

API factures pour logiciels métiers

Cette API doit être implémentée par tout logiciel générant des factures (chez nous chaque instance d'un tel service se nomme une régie) et qui souhaite que celles-ci puissent être payées dans le portail Publik, il permet à Publik de :

- lister les factures à payer
- lister l'historique des factures déjà payées ou qui ne sont plus payables en ligne (ex.: recouvrement trésor)
- lister les identifiants des utilisateurs ayant des factures à payer (pour envoi des mails de notification)
- récupérer le détail d'une facture
- récupérer le fichier PDF d'une facture
- signaler le paiement d'une facture

Publik gère la relation avec le prestataire de paiement et présente les interfaces aux usager pour le paiement des factures.

La ville et le prestataire du logiciel doivent mettre en place un moyen d'identification des débiteurs (codes sur les factures papiers ou envoyés par mail, ou tout autre moyen de communication).

On appellera dans le reste de ce texte "débiteur" l'utilisateur qui a des factures à payer.

Style de l'API

Une API dans le style "REST" (GET/POST sur différentes URLs structurées) utilisant le format d'échange JSON.

Authentification de Publik sur les web-services du logiciel métier

Un seul mode d'authentification est supporté: HTTP Basic (https://fr.wikipedia.org/wiki/Authentification_HTTP#M%C3%A9thode_%C2%AB_Basic_%C2%BB)

Variables

Ces variables sont susceptibles d'apparaître dans la description des web-services.

Nom	Description
<base_url>	l'URL de base des web-services (ex.: https://logiciel-metier.maville.fr/api/factures/)
<base_path>	le chemin de base des web-services (ex.: /api/factures/)
<base_host>	le nom de domaine de base des web-services (ex.: logiciel-metier.maville.fr)
<name_id>	l'identifiant de fédération avec l'application métier (ou sub OpenID connect si l'application métier propose la fédération de son portail web avec Publik via le protocole OpenID Connect), généré par le portail/la GRU Publik

Appairage / Identification des débiteurs

Il est nécessaire de faire correspondre l'identifiant de l'utilisateur au sein du portail Publik avec l'identifiant du débiteur dans le logiciel métier. Pour cela il sera remis des identifiants aux utilisateurs, la forme de ces identifiants dépend du cas d'usage/logiciel métier.

Création d'une liaison

Cette API est en partie définie par l'éditeur, en effet il devra préciser en plus du paramètre obligatoires la définition des paramètre nécessaire à l'identification d'un débiteur.

POST /link?NameID=<nameid>&<other_parameters> HTTP/1.1

<pas de contenu>

Un exemple d'autres paramètres pourrait être code_famille et code_secret par exemple pour une logiciel de gestion pour les services

enfance/petite enfance.

```
POST /link?NameID=<nameid>&codeFamille=12345&codeSecret=XYZ HTTP/1.1
```

<pas de contenu>

Retour passant

La liaison a été créée ou elle existait déjà.

200 Ok

Content-Type: application/json

```
{"err": 0}
```

Retour non passant

La liaison ne peut-être créée.

200 Ok

Content-Type: application/json

```
{"err": 0, "err_desc": "identifiants inconnus"}
```

Listing des liaisons

```
GET /links?NameID=<nameid> HTTP/1.1
```

Retour passant

200 Ok

Content-Type: application/json

```
{
  "err": 0,
  "data": {
    "links": [
      "identifiant-débiteur-1",
      "identifiant-débiteur-2",
    ]
  }
}
```

Retour non-passant

404 Not found

```
{"err": 1, "err_desc": "aucune liaison"}
```

Suppression des liaisons

API via un POST sans contenu.

```
POST /unlink?NameID=<nameid> HTTP/1.1
```

<pas de contenu>

Retour passant

200 Ok

Content-Type: application/json

```
{"err": 0}
```

Retour non-passant

Aucun retour non passant n'est possible, toutes les liaisons depuis le NameID précisé devront être supprimées.

Remontée de la liste des factures à payer pour une régie donnée:

GET /<base_path>/invoices/?NameID=<nameid> HTTP/1.1
Host: <base_host>

Retour passant

200 Ok
Content-Type: application/json

```
{
  "data": [
    {
      "id": "934395",
      "label": "restauration août 2015",
      "amount": "37.26",
      "total_amount": "37.26",
      "online_payment": true,
      "created": "2015-08-01",
      "pay_limit_date": "2015-09-29",
      "has_pdf": true,
      "paid": false,
      "no_online_payment_reason": "litigation",
      ... (autres informations si le logiciel backend en donne)
    },
    {
      "id": "93439XX",
      "label": "vacances scolaires novembre 2015",
      "amount": "33.26",
      "total_amount": "33. 26",
      "online_payment": false,
      "created": "2015-08-01",
      "pay_limit_date": "2015-09-29",
      "has_pdf": true,
      "paid": false,
      "no_online_payment_reason": "autobilling",
      ... (autres informations si le logiciel backend en donne)
    },
    { ... }
  ],
  "err": 0
}
```

Détail des champs

Identifiant	Description	Type JSON
id	l'identifiant de la facture dans le logiciel métier	string
label	l'intitulé de la facture	string
amount	le reste à payer si paiement partiel par ailleurs (Publik lui même ne permet pas les paiements partiels)	string, nombre décimal, le point est le séparateur décimal
total_amount	le montant total de la facture initiale	idem amount
online_payment	false si la facture est à payer par tout autre moyen de paiement que en ligne(prélèvement automatique, contentieux).	booléen
no_online_payment_reason	contient un "slug" du motif de l'impossibilité de non paiement en ligne, par exemple: autobilling (paiement par	string

	prélèvement bancaire), litigation (il y a un souci avec cette facture)	
has_pdf	un PDF de la facture est disponible	booléen
paid	facture payée oui/non	booléen
created, pay_limit_date	dates de création et limite de paiement (date incluse, i.e. si 2015-09-29 alors le paiement ne sera plus possible le 30 septembre),	string, date au format YYYY-MM-DD, dit ISO8601

Retour non-passant

Tout code HTTP non-200 ou "err": 1 ou si le content-type n'est pas application/json. On signalera la source de l'erreur dans "err_desc" :

```
{
  "err": 1,
  "err_desc": "aucune liaison pour cet identifiant Publik"
}
```

Remontée des factures passées(payées):

GET /<base_path>/invoices/history/?NameID=<nameid>
Host: <base_host>

Retour passant

200 Ok
Content-Type: application/json

```
{
  "data": [
    {
      "id": "934395",
      "label": "restauration août 2015",
      "amount": "37.26",
      "total_amount": "37.26",
      "online_payment": false,
      "created": "2015-08-01",
      "pay_limit_date": "2015-09-29",
      "payment_date": "2015-09-19T00:00:00",
      "has_pdf": true,
      "paid": "true",
      ... (autres informations si le logiciel backend en donne)
    },
    {
      "id": "9343xx",
      "label": "vacances scolaires novembre 2015",
      "amount": "32.64",
      "total_amount": "32.64",
      "online_payment": false,
      "created": "2015-08-01",
      "pay_limit_date": "2015-09-29",
      "payment_date": "2015-09-19T00:00:00",
      "has_pdf": true,
      "paid": "true",
      ...
    },
    { ... }
  ],
  "err": 0
}
```

Détail des champs

Identifiant	Description	Type JSON
payment_date	donnée optionnelle, mais qui pourrait être utile	string, date/heure au format YYYY-MM-DDTHH:MM:SS, dit ISO8601

Remontée des utilisateurs ayant des factures à payer:

```
GET /<base_path>/users/with-pending-invoices/ HTTP/1.1
Host: <base_host>
```

Retour passant

Retourne les nameld des comptes, et si possible la lise des factures à payer

```
200 Ok
Content-Type: application/json

{
  "err": 0,
  "data" : {
    "uuid1": {
      "invoices": [{ ... }, ... ]
    },
    "uuid2": {
      "invoices": [{ ... }, ... ]
    }
  }
}
```

Détail des champs

Identifiant	Description	Type JSON
"<uuid>X"	le NameID (identifiant Publik) de l'utilisateur concerné	string
invoices	peut-être une liste vide s'il n'est pas possible à ce moment de donner la liste des factures à payer	tableau/liste d'objets/dictionnaires

Détails d'une facture:

```
GET /<base_path>/invoices/<identifiant_facture>/ HTTP/1.1
Host: <base_host>
```

Détail des champs

<identifiant_facture>: l'identifiant métier de la facture retourné dans le champ id des APIs de listing

Retour passant

```
200 Ok
Content-Type: application/json

{
  "data": {
    "id": "934395",
    "label": "restauration août 2015",
    "amount": "37.26",
    "total_amount": "37.26",
    "online_payment": true,
    "payable": true,
  }
}
```

```
        "created": "2015-08-01",
        "pay_limit_date": "2015-09-29",
        "payment_date": "2015-09-19",
        "has_pdf": true,
        ... (autres informations si le logiciel backend en donne)
    },
    "err": 0
}
```

Récupération du PDF d'une facture:

GET /<base_path>/invoice/<identifiant_facture>/pdf/ HTTP/1.1
Host: <base_host>

Retour passant

Le fichier PDF de la facture (réponse HTTP 200, Content-Type: application/pdf, pas d'encapsulation dans du JSON ou autre).

```
200 Ok
Content-Type: application/pdf
```

....

Retour non passant

Tout code non 200 ou dont le content-type n'est pas "application/pdf".

Signaler le paiement d'une facture:

POST /<base_path>/invoice/<identifiant_facture>/pay/?NameID=<nameid>
Host: <base_host>
Content-Type: application/json

```
{
  "transaction_id": "<identifiant_transaction>",
  "transaction_date": "2016-01-05T12:00:00"
}
```

Retour passant

Le paiement a bien été pris en compte.

```
200 Ok
Content-Type: application/json
```

```
{
  "err": 0
}
```

Retour non passant

Le paiement n'a pas été pris en compte, recommencer, tout code non-200 ou contenu avec "err"=1 sera considéré comme non-passant.

```
500 Internal Server Error
Content-Type: application/json
```

```
{
  "err": 1
}
```