

Guide d'utilisation de LGoodDatePicker dans un projet Swing

Ce guide explique comment intégrer le composant **LGoodDatePicker** dans un projet Java Swing pour créer un champ de date avec pop-up calendrier, stylé et moderne, compatible avec FlatLaf.

1. Télécharger le JAR

Télécharge le dernier JAR depuis le GitHub officiel : - <https://github.com/LGoodDatePicker/LGoodDatePicker> - Exemple : `LGoodDatePicker-11.2.1.jar`

Place le fichier `.jar` dans un dossier `libs/` de ton projet :

```
projet/
├─ src/
└─ libs/
    └─ LGoodDatePicker-11.2.1.jar
```

2. Ajouter le JAR au classpath

Via IDE

- **Eclipse** : clic droit sur le projet → Properties → Java Build Path → Libraries → Add External JARs → sélectionne le JAR.
- **IntelliJ IDEA** : File → Project Structure → Modules → Dependencies → + → JARs or directories → sélectionne le JAR.

Via ligne de commande

```
javac -cp "libs/LGoodDatePicker-11.2.1.jar:." src/**/*.java
java -cp "libs/LGoodDatePicker-11.2.1.jar:." ton.package.Main
```

3. Importer les classes dans ton code

```
import com.github.lgooddatepicker.components.DatePicker;
import com.github.lgooddatepicker.components.DatePickerSettings;
import java.time.LocalDate;
import java.sql.Date;
```

4. Créer un composant DatePicker dans un JPanel

```
// Créer les paramètres pour personnaliser le DatePicker
DatePickerSettings dateSettings = new DatePickerSettings();
dateSettings.setFormatForDatesCommonEra("dd/MM/yyyy"); // format affiché

// Créer le DatePicker
DatePicker datePicker = new DatePicker(dateSettings);

// Ajouter au panel
JPanel panel = new JPanel();
panel.add(datePicker);
```

5. Récupérer la date sélectionnée

```
// Récupérer la date en LocalDate
LocalDate localDate = datePicker.getDate();

// Convertir en java.sql.Date si nécessaire pour la base
java.sql.Date sqlDate = java.sql.Date.valueOf(localDate);

System.out.println("Date sélectionnée (LocalDate): " + localDate);
System.out.println("Date SQL: " + sqlDate);
```

6. Intégration avec FlatLaf (optionnel mais conseillé)

```
import com.formdev.flatlaf.FlatLightLaf;

public static void main(String[] args) {
    try {
        UIManager.setLookAndFeel(new FlatLightLaf());
    } catch (Exception e) {
        e.printStackTrace();
    }

    // Crée ton JFrame et panel normalement
}
```

Le `DatePicker` s'adapte automatiquement au style FlatLaf.

7. Notes importantes

- LGoodDatePicker est compatible **Java 8+**.
 - Tu peux ajouter un `TimePicker` pour sélectionner l'heure si besoin.
 - Tu peux personnaliser le style, la couleur des weekends, des jours fériés, etc. via `DatePickerSettings`.
-

Résumé : LGoodDatePicker est moderne, facile à intégrer, supporte LocalDate et SQL Date, et s'intègre parfaitement avec FlatLaf. C'est le composant recommandé pour des formulaires Swing avec saisie de dates.