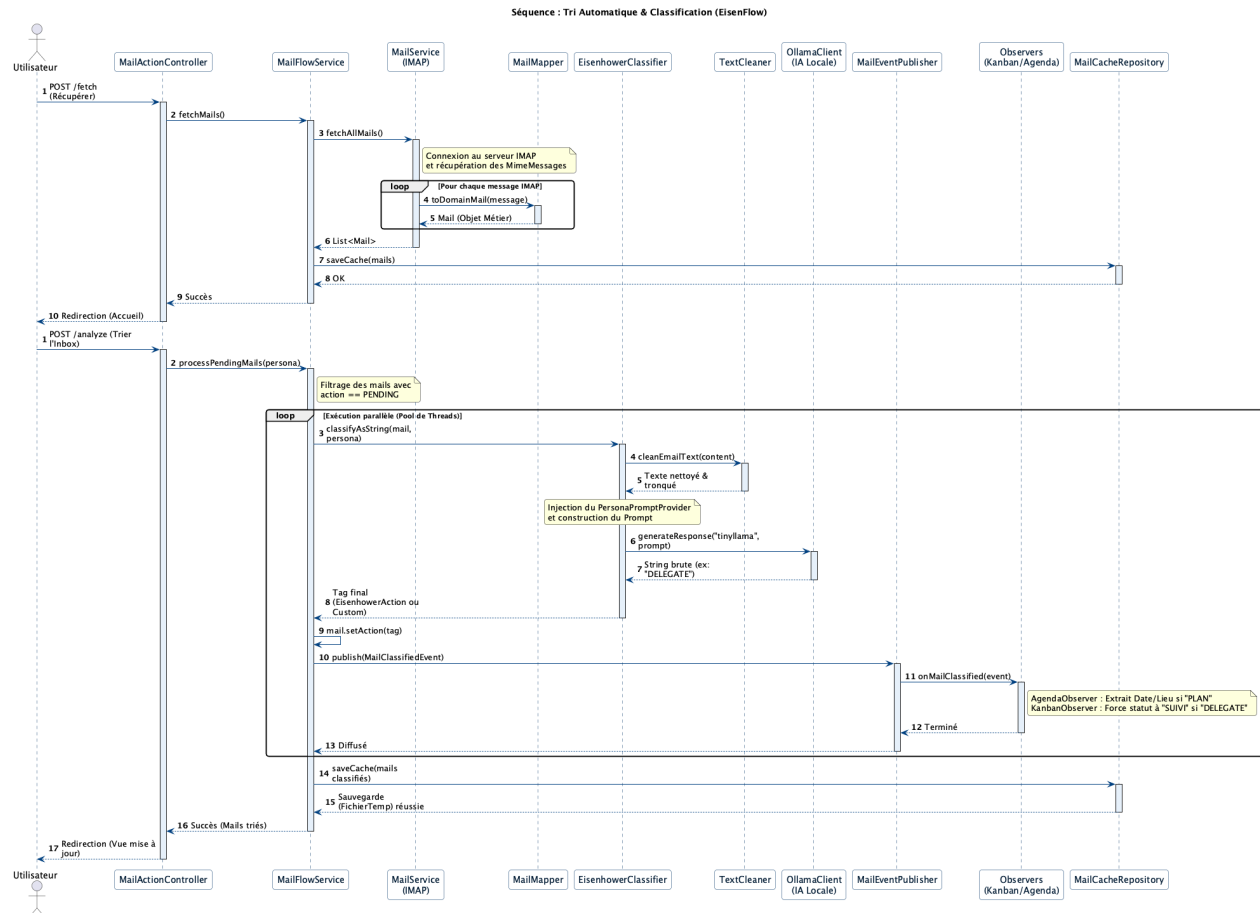
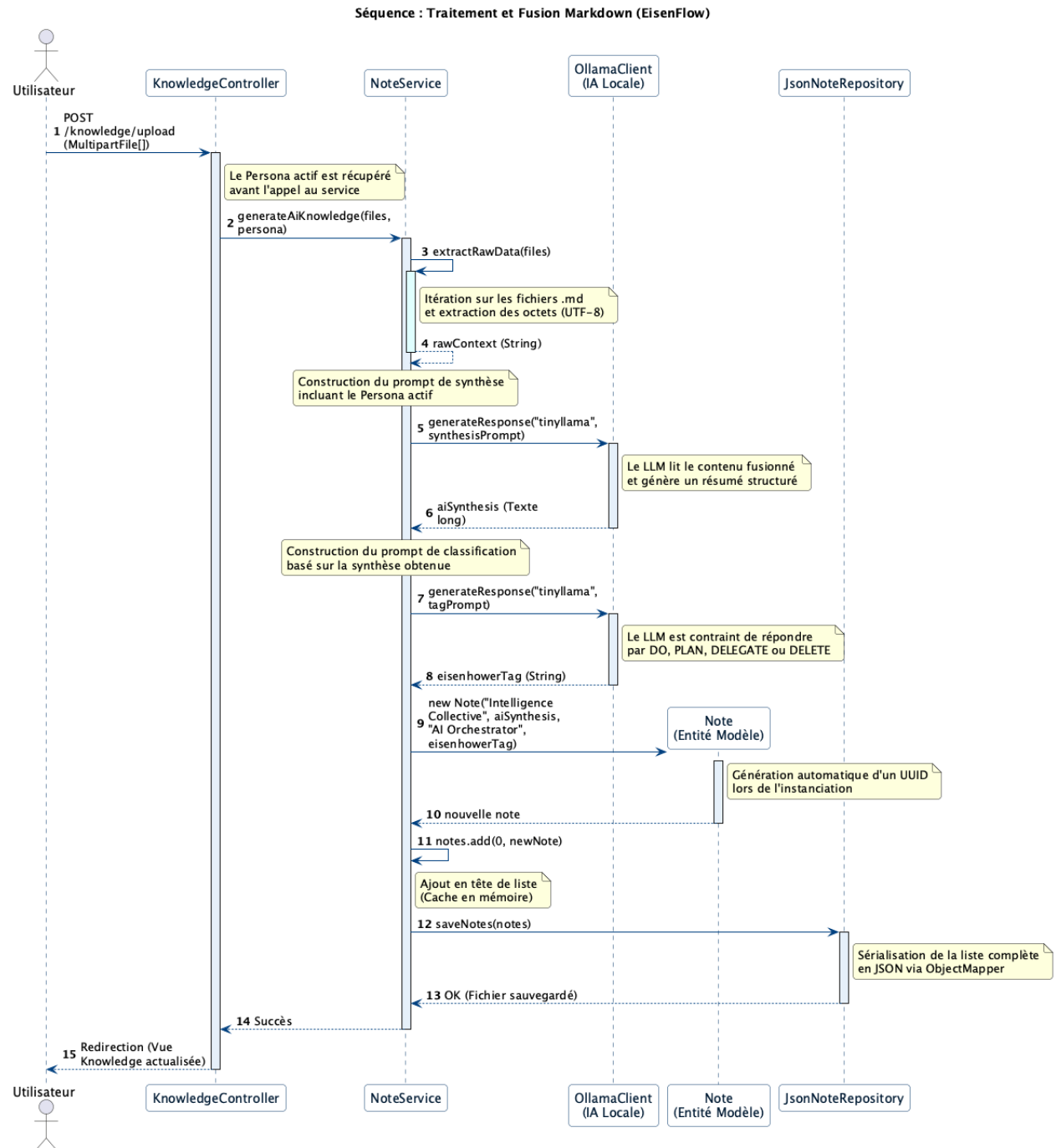


Figure 1: Diagramme de classes

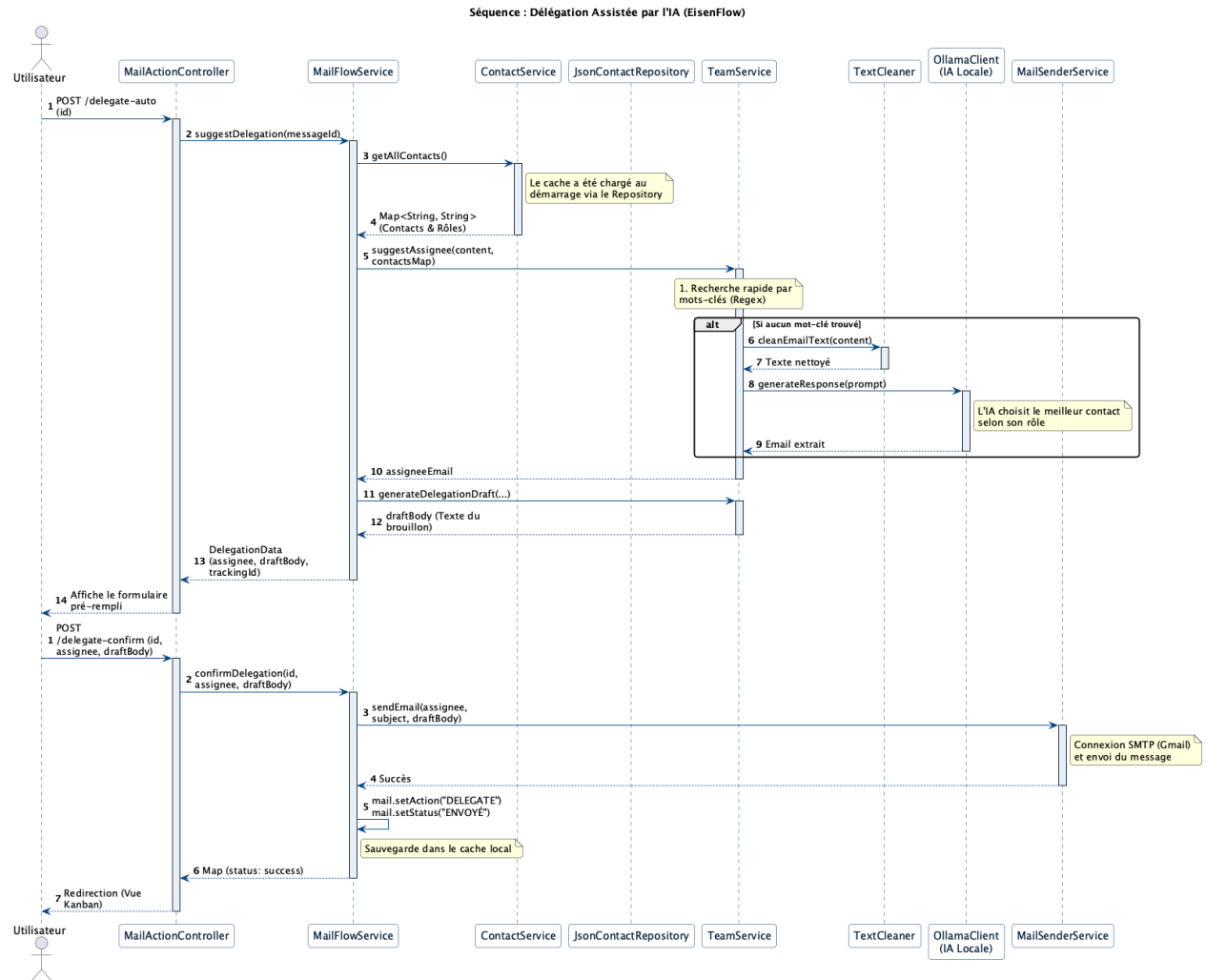
2.2.2 6.2 Scénarios

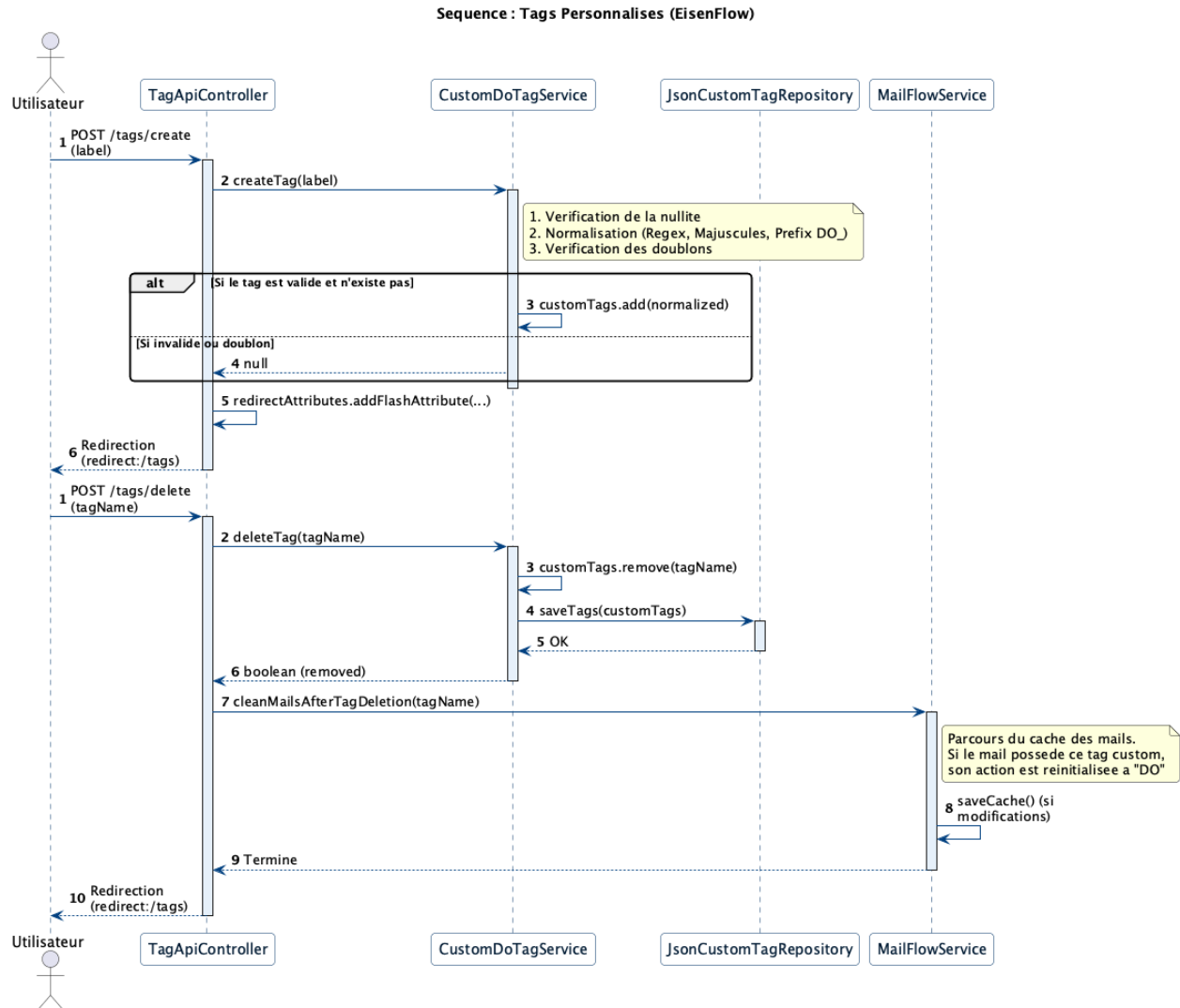
- **Scénario F1 & F2 (Synchronisation et Tri) :** L'utilisateur connecte son compte. Le backend relève les mails via Jakarta Mail. L'IA Ollama analyse sémantiquement chaque message en fonction du persona choisi et les répartit dans les colonnes du Kanban (DO, PLAN, etc.).





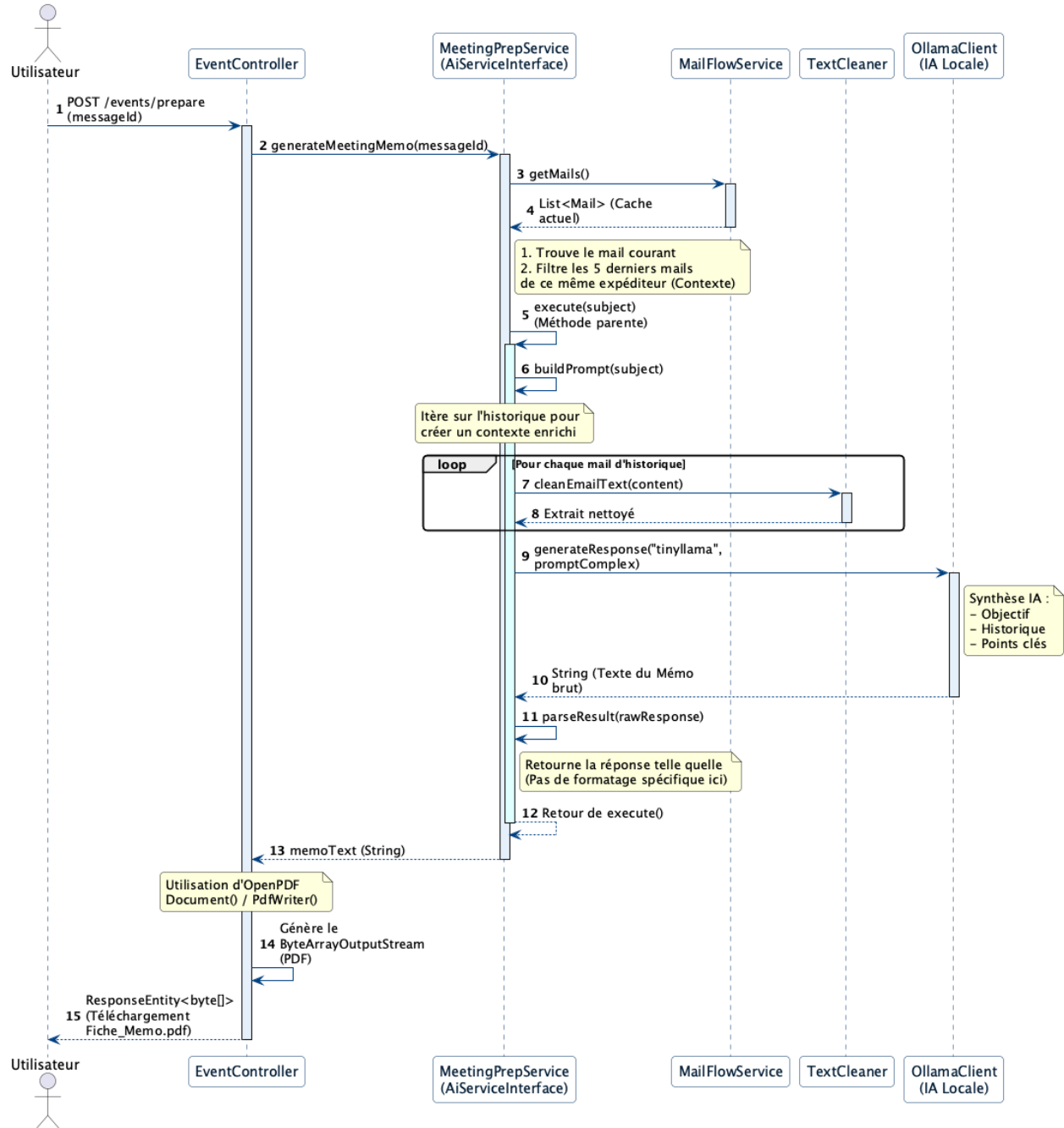
- **Scénario F3 & F4 (Délégation et TAG personnalisé) :** L'utilisateur reçoit une demande technique qu'il ne peut pas traiter. L'application utilise le routage hybride pour identifier l'expert concerné, prépare un brouillon de réponse expliquant le contexte, prêt à être envoyé.

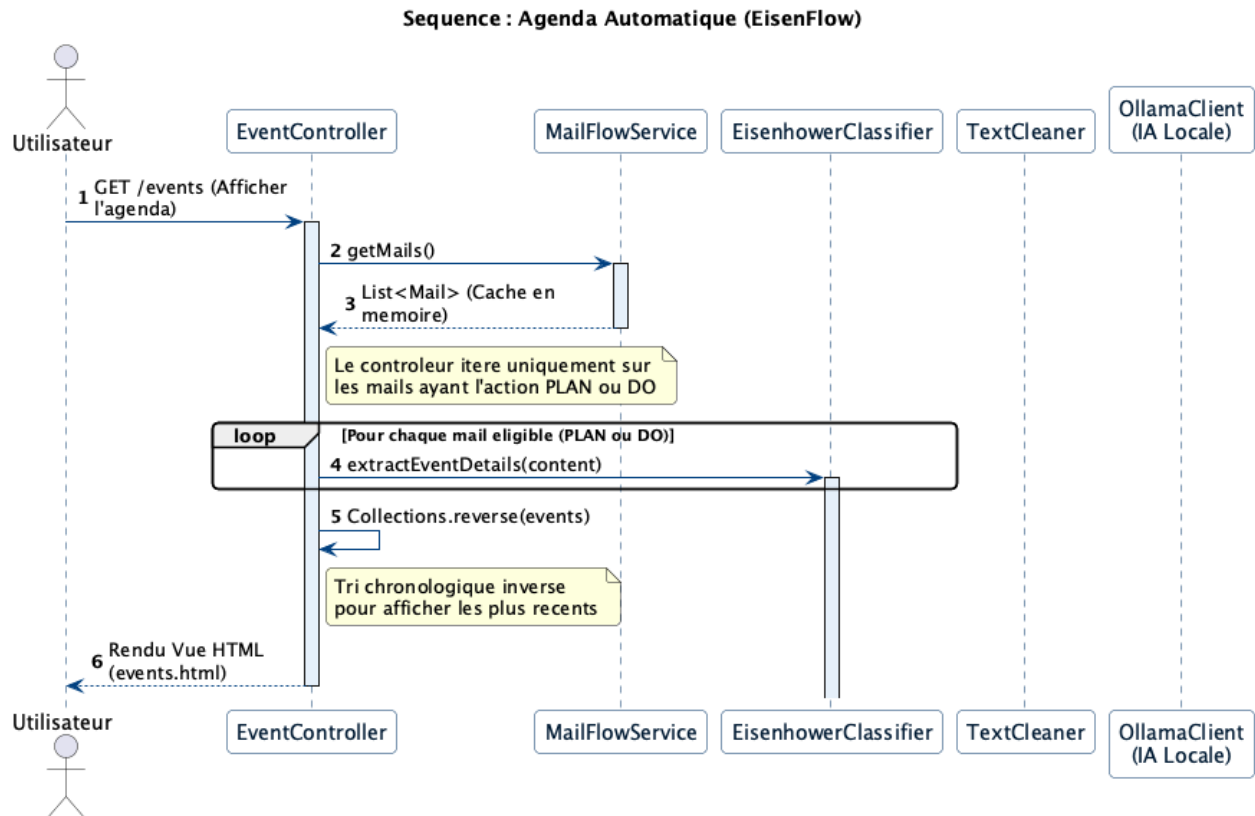




- **Scénario F5 & F6 (Réunion et Agenda)** : Un fil de mails contient une invitation pour un point de synchronisation vendredi à 10h. L'IA (via NER) extrait la date, propose d'ajouter le rendez-vous à l'agenda (F6) et génère simultanément un PDF via OpenPDF résumant les échanges précédents pour préparer la réunion (F5).

Séquence : Préparation de Réunion & Fiche Mémo (EisenFlow)





2.2.3 6.3 Wireframe et screenshots

F1 : Tri Intelligent (IA)

F2 : Traitement et fusion de fichier markdown

F3 : Tags Personnalisés

F4 : Délégation Assistée

F5 : Préparation Réunions

F6 : Agenda & Rendez-vous

2.3 7. Résumé des fonctionnalités

- **Synchronisation** : Relève et traitement sécurisés via les protocoles IMAP/SMTP pour vous accompagner vers l'objectif de l'*Inbox Zero*.
- **Tri IA (Local)** : Classification sémantique selon la matrice d'Eisenhower, propulsée localement par le modèle LLM *TinyLlama* pour garantir une confidentialité totale de vos données.
- **Tags personnalisés** : Système d'étiquetage dynamique et hautement flexible (gestion des "Sous-tags DO"), avec une persistance légère et rapide gérée via des fichiers JSON.
- **Délégation Intelligente** : Routage automatisé vers les collaborateurs les plus pertinents et génération de brouillons de transfert, assistés par l'IA et l'analyse de mots-clés.
- **Préparation & Synthèse** : Compilation et génération automatique de mémos récapitulatifs au format PDF pour faciliter la préparation de vos réunions ou le suivi de vos dossiers.
- **Planification (Agenda)** : Utilisation de la détection d'entités nommées (NER - *Named Entity Recognition*) pour identifier et extraire automatiquement les dates et événements de vos emails afin de les planifier.

3 8. Annexe API REST

3.1 Documentation de l'API Mails

URL de base : `/api/emails`

Format des requêtes et réponses : `application/json`

3.1.1 1. Actions Globales (Collection)

Ces routes agissent sur l'ensemble de la boîte mail ou sur la configuration globale.

Méthode	Route	Description	Corps (Body)
GET	<code>/</code>	Récupère tous les emails avec leurs liens d'actions (HATEOAS).	<i>Aucun</i>
POST	<code>/fetch</code>	Récupère les nouveaux emails depuis la source.	<i>Aucun</i>
POST	<code>/sync</code>	Synchronise avec Gmail puis récupère les nouveaux emails.	<i>Aucun</i>
POST	<code>/analyze</code>	Analyse et classe automatiquement les emails en attente (PENDING).	<i>Aucun</i>

Méthode	Route	Description	Corps (Body)
POST	<code>/auto-status</code>	Met à jour automatiquement les statuts grâce à l'IA.	<i>Aucun</i>
PUT	<code>/persona</code>	Met à jour le Persona de l'utilisateur.	<code>{"persona": "NOM_DU_PERSONA"}</code>

3.1.2 2. Actions Spécifiques (Ressource Unique)

Ces routes ciblent un email précis grâce à son identifiant (`{id}`).

Méthode	Route	Description	Corps (Body) Requis
GET	<code>/ {id}</code>	Récupère les détails d'un email spécifique.	<i>Aucun</i>
PUT	<code>/ {id}/status</code>	Modifie le statut d'un email (ex: TODO, DONE).	<code>{"status": "DONE"}</code>
PUT	<code>/ {id}/tag</code>	Ajoute ou modifie un tag sur l'email.	<code>{"tag": "URGENT"}</code>
DELETE	<code>/ {id}</code>	Marque l'email comme supprimé (Action = DELETE).	<i>Aucun</i>

3.1.3 3. Actions de Délégation

Routes dédiées au processus de délégation (Eisenhower : Delegate).

Méthode	Route	Description	Corps (Body) Requis
POST	<code>/ {id}/delegate-auto</code>	Génère une suggestion de délégation via l'IA.	<i>Aucun</i>
POST	<code>/ {id}/delegate-confirm</code>	Valide la délégation avec l'assigné et le brouillon.	<code>{"assignee": "email@...", "draftBody": "texte..."}</code>
POST	<code>/ {id}/delegate-manual</code>	Force une délégation manuelle sans brouillon IA.	<code>{"assignee": "email@..."}</code>

Note sur le format de réponse (HATEOAS) : > La plupart des requêtes GET (et certaines PUT) renvoient l'objet modifié accompagné d'un nœud `_links`. Ces liens indiquent au front-end quelles sont les prochaines actions possibles en fonction de l'état actuel de l'email.

3.2 Documentation des Contrôleurs (Application Web)

Ce document liste l'ensemble des routes exposées par les contrôleurs de l'application. La majorité des routes utilisent le pattern MVC classique (redirection vers des vues ou d'autres pages), avec quelques endpoints spécifiques renvoyant des données brutes (JSON ou PDF).

3.2.1 1. Vues Principales (Navigation)

Géré par `HomeController`. Ces routes permettent d'afficher les différentes pages de l'interface utilisateur.

Méthode	Route	Description	Paramètres
GET	<code>/</code>	Affiche la page d'accueil (Boîte de réception / Inbox).	<i>Aucun</i>
GET	<code>/kanban</code>	Affiche la vue Kanban (tâches en cours et finalisées).	<i>Aucun</i>
GET	<code>/tags</code>	Affiche la page de gestion des tags personnalisés.	<i>Aucun</i>

3.2.2 2. Actions sur les Mails

Géré par `MailActionController`. Ces routes traitent les formulaires d'action sur les emails et redirigent vers les vues.

Méthode	Route	Description	Paramètres attendus
POST	<code>/fetch</code>	Récupère les nouveaux mails depuis la source.	<i>Aucun</i>
POST	<code>/sync</code>	Synchronise avec Gmail et récupère les mails.	<i>Aucun</i>
POST	<code>/analyze</code>	Lance l'analyse IA des mails en attente.	<i>Aucun</i>
POST	<code>/auto-status</code>	Met à jour les statuts automatiquement avec l'IA.	<i>Aucun</i>

Méthode	Route	Description	Paramètres attendus
POST	<code>/update-mail-tag</code>	Met à jour le tag d'un mail spécifique.	messageId, tag
POST	<code>/update-status</code>	Met à jour le statut d'un mail.	messageId, status
POST	<code>/persona</code>	Modifie le Persona global de classification.	persona

3.2.3 Délégation

Méthode	Route	Description	Paramètres attendus	Retour
POST	<code>/delegate-auto</code>	Suggère une délégation via l'IA.	messageId	JSON
POST	<code>/delegate-confirm</code>	Confirme la délégation avec le brouillon.	messageId, assignee, draftBody	JSON
POST	<code>/delegate-manual</code>	Force une délégation sans brouillon IA.	messageId, assignee	Redirection

3.2.4 3. Événements et Planification

Géré par `EventController`. Permet de gérer les mails classifiés comme “À planifier”.

Méthode	Route	Description	Paramètres attendus	Retour
GET	<code>/events</code>	Affiche la liste des événements et tâches à faire.	<i>Aucun</i>	Vue HTML
POST	<code>/events/prepare</code>	Génère une fiche de préparation de réunion.	messageId	Fichier PDF

3.2.5 4. Base de Connaissances (Knowledge)

Géré par `KnowledgeController`. Permet de gérer les notes et fichiers de contexte pour l'IA.

Méthode	Route	Description	Paramètres attendus
GET	<code>/knowledge</code>	Affiche la base de connaissances.	<i>Aucun</i>
POST	<code>/knowledge/upload</code>	Uploade et génère des notes IA depuis des fichiers.	files (Multipart)
POST	<code>/update-note-tag</code>	Modifie le tag d'une note spécifique.	index, tag
POST	<code>/knowledge/delete</code>	Supprime une note de la base.	index

3.2.6 5. Gestion des Tags (Tags & API)

Géré par `TagApiController`. Gère la création et la suppression des étiquettes personnalisées.

Méthode	Route	Description	Paramètres attendus	Retour
GET	<code>/api/tags</code>	Récupère la liste de tous les tags.	<i>Aucun</i>	JSON
POST	<code>/tags/create</code>	Crée un nouveau tag via le formulaire classique.	label	Redirection
POST	<code>/tags/create-ajax</code>	Crée un nouveau tag de manière asynchrone (AJAX).	label	JSON
POST	<code>/tags/delete</code>	Supprime un tag et nettoie les mails associés.	tagName	Redirection