

Java Mail

EVRARD Nathan – DEBEFVE Theo

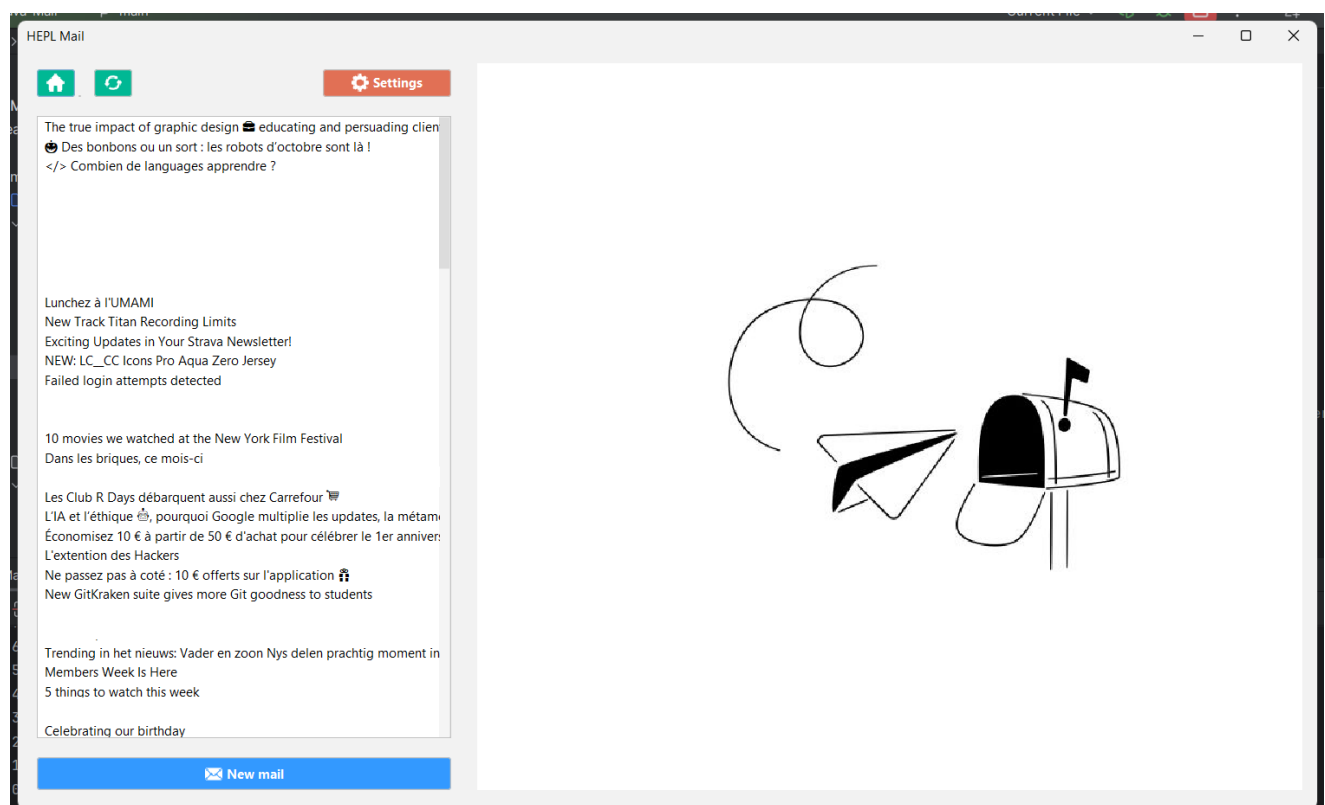
HEPL Seraing – 2023

Cette application Java est un agent de messagerie permettant la gestion des courriers électroniques entre un maraîcher et ses fournisseurs. Elle offre une interface conviviale pour envoyer et recevoir des e-mails, y compris des e-mails simples et composites avec des pièces jointes telles que des images gif, jpg, des fichiers de contrôle d'intégrité, et des fichiers texte. L'application utilise un thread de polling pour avertir l'utilisateur de l'arrivée de nouveaux messages dans un délai de 5 minutes. De plus, elle peut tracer les agents de messagerie par lesquels les messages sont passés.

Interface

Ecran Principale

Lorsque vous ouvrez l'application, vous êtes accueilli par la fenêtre principale illustrée ci-dessous :

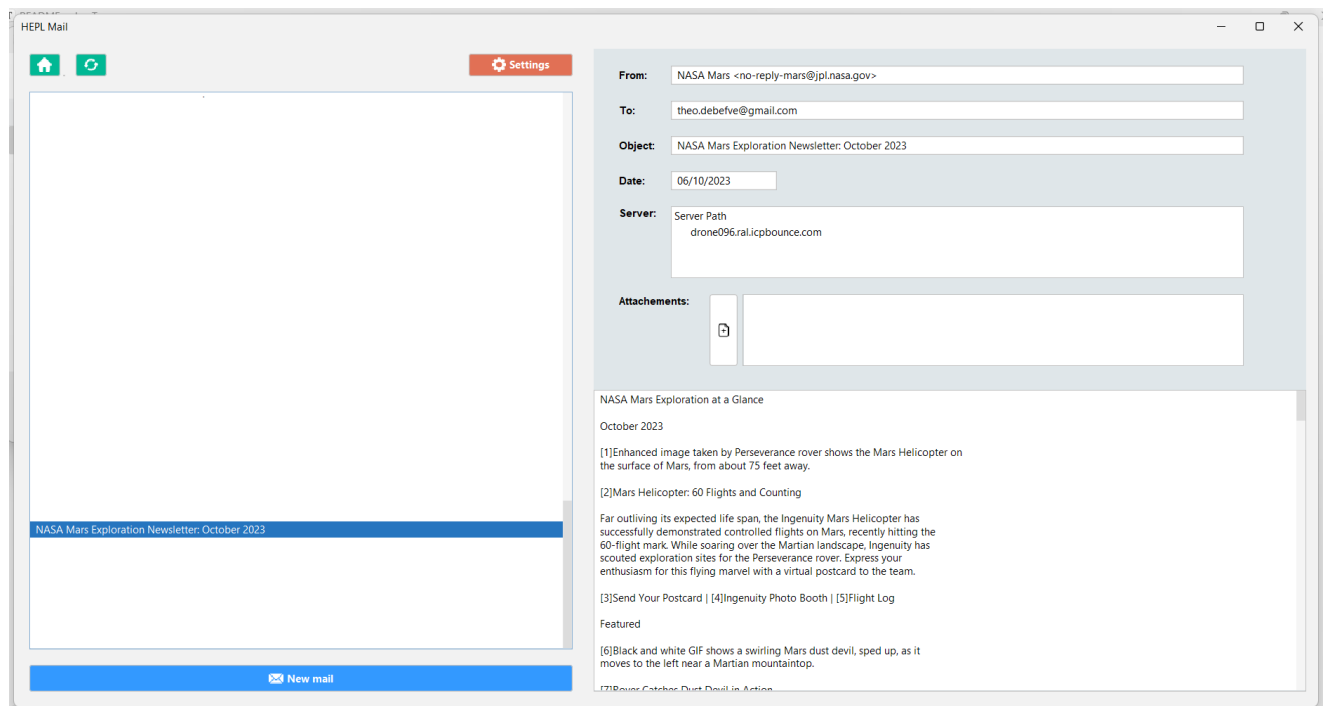


L'interface de l'application est divisée en deux parties distinctes :

- À gauche, consultez votre liste d'e-mails, triés du plus récent au plus ancien. Vous pouvez également retourner à l'accueil, charger les derniers e-mails reçus, accéder aux paramètres et rédiger un nouvel e-mail.
- À droite, consultez et rédigez vos e-mails en détail.

Lecture de Mail

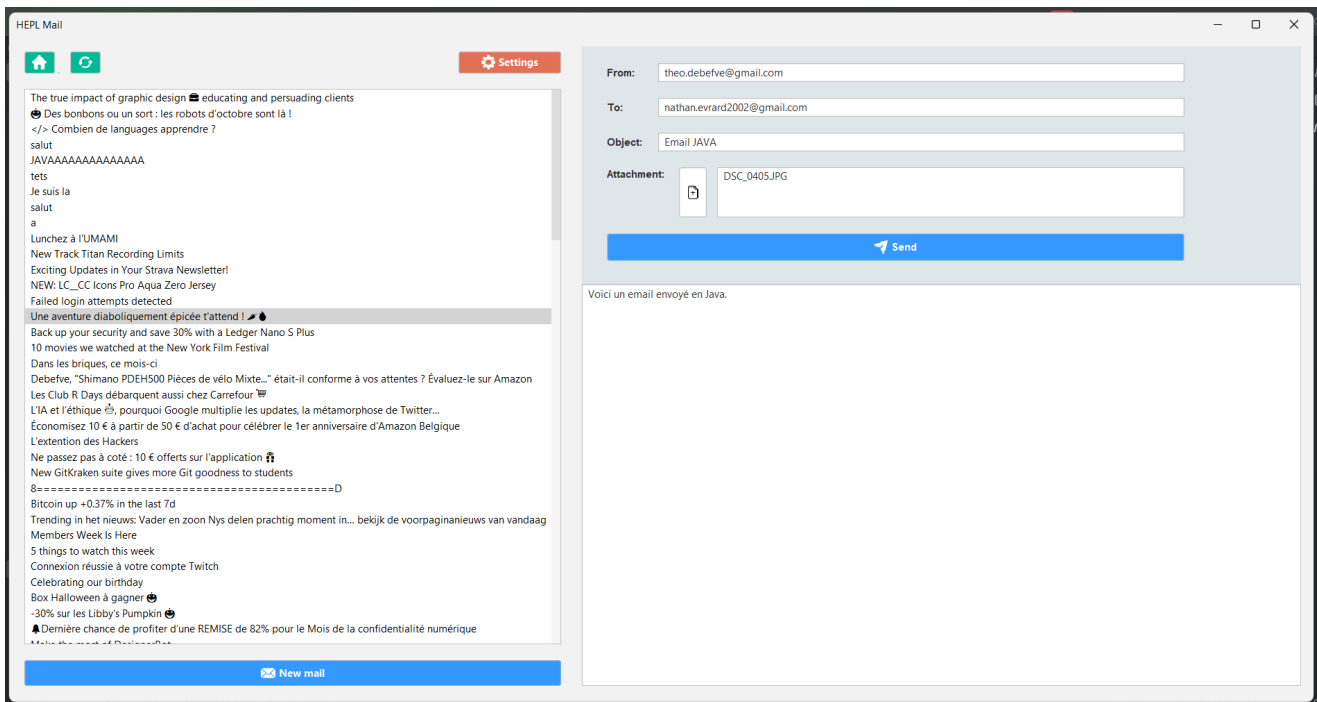
Lorsque vous cliquez sur l'un des e-mails de la liste, vous accédez à la fenêtre suivante :



Vous pouvez voir l'expéditeur, l'objet, la date d'envoi, les serveurs de passage, les pièces jointes téléchargeables, ainsi que le contenu du message.

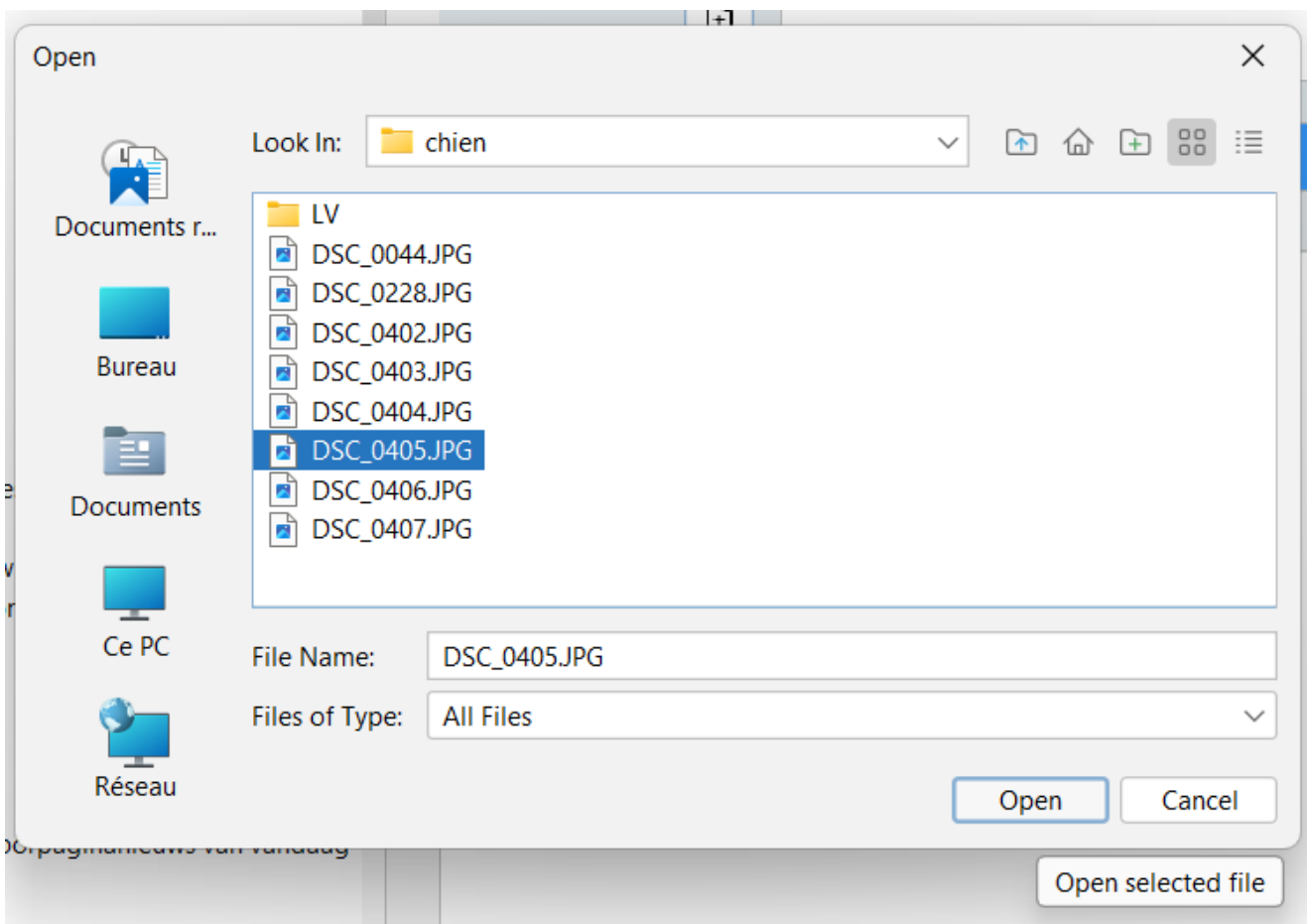
Lecture d'un Mail

Lorsque vous appuyez sur le bouton "New mail", vous ouvrez la fenêtre ci-dessous :



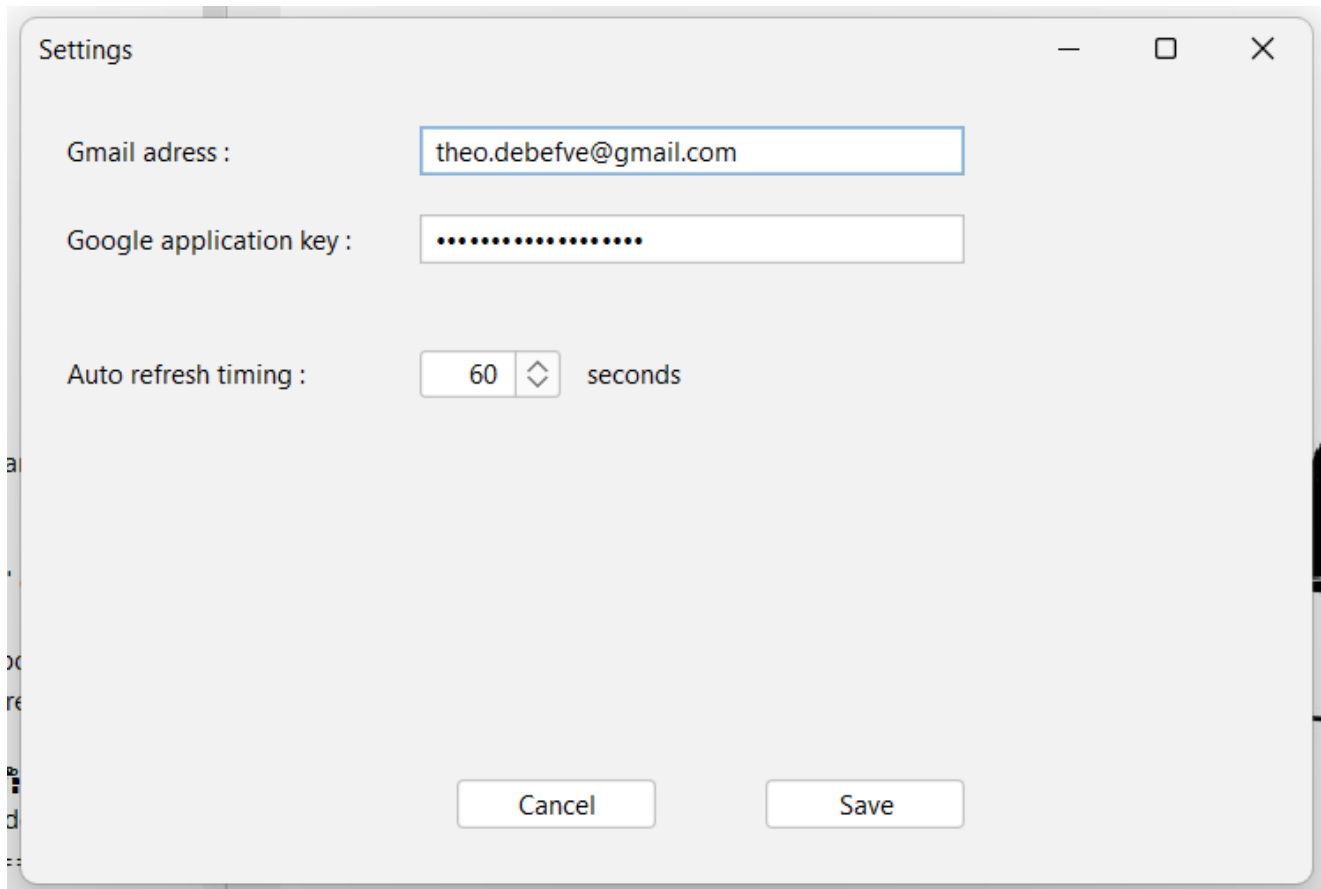
Votre propre adresse e-mail est automatiquement remplie dans le champ du destinataire. Vous avez juste besoin de saisir l'adresse du destinataire, l'objet du mail, les pièces jointes et le message. Enfin, cliquez sur "Envoyer" pour transmettre votre message.

Pour ajouter une pièce jointe, il vous suffit de cliquer sur le bouton dédié. Une boîte de dialogue apparaîtra, vous permettant de sélectionner le fichier à envoyer.



Paramètres

Vous avez la possibilité d'accéder à une fenêtre de paramètres où vous pouvez ajouter votre adresse e-mail (qui doit être une adresse Gmail), votre mot de passe d'application, ainsi que définir l'intervalle de temps en secondes pour la récupération des nouveaux e-mails.



The screenshot shows a 'Settings' dialog box with three input fields and two buttons. The first field, 'Gmail adress :', contains the email 'theo.debefve@gmail.com'. The second field, 'Google application key :', is filled with dots. The third field, 'Auto refresh timing :', has a spinner set to '60' and the unit 'seconds'. At the bottom are 'Cancel' and 'Save' buttons.

Gmail adress :	theo.debefve@gmail.com
Google application key :
Auto refresh timing :	60 seconds

Buttons: Cancel, Save

Classes

JMailSend

La classe `JMailSend` propose des fonctionnalités pour l'envoi d'e-mails à l'aide d'un serveur SMTP, en utilisant la bibliothèque JavaMail. Elle offre les fonctionnalités suivantes pour la composition et l'envoi d'e-mails :

- `SendMail`: Cette méthode permet d'envoyer un e-mail en utilisant un serveur SMTP spécifié. Elle prend en charge l'envoi d'e-mails simples ou composés, avec des pièces jointes, en utilisant le protocole SMTP. La méthode gère la création du message, l'ajout des destinataires, de l'objet et du contenu, ainsi que l'envoi effectif du message.

La classe `JMailSend` facilite le processus d'envoi d'e-mails en offrant des fonctionnalités pratiques pour composer et transmettre des messages à des destinataires spécifiés via un serveur SMTP.

JMailReceive

La classe `JMailReceive` propose des fonctionnalités pour récupérer et traiter les e-mails à partir d'un serveur POP3, en utilisant la bibliothèque JavaMail. Voici un aperçu des fonctionnalités principales :

- `getMails`: Cette méthode permet de récupérer une liste d'e-mails à partir de la boîte de réception en se connectant au serveur POP3 spécifié. Elle récupère des informations telles que l'expéditeur, le destinataire, la date, l'objet, les en-têtes, le contenu du message et les pièces jointes le cas échéant.

La classe `JMailReceive` facilite le processus de récupération et de traitement des e-mails en fournissant des fonctionnalités pratiques pour interagir avec le serveur de messagerie POP3 et traiter les différents composants des e-mails, y compris les pièces jointes.

ThreadMail

La classe `ThreadMail` étend la fonctionnalité de thread dans Java pour gérer la récupération périodique des e-mails dans l'application. Voici un aperçu des caractéristiques clés de cette classe :

- Utilisation d'un thread pour effectuer des opérations de récupération périodique des e-mails dans l'application.
- Intégration avec la classe `MainWindow` pour mettre à jour l'interface utilisateur avec les nouveaux e-mails reçus.
- Utilisation d'une liste de mails pour stocker les e-mails récupérés.
- Implémentation d'une boucle pour assurer une récupération continue des e-mails à des intervalles spécifiés.

La classe `ThreadMail` facilite la gestion asynchrone de la récupération des e-mails, garantissant une expérience utilisateur fluide tout en maintenant une surveillance régulière de la boîte de réception.

Utils

La classe `Utils` fournit des méthodes pratiques pour la manipulation de fichiers de propriétés dans votre application Java. Elle offre une solution simple pour la gestion de la configuration et des paramètres. Les fonctionnalités clés comprennent :

- `setProperty`: Cette méthode permet d'ajouter ou de mettre à jour une propriété spécifique avec une valeur donnée dans le fichier de propriétés.
- `getProperty`: Cette méthode permet de récupérer la valeur d'une propriété spécifique à partir du fichier de propriétés.

L'utilisation de la classe `Utils` simplifie la lecture et l'écriture de données de configuration, ce qui la rend particulièrement utile pour la gestion des paramètres et des préférences au sein de l'application Java.

Mail

La classe `Mail` permet la création et la gestion d'objets représentant des e-mails au sein de l'application. Voici un aperçu des fonctionnalités principales de cette classe :

- Création d'objets `Mail` avec des attributs tels que l'objet, l'expéditeur, le destinataire, la date, le message, les en-têtes et les pièces jointes.
- Gestion des attributs tels que la définition et la récupération de l'objet, de l'expéditeur, du destinataire et de la date de l'e-mail.
- Traitement des dates sous forme de chaînes de caractères et d'objets `Date`, en utilisant le format "dd/MM/yyyy".
- Gestion du message, des en-têtes et des pièces jointes associés à l'e-mail.

La classe `Mail` facilite la manipulation des données d'e-mails, offrant des méthodes pratiques pour accéder et gérer les informations relatives aux e-mails au sein de l'application.

Attachement

La classe `Attachment` offre des fonctionnalités pour gérer les pièces jointes dans le contexte de la messagerie électronique. Elle propose des méthodes permettant de télécharger les pièces jointes, de définir et obtenir des informations sur les noms de fichiers et les chemins d'accès. Voici un aperçu des fonctionnalités principales :

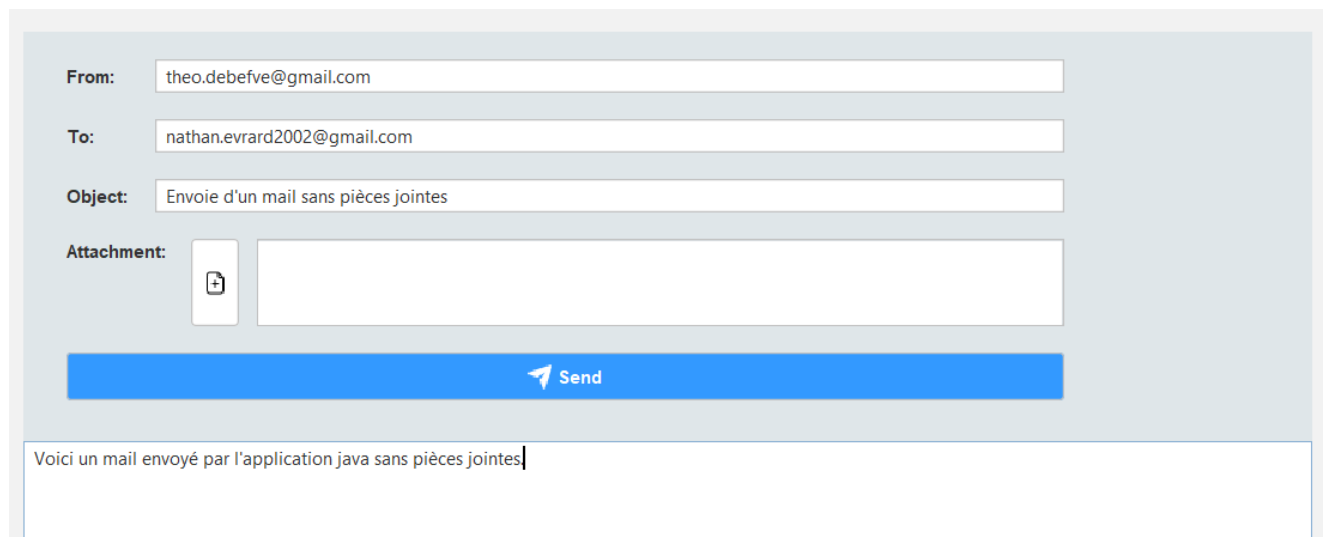
- `download`: Cette méthode permet de télécharger la pièce jointe et de l'enregistrer localement à l'aide de son flux d'entrée.

- `getNom` et `setNom`: Ces méthodes permettent respectivement d'obtenir et de définir le nom de la pièce jointe.
- `getPath` et `setPath`: Ces méthodes permettent respectivement d'obtenir et de définir le chemin d'accès de la pièce jointe.
- `getPart` et `setPart`: Ces méthodes permettent respectivement d'obtenir et de définir la partie (Part) de la pièce jointe.

La classe `Attachment` simplifie le processus de gestion des pièces jointes en offrant des fonctionnalités pratiques pour télécharger et manipuler ces fichiers au sein de l'application.

Exemple

Envoie d'un mail sans pièces jointe

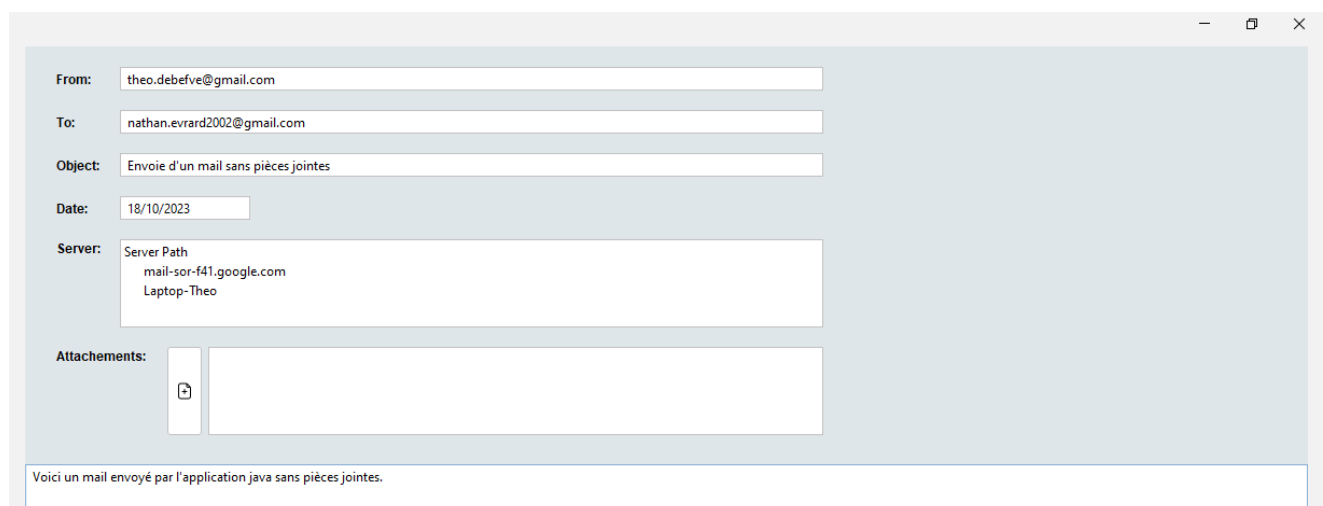


The screenshot shows a Java Mail application window with a light blue background. It contains several input fields for email details:

- From:** `theo.debefve@gmail.com`
- To:** `nathan.evrard2002@gmail.com`
- Object:** `Envoie d'un mail sans pièces jointes`
- Attachment:** A section with a small icon of a document with a plus sign and an empty rectangular box for adding attachments.

 Below these fields is a prominent blue button with a white paper plane icon and the text `Send`. At the bottom of the window, there is a text area containing the message: `Voici un mail envoyé par l'application java sans pièces jointes`.

Réception d'un mail sans pièce jointe



The screenshot shows a Java Mail application window, similar to the one above but for receiving mail. It includes the same header fields:

- From:** `theo.debefve@gmail.com`
- To:** `nathan.evrard2002@gmail.com`
- Object:** `Envoie d'un mail sans pièces jointes`
- Date:** `18/10/2023`
- Server:** A section containing the text `Server Path`, `mail-sor-f41.google.com`, and `Laptop-Theo`.
- Attachments:** A section with a small icon of a document with a plus sign and an empty rectangular box for displaying attachments.

 The bottom text area contains the same message: `Voici un mail envoyé par l'application java sans pièces jointes`. The window has standard OS window controls (minimize, maximize, close) in the top right corner.

Envoie d'un mail avec pièces jointe

From:

theo.debefve@gmail.com


To:

nathan.evrard2002@gmail.com


Object:

Envoie d'un mail avec pièces jointes

Attachment:



DSC_0405.JPG
eno-labo-ecom-prog-reseaux-2023-2024.pdf

 Send

Voici un mail envoyé par l'application java avec deux pièces jointes.

1 document pdf
1 image .jpg

Réception d'un mail avec pièce jointe

From:

theo.debefve@gmail.com

To:

nathan.evrard2002@gmail.com

Object:

Envoie d'un mail avec pièces jointes


Date:

18/10/2023

Server:

Server Path
mail-sor-f41.google.com
Laptop-Theo

Attachments:



DSC_0405.JPG
eno-labo-ecom-prog-reseaux-2023-2024.pdf

Voici un mail envoyé par l'application java avec deux pièces jointes.

1 document pdf
1 image .jpg