

# Documentation des Web APIs Cegid Loop, "Ecritures comptables (import au format JSON)"

Cette fiche technique traite le domaine fonctionnel des imports dans Cegid Loop.

Elle décrit l'ensemble des APIs liées à la gestion des imports au format JSON(POST)

## 1/ Comment accéder à Loop API publiques

URL, [**consultez LOOP API Publiques**] : Catalogue des API Cegid

## 2/ Ecritures comptables (Import) dans le catalogue des APIs Cegid

Choisissez l'option "**Voir les APIS**", puis "**Loop API Publiques**"

Cherchez le tag "Ecritures comptables", "Import":

- Soit affichage par tag selectionnez le bouton : Group by tag
- Soit affichage par "Search operations", "import"

Cegid Loop - API Publiques

IMPORT

Group by tag

Ecritures comptables (import)

**POST** Import au format JSON

## 3/ Import des écritures au format JSON (POST)



Assurez-vous que le dossier est déjà créé

1. Pour tester un API, vous devez disposer d'un "Subscription Key" et d'une "API Key"
2. Recherchez les APIS sur "Ecritures comptable (import)", et testez l'API comme suit :

# Import au format JSON

API permettant l'import des fichiers au format JSON

Écritures comptables (import)

## Request

`POST` <https://inte-api.cegid.com/loop-api-publiques/importJson>

## Request body

```
{ "codelbs": "string", "data": {} }
```

## Response: 200 OK

```
{ }
```

3. Cliquez sur

[Try it ▶](#)

4. Renseignez "subscription key", elle est obtenue par une demande au référent partenaire de Cegid.

- Ajoutez dans le Headers le champ **x-apikey** et renseignez la valeur de ApiKey
- Renseignez le code "**Dossier du client**" au niveau "**codelbs**" de Body

- Au niveau de Body renseignez aussi le paramètre "data" :

```
{
  "codeIbs": "MANON003",
  "data": {
    "contexte": {
      "from": "2020-02-01T00:00:00.000Z",
      "to": "2020-06-05T00:00:00.000Z"
    },
    "options": {
      "separatorDecimal": ".",
      "formatDate": "AAAA-MM-JJThh:mm:ss.nnnZ",
      "multiPeriode": false,
      "failOnUnbalanced": true,
      "createNewJournaux": true,
      "createNewComptes": true,
      "createNewTiers": true
    },
    "ecritures": [
      {
        "date": "2020-06-02T00:00:00.000Z",
        "debit": {
          "amount": 1000,
          "currency": "EUR",
          "currencyAmount": 1000,
          "currencyRate": 1
        }
      }
    ]
  }
}
```

Cegid Loop - API Publiques / Import au format JSON

POST /importJson

Authorization

Subscription key

Parameters

+ Add parameter

Headers

<input type="text" value="x-apkey"/>	<input type="text" value="1EKCKNJSIAI825JQG0EXX8JPI4:"/>	Remove
<input type="text" value="Cache-Control"/>	<input type="text" value="no-cache"/>	Remove
<input type="text" value="Ocp-Apim-Subscr"/>	<input type="text" value="af72af01b41d46d79f55b9e6ef"/>	Remove
<input type="text" value="sessionId"/>	<input type="text" value="77f98472-9651-4b01-805f-ac"/>	Remove

5. Appuyez sur le bouton



6. La réponse est correcte, si à l'exécution vous obtenez un "accountingImportRequestId" et le Code à 200

**Server response**

Code	Details
200	<p><b>Response body</b></p> <pre>{   "accountingImportRequestId": "4dfa8727-0413-4e80-9fe2-3bea94b22833" }</pre> <p><b>Response headers</b></p> <pre>content-encoding: gzip content-type: application/json date: Fri, 16 Jul 2021 07:28:50 GMT request-id: 2cc24273-03d4-4ad5-9be6-1c63c7c0d8f0 strict-transport-security: max-age=15724800; includeSubDomains vary: Accept-Encoding x-request-id: 9c317f31040f32c7d85c0f15c46126dc</pre>

## Gestion des Imports JSON



Le format du corps de la demande est JSON

1. Le format est composé de deux clés : CodeIbs et data

### Structure du corps de la demande :



Clé	Type	Description
<u>codeIbs</u>	String	Code du dossier dans Loop
<u>data</u>	Objet JSON	Objet JSON contenant l'ensemble des paramètres et données à importer

2. L'objet DATA est constitué par des clés :

### Structure JSON de l'objet data :

Clé	Type	Description
<u>contexte</u>	Objet JSON	Objet JSON contenant 2 propriétés, les dates de début et de fin des écritures à importer
<u>options</u>	Objet JSON	Objet JSON contenant des options pour l'import
<u>ecritures</u>	Array JSON	Tableau de JSON décrivant une écriture

### Structure JSON de l'objet contexte :

Clé	Type	Description
<u>from</u>	String	Date de début des écritures à importer au format iso
<u>to</u>	String	Date de fin des écritures à importer au format iso

3. Exemple de dates : - "from" : "2020-02-01T00 :00 :00.000Z" - "to" : "2020-06-04T00:00:00.000Z" \* La propriété contexte n'est pas obligatoire si l'option multiPeriode vaut true

#### 4. L'objet options

##### Structure JSON de l'objet options :

Clé	Type	Description
<u>separatorDecimal</u>	String	Séparateur de décimal pour les montants
<u>formatDate</u>	String	Format de date à utiliser pour l'import des écritures
<u>multiPeriode</u>	Boolean	Booléen permettant d'indiquer si les écritures sont autorisées sur toutes les périodes ouvertes ( <u>true</u> ) ou juste sur la dernière période ouverte ( <u>false</u> )
<u>failOnUnbalanced</u>	Boolean	Booléen permettant de déterminer si l'on souhaite lever une erreur en cas de groupes d'écritures non équilibrées
<u>createNewJournaux</u>	Boolean	Booléen indiquant s'il faut créer un nouveau journal quand le journal de l'écriture n'existe pas dans le dossier
<u>createNewComptes</u>	Boolean	Booléen indiquant s'il faut créer un nouveau compte quand le compte de l'écriture n'existe pas dans le dossier
<u>createNewTiers</u>	Boolean	Booléen indiquant s'il faut créer un nouveau tiers quand le tiers de l'écriture n'existe pas dans le dossier. Si <u>false</u> , alors l'écriture sera créée avec les tiers d'attente de l'écriture

- Exemple d'options: - "separatorDecimal": ".", - "formatDate": "AAAA-MM-JJThh:mm:ss.nnnZ", - "multiPeriode": false, - "failOnUnbalanced": true, - "createNewJournaux": false, - "createNewComptes": true, - "createNewTiers": true
- Le code journal

#### 5. L'objet écriture

##### Structure JSON de l'objet écritures :

Clé	Type	Description
<u>fichier</u>	Object JSON	Objet JSON contenant l'url de la pièce jointe associée au mouvement
<u>date</u>	String	Date comptabel de l'écriture
<u>debit</u>	Object JSON	Objet JSON contenant les informations sur le débit (amount, currency, currencyAmount, currencyRate)
<u>credit</u>	Object JSON	Objet JSON contenant les informations sur le débit (amount, currency, currencyAmount, currencyRate)
<u>journal</u>	String	Code du journal
<u>compte</u>	String	Numéro de compte
<u>libelle</u>	String	Libelle de l'écriture
<u>reference</u>	String	Référence de l'écriture
<u>écritureOrigine</u>	String	Ecriture d'origine au format uuid
<u>axeAna[codeAxe]</u>	Array JSON	Objet JSON décrivant l'analytique avec [codeAxe] représentant le code l'axe analytique

- Exemple d'écriture: - "fichier": { "uri": https://test.rdd.com/123456754321.txt // max 256 caractères } - "date": "2019-06-02T00:00:00.000Z", - "debit": { "amount": 1000, // montant avec séparateur pour les décimales (decimal(18,4)) "currency": "EUR", // ISO 4217 "currencyAmount": 1000, // (decimal(18,4)) "currencyRate": 1 // (decimal(18,4)) }, - "credit": { "amount": 0, // (decimal(18,4)) "currency": "EUR", // ISO 4217 "currencyAmount": 0, // (decimal(18,4)) "currencyRate": 1 // (decimal(18,4)) }, - "journal": "ACH", // varchar(100) - "compte" : "21830000", // varchar(100) - "libelle" : "Facture PC DELL AV01", // varchar(100) -

"reference" : "FAC-2020-01-29", // varchar(100) - "écritureOrigine" : "0d07ade2-c4fa-4892-90bd-2dbd50d1c6b7" // guid de l'écriture d'origine

## 6. L'objet AxeAna

### Structure JSON de l'objet axeAna[codeAxe]:

Clé	Type	Description
<u>debit</u>	Object JSON	Objet JSON contenant les informations sur le débit (amount, currency, currencyAmount, currencyRate)
<u>credit</u>	Object JSON	Objet JSON contenant les informations sur le débit (amount, currency, currencyAmount, currencyRate)
<u>section</u>	String	Section de l'axe

### *Exemple d'analytique :*

```
- "debit": {
  "amount": 700, // montant avec séparateur pour les décimales (decimal(18,4))
  "currency": "EUR", // ISO 4217
  "currencyAmount": 700, // (decimal(18,4))
  "currencyRate": 1 // (decimal(18,4))
},
- "credit": {
  "amount": 0, // montant avec séparateur pour les décimales (decimal(18,4))
  "currency": "EUR", // ISO 4217
  "currencyAmount": 0, // (decimal(18,4))
  "currencyRate": 1 // (decimal(18,4))
},
- "section": "RH" // varchar(100)
```

## Le retour

1. Code retour http de la réponse si succès : 200 Format corps de la réponse si succès : JSON

### Structure du corps de la réponse si succès :

Clé	Type	Valeur
<u>accountingImportRequestId</u>	String	String sous forme de guid par exemple: "e8bec446-1d5c-4e78-9017-6e2600887a09". Ce guid permettra de suivre la demande via un autre endpoint <i>getImportStatus</i>

2. En cas d'erreur :

Structure du corps de la réponse si erreur:



Clé	Type	Valeur
<u>error</u>	Object JSON	Objet JSON contenant 2 propriétés : <i>instanceId</i> et <i>message</i>
<u>instanceId</u>	String	String sous forme de guid correspondant à l'id de l'instance du service d'import
<u>message</u>	String	Message explicatif de l'erreur : cf liste ci-dessous

Liste des messages d'erreur possibles :

- Il n'y a pas de payload : la méthode est-elle bien en POST dans la requête ?
- Le contexte est obligatoire sans l'option multiPeriode
- Il n'y a pas d'écritures à importer (l'objet écritures est vide)
- Les écritures doivent être présentées sous formes de tableau
- Le tableau d'écritures doit comporter au moins deux lignes
- Les dates du contexte doivent être au format ISO 'YYYY-MM-DDTHH:mm:ss.SSSZ'