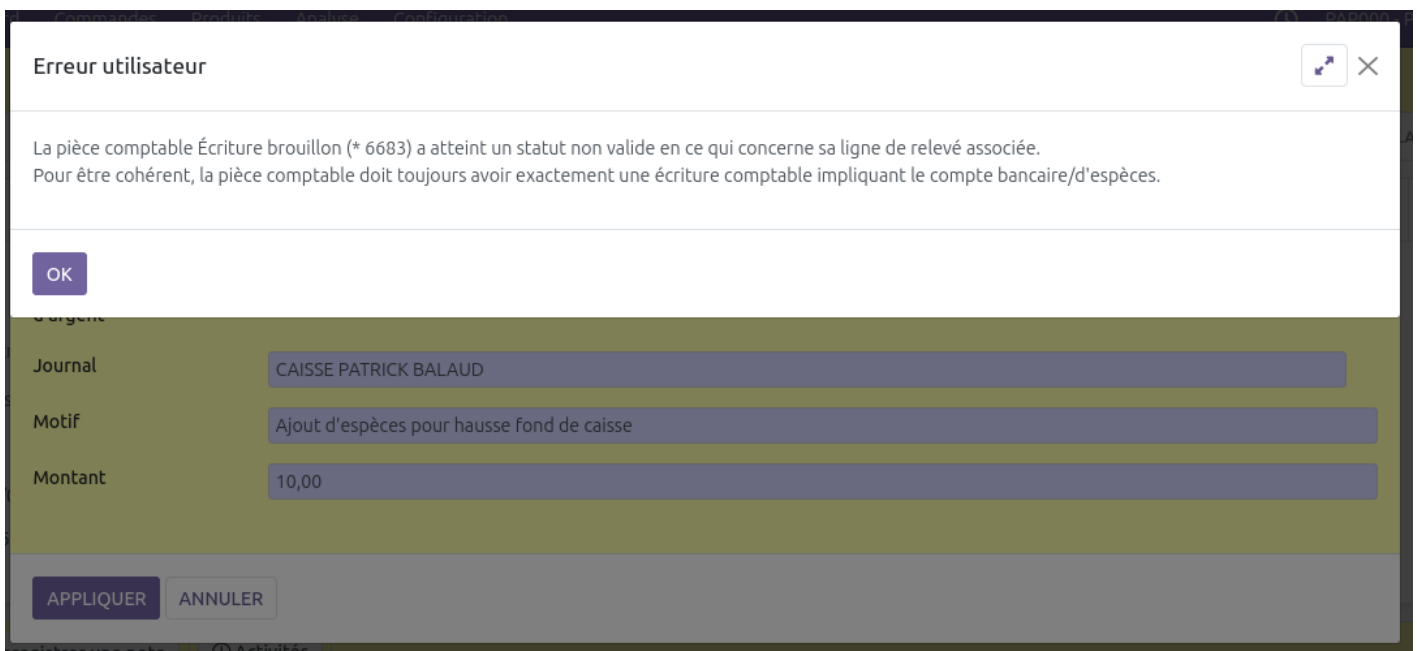


Pour plus de contenu, connectez-vous !

Erreur à la validation d'une entrée / sortie d'argent à la caisse



Erreur utilisateur

La pièce comptable Écriture brouillon (* 6683) a atteint un statut non valide en ce qui concerne sa ligne de relevé associée.
Pour être cohérent, la pièce comptable doit toujours avoir exactement une écriture comptable impliquant le compte bancaire/d'espèces.

OK

Journal CAISSE PATRICK BALAUD

Motif Ajout d'espèces pour hausse fond de caisse

Montant 10,00

APPLIQUER ANNULER

Transcription de l'Erreur utilisateur

La pièce comptable Écriture brouillon (* 6683) a atteint un statut non valide en ce qui concerne sa ligne de relevé associée.

Pour être cohérent, la pièce comptable doit toujours avoir exactement une écriture comptable impliquant le compte bancaire/d'espèces.

Raison

C'est lié au paramétrage comptable du mouvements de caisse.

Il faut **pas** que les comptes associés aux mouvements de caisse (revenus/dépenses) soient les mêmes que le journal du mouvement (journal de caisse dans l'exemple), sinon ça bloque à la création de la pièce comptable car ça va mettre le même compte en débit / crédit

The image shows two screenshots of a software interface for setting up a cash movement journal. The left screenshot shows the 'Dépôt d'espèces' form with fields for 'Compte de revenus' (53000000) and 'Compte de dépenses' (53000000). The right screenshot shows the 'CAISSE CAGETTE DU SILLON' journal configuration with 'Type' set to 'Espèces' and 'Compte d'attente' set to '51200100 Compte d'attente de la banque'.

Par exemple, pour le dépôt d'espèces, il faut donc utiliser le compte d'attente **51200100** **Compte d'attente de la banque** ou équivalent.

explications techniques

pour note technique

c'est la fonction `_synchronize_from_moves` qui appelle `_seek_for_lines` dans `account_bank_statement_line.py`

ça prend les deux lignes du statement, ça check dans quelles catégories classés les comptes, et si le compte est similaire à celui du journal, ça met dans `liquidity_lines`, sauf que dans

`_synchronize_from_moves` ça check qu'il y ait qu'une ligne lié à la liquidité `if len(liquidity_lines) != 1:`

```
def seek_for_lines(self):
    """ Helper used to dispatch the journal items between:
    - The lines using the liquidity account.
    - The lines using the suspense account.
    - The lines being not in one of the two previous categories.
    :return: (liquidity_lines, suspense_lines, other_lines)
    """
    liquidity_lines = self.env['account.move.line']
    suspense_lines = self.env['account.move.line']
    other_lines = self.env['account.move.line']

    for line in self.move_id.line_ids:
        if line.account_id == self.journal_id.default_account_id:
            liquidity_lines += line
        elif line.account_id == self.journal_id.suspense_account_id:
            suspense_lines += line
        else:
            other_lines += line
    if not liquidity_lines:
        liquidity_lines = self.move_id.line_ids.filtered(lambda l: l.account_
        other_lines -= liquidity_lines
    return liquidity_lines, suspense_lines, other_lines

# SYNCHRONIZATION account bank statement line <-> account move
```

Revision #1

Created 26 June 2026 14:06:58 by Quentin Dupont

Updated 26 June 2026 14:13:38 by Quentin Dupont