Introduction à JavaFX

A travers la création d'une application « TodoList »

Sommaire

Envoyer un e-mail sécurisé avec un code de vérification

Nous allons utiliser **JavaMail** pour envoyer un e-mail contenant un code de réinitialisation de mot de passe à un utilisateur.

Installation de JavaMail avec Maven

Sur le site de Maven, recherchez la dépendance suivante :

- La dépendance javax.mail de com.sun.mail

Création du service d'envoi d'e-mail

- Créer un nouveau package nommé « service »
- Créer une classe nommée « EmailService »
- Placez y le code suivant :

```
import java.util.Properties;
import javax.mail.*;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class EmailService {
    private static final String EMAIL_SENDER = "votre-email@gmail.com";
    private static final String PASSWORD = "votre-mot-de-passe";

    public static void envoyerEmail(String destinataire, String sujet, String messageTexte) {
        Properties properties = new Properties();
        properties.put("mail.smtp.auth", "true");
        properties.put("mail.smtp.starttls.enable", "true");
        properties.put("mail.smtp.host", "smtp.gmail.com");
        properties.put("mail.smtp.port", "587");

        Session session = Session.getInstance(properties, new Authenticator() {
            @Override
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(EMAIL_SENDER, PASSWORD);
            }
        });

        try {
            Message message = new MimeMessage(session);
            message.setFrom(new InternetAddress(EMAIL_SENDER));
            message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(destinataire));
            message.setSubject(sujet);
            message.setText(messageTexte);

            Transport.send(message);
            System.out.println("E-mail envoyé avec succès !");
        } catch (MessagingException e) {
            e.printStackTrace();
        }
    }
}
```

Pourquoi utiliser Properties et Authenticator ?

- Permet de configurer l'envoi via SMTP.
- Sécurise l'accès au compte expéditeur.

La configuration actuelle est liée à un compte gmail mais vous pouvez créer n'importe quel courriel et le paramétrer différemment. Il faudra simplement changer les serveuses liées aux fournisseur courriel (le smtp par exemple)

Génération d'un code de réinitialisation aléatoire

Ajoutez cette méthode à EmailService.java :

```
import java.security.SecureRandom;

public class EmailService {
    private static final SecureRandom random = new SecureRandom();

    public static String genererCode() {
        int code = 100000 + random.nextInt(900000);
        return String.valueOf(code);
    }
}
```

Pourquoi SecureRandom ?

- Assure une meilleure sécurité qu'un simple Random.
- Génère un code à 6 chiffres difficile à deviner.

Envoi du code et intégration dans l'interface utilisateur

Dans src/main/java/appli/accueil, créez **MotDePasseOublieController.java** :

```
import javafx.fxml.FXML;
import javafx.scene.control.TextField;
import util.EmailService;

public class MotDePasseOublieController {
    @FXML
    private TextField emailField;


    @FXML
    private void envoyerCode() {
        String email = emailField.getText();
        if (email.isEmpty()) {
            System.out.println("Veuillez entrer une adresse e-mail.");
            return;
        }

        String code = EmailService.genererCode();
        EmailService.envoyerEmail(email, "Réinitialisation de mot de passe", "Votre code de
réinitialisation est : " + code);
        System.out.println("Code envoyé à : " + email);
    }
}
```

Explication :

- Récupère l'adresse e-mail entrée par l'utilisateur.
- Génère un code de réinitialisation.
- Envoie l'e-mail contenant le code.

Il est nécessaire de complétez la méthode car, avant d'envoyer l'email, il faudrait trouver l'utilisateur liée à l'email.

- Si l'utilisateur existe, on envoie l'email
- Sinon, non

Finalisation du process

Sur votre interface :

- Complétez votre interface en ajoutant un passwordField
  - Ajoutez un fx:id au field
- Et un bouton de vérification afin d'être rediriger vers la page de changement de mot de passe (en envoyant l'utilisateur sélectionnée)
  - Si le code est correct, on le redirige
  - Sinon, on l'invite à vérifier le code saisi

Créer une nouvelle page avec :

- PasswordField : Nouveau mot de passe
- PasswordField : Confirmation du mot de passe
- Button : vérification des mots de passe
  - Vérifie si les mots de passe coïncident
  - Modifie le mot de passe utilisateur en bdd
  - Redirige l'utilisateur vers la page de connexion

Tester la connexion

Tester le changement de mot de passe en essayant de se connecter avec l'utilisateur dont le mot de passe a été réinitialisé !