

Lingo - Development #76130

Facturation: pouvoir déclarer un règlement sur une facture

03 avril 2023 09:43 - Lauréline Guérin

| | | | |
|------------------------|------------------|----------------------|---------------|
| Statut: | Fermé | Début: | 03 avril 2023 |
| Priorité: | Normal | Echéance: | |
| Assigné à: | Lauréline Guérin | % réalisé: | 0% |
| Catégorie: | | Temps estimé: | 0:00 heure |
| Version cible: | | Planning: | Non |
| Patch proposed: | Non | | |

Description

Ajouter des modèles pour gérer les règlements:

model Payment:

- FK transaction
- FK invoice
- date règlement
- montant
- type de règlement (CB, espèce, chèque, virement, prélèvement, ...)

model Transaction:

- FK Regie
- date transaction
- montant total
- type de règlement

Endpoint pour ajouter un règlement:

- invoice ids (de la même régie !, et pour un même payeur !)
- montant
- type de règlement
- si plusieurs invoices, on commence par solder la facture la plus ancienne
- vérifier que le montant ne dépasse pas le reste à payer

Révisions associées

Révision 8411c316 - 21 avril 2023 17:02 - Lauréline Guérin

invoicing: models for payment (#76130)

Révision d37d31ae - 21 avril 2023 17:02 - Lauréline Guérin

invoicing: add uuid on Invoice model (#76130)

Révision 73d53913 - 21 avril 2023 17:07 - Lauréline Guérin

api: endpoint to add payments (#76130)

Historique

#1 - 03 avril 2023 17:49 - Robot Gitea

- Statut changé de Nouveau à En cours
- Assigné à mis à Lauréline Guérin

Lauréline Guérin (lguerin) a ouvert une pull request sur Gitea concernant cette demande :

- URL : <https://git.entrouvert.org/entrouvert/lingo/pulls/49>
- Titre : WIP: Facturation: pouvoir déclarer un règlement sur une facture (#76130)
- Modifications : <https://git.entrouvert.org/entrouvert/lingo/pulls/49/files>

#2 - 04 avril 2023 10:04 - Lauréline Guérin

- Description mis à jour

#3 - 04 avril 2023 16:00 - Robot Gitea

- Statut changé de En cours à Solution proposée

#4 - 18 avril 2023 16:51 - Benjamin Dauvergne

Quel est la distinction qui est faite ici entre transaction et paiement ? Lingo ne gérant pas le paiement lui même mais ne recevant que les évènements finaux payé/pas payé, d'où vient la différence ?

#6 - 18 avril 2023 16:56 - Lauréline Guérin

Ici ce sont juste des endpoints pour pouvoir déclarer un paiement effectué en guichet: le parent a déposé 10 cesu de 10 euros chacun, mais il a 3 factures à régler d'un montant de 40€ chacune: ce dépôt permet de solder complètement les 2 premières factures et une partie de la 3e.

transaction = 100€ en cesu

3 paiements:

paiement 1 facture 1 40€

paiement 2 facture 2 40€

paiement 3 facture 3 20€

#7 - 18 avril 2023 17:46 - Benjamin Dauvergne

Ok bien compris, on retombe sur un concept habituel de transaction comptable entre un compte client et un compte de facturation, une transaction ventilant un paiement de manière automatique en prenant pour chaque facture le reste à payer.

Je pense qu'il faudrait se rapprocher d'un modèle comptable, ça commencerait par avoir un modèle pour le payeur plutôt que de le référencer implicitement via Invoice.payer_external_id. C'est pas que ce soit mal de faire ça mais je trouve bien de toujours matérialiser les objets métiers.

Ensuite je ne matérialiserais pas le solde des factures sur les factures elle même c'est quelque chose qu'on peut recalculer à la volée sans aucune erreur donc pas de Invoice.paid_amount/remaining_amount.

Ensuite j'aurai un objet Payment pour le paiement en lui même qui référencerait le payeur (Payer) et la régie/compte de facturation concerné et des InvoicePayment pour la ventilation et aurait un identifiant de paiement (payment_reference) venant de combo/lingo, ou n'importe quoi qui nous déclare un paiement (formulaire pour déclarer un paiement en espèce/chèque au guichet) pour éviter de compter un paiement deux fois, lingo remontant pas mal de données, il faudrait les ranger dans un payment_data, c'est plus pour du debug qu'autre chose mais c'est bien pratique.

Ça ne change pas grand chose dans make_payment, sauf que sans paid_amount/remaining_amount on devra calculer ainsi la création des InvoicePayment:

```
invoice_remaining_amount = invoice.payment - sum(invoice_payment.amount for invoice_payment in invoice.payments)
invoice_amount_paid = min(remaining_amount, invoice_remaining_amount)
InvoicePayment.objects.create(invoice=invoice, amount=invoice_amount_paid)
remaining_amount -= invoice_amount_paid
```

Au final comme dans le code actuel il peut arriver (dans l'absolu) que le montant payé dépasse le montant des factures visés ou soit inférieur, pas bien grave, parce qu'on a la balance du payeur pour faire des ajustements sur de futures factures :

```
Payer.balance = lambda: sum(invoice.amount for invoice in payer.invoices) - sum(payment.amount for payment in payer.payments)
```

de même la balance d'une régie (ou d'un compte pour reprendre le terme que j'aimerais plus) :

```
Account.balance = sum(payment.amount for payment in account.payments) - sum(invoice.amount for invoice in account.invoices)
```

Avec l'introduction du modèle Payer je serai aussi d'avis de le verrouiller lui plutôt que chaque facture (via .select_for_update()) parce que c'est plus simple pour raisonner, pendant une transaction on s'occupe de toute ce qui concerne un payeur et rien d'autre.

#8 - 18 avril 2023 20:07 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

Ensuite j'aurai un objet Payment pour le paiement en lui même qui référencerait le payeur (Payer) et la régie/compte de facturation concerné et des InvoicePayment pour la ventilation et aurait un identifiant de paiement (payment_reference) venant de combo/lingo, ou n'importe quoi qui nous déclare un paiement (formulaire pour déclarer un paiement en espèce/chèque au guichet) pour éviter de compter un paiement deux fois, lingo remontant pas mal de données, il faudrait les ranger dans un payment_data, c'est plus pour du debug qu'autre chose mais c'est bien pratique.

Je pense même qu'il faudrait prévoir une ventilation qui descende niveau des lignes de facture, supposons une ligne "Cantine du 01/14 au 30/04 | 20

repas | payé: 0 euros, du: 20 euros ", on facture c'est payé. 1 mois plus tard le pointage est modifié ou le tarif révisé ou le QF change, que sais-je, mais le montant de la dette change, et donc la ligne aurait du être 18 euros, il faut pouvoir régénérer une ligne "payé: 20 euros du: -2 euros".

#9 - 21 avril 2023 16:42 - Robot Gitea

- Statut changé de Solution proposée à En cours

Benjamin Dauvergne (bdauvergne) a relu et demandé des modifications sur une pull request sur Gitea concernant cette demande :

- URL : <https://git.entrouvert.org/entrouvert/lingo/pulls/49>

#10 - 21 avril 2023 17:11 - Robot Gitea

- Statut changé de En cours à Solution proposée

#11 - 21 avril 2023 17:11 - Robot Gitea

- Statut changé de Solution proposée à Solution validée

Benjamin Dauvergne (bdauvergne) a approuvé une pull request sur Gitea concernant cette demande :

- URL : <https://git.entrouvert.org/entrouvert/lingo/pulls/49>

#12 - 24 avril 2023 09:19 - Robot Gitea

- Statut changé de Solution validée à Résolu (à déployer)

Lauréline Guérin (lguerin) a mergé une pull request sur Gitea concernant cette demande :

- URL : <https://git.entrouvert.org/entrouvert/lingo/pulls/49>
- Titre : Facturation: pouvoir déclarer un règlement sur une facture ([#76130](#))
- Modifications : <https://git.entrouvert.org/entrouvert/lingo/pulls/49/files>

#13 - 24 avril 2023 16:14 - Transition automatique

- Statut changé de Résolu (à déployer) à Solution déployée

#14 - 25 juin 2023 04:42 - Transition automatique

Automatic expiration