

# Documentation API GP



Révision	COMMENTAIRES	DATE	REDACTEUR
0.1	- Version initiale - API Eligibilité	18/01/2023	S.Dupays
0.2	- Ajout API Souscription	25/01/2023	S.Dupays
0.3	- Ajout API infos client	27/01/2023	S.Dupays
0.4	- Ajout du code BIC dans l'API Souscription	02/02/2023	S.Dupays
0.5	- Ajout API CoordonneesBancaires - Ajout API Factures	21/02/2023	S.Dupays
0.6	- Ajout API Mandat SEPA	28/02/2023	S.Dupays

## SOMMAIRE

1. [WebService Eligibilite](#)
  - 1.1 [Obtention de la ville](#)
  - 1.2 [Obtention des adresses possibles](#)
  - 1.3 [Obtention du résultat de l'éligibilité](#)
  
2. [WebService Souscription](#)
  - 2.1 [Effectuer une commande](#)
  
3. [WebService Client](#)
  - 3.1 [Obtention des infos Client](#)
  - 3.2 [Obtention des coordonnées bancaire](#)
  - 3.3 [Modification des coordonnées bancaire](#)
  - 3.4 [Obtention du mandat SEPA](#)
  
4. [WebService Facture](#)
  - 4.1 [Obtention de la liste des factures d'un client](#)
  - 4.2 [Obtention d'une facture client](#)

## 1. WebService Eligibilite

Le Webservice Eligibilite se trouve aux adresses suivantes :

- Pour l'environnement de test : <https://sandbox-api.gotico-telecom.fr/sicomGP/api-v1/>
- Pour la production :

Il est composé de plusieurs méthodes :

- **GET /adresseGouv/communes/codePostal/{codePostal}** : elle permet de lister les villes correspondantes par rapport au code postal fourni.
- **POST /adresseGouv/autocomplete** : elle permet d'obtenir une liste d'adresses normées.
- **POST /eligibilites/operateurFacturant** : elle permet d'obtenir la liste des offres éligibles pour une adresse donnée.

### 1.1 Obtention de la ville

La méthode GET /adresseGouv/communes/codePostal/{codePostal} permet de lister les villes correspondantes par rapport au code postal fourni.

Cette méthode prend donc en paramètre un codePostal. Exemple d'appel :

```
curl -X 'GET' \  
  'https://sandbox-api.gotico-telecom.fr/sicomGP/api-  
v1/adresseGouv/communes/codePostal/92290' \  
  -H 'accept: */*'
```

et retourne un JSON au format suivant :

```
[  
  {  
    "nom": "Châtenay-Malabry",  
    "code": "92019",  
    "codeDepartement": "92",  
    "siren": "219200193",  
    "codeEpci": "200054781",  
    "codeRegion": "11",  
    "codesPostaux": [  
      "92290"  
    ],  
    "population": 34021  
  }  
]
```

## 1.2 Obtention des adresses possibles

La méthode POST /adresseGouv/autocomplete permet d'obtenir une liste d'adresses normées avec les meilleurs scores (c'est-à-dire correspondant au maximum à la saisie utilisateur parmi les adresses de la BAN).

Cette méthode prend donc en paramètre un JSON constitué d'un codePostal, d'une ville et d'une adresse partiellement remplie (minimum 3 caractères) ou complètement saisie par l'utilisateur.

Exemple d'appel :

```
curl -X 'POST' \
  'https://sandbox-api.qotico-telecom.fr/sicomGP/api-v1/adresseGouv/autocomplete' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "codePostal": "92290",
    "ville": "Châtenay-Malabry",
    "adresse": "1 rue des "
  }'
```

La réponse sera un JSON composé de la liste des adresses possibles avec les meilleurs scores :

```
{
  "features": [
    {
      "properties": {
        "label": "1 Rue des Vallées 92290 Châtenay-Malabry",
        "score": 0.8766481818181817,
        "houseNumber": "1",
        "id": "92019_9625_00001",
        "type": "houseNumber",
        "name": "1 Rue des Vallées",
        "postcode": "92290",
        "cityCode": "92019",
        "x": "647154.67",
        "y": "6851869.36",
        "city": "Châtenay-Malabry",
        "context": "92, Hauts-de-Seine, Île-de-France",
        "importance": "0.64313",
        "street": "Rue des Vallées"
      },
      "geometry": {
        "type": "Point",
        "coordinates": [
          2.281036,
          48.764762
        ]
      }
    },
    ...
  ]
}
```

## 1.3 Obtention du résultat de l'éligibilité

La méthode POST /eligibilites/operateurFacturant permet d'obtenir la liste des offres éligibles pour un opérateur (ici Qotico Télécom dont l'id est 1) pour une adresse donnée. Cette méthode prend donc en paramètre l'identifiant de l'opérateur (ici 1) et un JSON correspondant à l'adresse dont on veut vérifier l'éligibilité.

Exemple d'appel :

```
curl -X 'POST' \  
  'https://sandbox-api.qotico-telecom.fr/sicomGP/api-v1/eligibilites/operateurFacturant/1' \  
 \  
  -H 'accept: */*' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "codePostal": "92290",  
    "ville": "Châtenay-Malabry",  
    "adresse": "1 rue des Valleys"  
  }'
```

La réponse sera un JSON correspondant à liste d'offres et de FAS (frais d'accès au service). Pour le COAX 800M, les FAS sont offerts.

```
{  
  "offresFas": [  
    {  
      "offres": [  
        {  
          "idOffre": 1,  
          "dateDebut": "2022-12-31T23:00:00.000+00:00",  
          "dateFin": null,  
          "detail": {  
            "idDetail": 2,  
            "libelle": "GP-800Mb-COAX",  
            "description": "offre COAX 800Mb/s",  
            "prixHt": 13.99,  
            "prixTtc": 16.79,  
            "recurrent": 1,  
            "engagement": 12,  
            "prorata": 1,  
            "actif": 1,  
            "sortIndex": 1,  
            "periodicite": "MOIS",  
            "tva": {  
              "idTva": 3,  
              "libelle": "TVA A 20%"  
            }  
          }  
        }  
      ]  
    }  
  ]  
}
```

```
    },
    "debit": {
      "id": 1,
      "debitMontant": 800,
      "uniteMontant": "Mb/s",
      "debitDescendant": 800,
      "uniteDescendant": "Mb/s"
    }
  ],
  "fraisAccesService": {
    "idFraisAccesService": 1,
    "surDevis": 0,
    "detail": {
      "idDetail": 3,
      "libelle": "FAS",
      "description": "Frais d'accès au service",
      "prixHt": 0,
      "prixTtc": 0,
      "recurrent": 0,
      "engagement": 0,
      "prorata": 0,
      "actif": 1,
      "sortIndex": 4,
      "periodicite": "AUCUNE",
      "tva": {
        "idTva": 3,
        "libelle": "TVA A 20%"
      }
    }
  }
},
"adresseTestee": {
  "codePostal": "92290",
  "ville": "Châtenay-Malabry",
  "adresse": "1 Rue des Vallées 92290 Châtenay-Malabry"
}
```

## 2. WebService Souscription

Le Webservice souscription se trouve aux adresses suivantes :

- Pour l'environnement de test : <https://sandbox-api.qotico-telecom.fr/sicomGP/api-v1/>
- Pour la production :

Il est composé de plusieurs méthodes :

- **POST /souscriptions/operateurFacturant/{idOperateurFacturant}** : cette méthode d'effectuer une commande à un opérateur (ici Qotico Télécom)

### 2.1 Effectuer une commande

La méthode POST /souscriptions/operateurFacturant/{idOperateurFacturant} permet d'effectuer une commande à un opérateur (ici Qotico Télécom dont l'id est 1).

Cette méthode prend donc en paramètre l'identifiant de l'opérateur (ici 1) et un JSON correspondant aux informations clients et à l'offre voulant être souscrite.

Exemple d'appel :

```
curl -X 'POST' \  
  'https://sandbox-api.qotico-telecom.fr/sicomGP/api-v1/souscriptions/operateurFacturant/1' \  
 \  
  -H 'accept: */*' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "prenom": "stephane",  
    "nom": "dupays",  
    "civilite": 2,  
    "dateNaissance": "01/01/1980",  
    "telephone": "0123456789",  
    "email": "sdupays@infra-build.fr",  
    "iban": "q4Iuu4rGZHwt1ChtZ8MUNxU9t+h5ukKorRrzclMUxE=",  
    "bic": "F0WV9tCOG3c07vehJ2SEig==",  
    "codePostal": "92290",  
    "ville": "CHATENAY-MALABRY",  
    "adresse": "2 rue des vallees",  
    "escalier": "",  
    "etage": "",  
    "nomBatiment": "",  
    "numAppart": "",  
    "idOffre": 1,  
    "idOptions": []  
  }'
```

L'IBAN devra être chiffré en AES 256 avec une clé de cryptage qui vous sera communiqué par email.

En réponse, vous aurez un code 200 avec la référence de la commande mais il n'est pas utile de la communiquer au client.

Server response	
Code	Details
200	<p>Response body</p> <pre>CPQT202301-00002</pre>

En cas d'échec de la souscription, vous obtiendrez un retour sous la forme suivante :

```
{
  "errorId": "6058e63d-24ee-45e1-918a-3fb2617c28d2",
  "statusCode": 400,
  "status": "Validation failed",
  "errors": [
    "prenom": "Le nom doit avoir minimum 2 caractères"
  ],
  "path": "/sicomGP/api-v1/souscriptions/operateurFacturant/1"
}
```

Et dont voici une liste non exhaustive des erreurs pouvant être retournées par cette méthode :

- "prenom": "Le nom doit avoir minimum 2 caractères"
- "nom": "Le nom doit avoir minimum 2 caractères"
- "civilite": "La civilité doit être 1 ou 2"
- "Le format de la date n'est pas correct !"
- "le numéro de téléphone xxxxxx n'est pas valide"
- "email": "Le mail doit être une adresse électronique syntaxiquement correcte"
- "Erreur lors de decrypt data : Input length must be multiple of 16 when decrypting with padded cipher" : Erreur obtenue lorsque le décryptage des données chiffrées n'a pu être effectué.
- "codePostal": "Le code postal doit avoir 5 chiffres"
- "Aucune adresse n'est trouvée"
- "Pas Eligible"
- "Cette adresse e-mail : xxx est déjà utilisée !"



## 3. WebService Client

Le Webservice client se trouve aux adresses suivantes :

- Pour l'environnement de test : <https://sandbox-api.gotico-telecom.fr/sicomGP/api-v1/>
- Pour la production :

Il est composé de plusieurs méthodes :

- **GET /clients/informations** : cette méthode permet d'obtenir les informations client d'un utilisateur authentifié
- **GET /clients/coordonneesBancaire** : cette méthode permet d'obtenir les coordonnées bancaires cryptées d'un utilisateur authentifié
- **POST /clients/coordonneesBancaire** : cette méthode permet de modifier les coordonnées bancaires d'un utilisateur authentifié

### 3.1 Obtention des informations Client

La méthode GET /clients/informations permet d'obtenir les informations clients d'un utilisateur authentifié. Il n'y a pas de paramètre pour cette méthode, il faut uniquement transmettre token d'authentification.

Exemple d'appel :

```
curl -X 'GET' \  
  'https://sandbox-api.gotico-telecom.fr/sicomGP/api-v1/clients/informations' \  
  -H 'accept: */*' \  
  -H 'Authorization: Bearer TokenObtenuParLeClient0Auth2'
```

La réponse sera un JSON correspondant aux informations clients :

```
{  
  "prenom": "stef",  
  "nom": "dupays",  
  "civilite": 2,  
  "dateNaissance": "01/01/1980",  
  "telephone": "0123456789",  
  "email": "sdupays+client3@infra-build.fr",  
  "adresse": {  
    "codePostal": "92290",  
    "extension": null,  
    "numero": "2",  
    "rue": "des Vallées",  
    "ville": "Châtenay-Malabry",  
    "typeVoie": "Rue",  
    "escalier": "",  
    "etage": "",  
    "extAppart": null,  
    "localisation": null,  
    "nomBatiment": ""  
  }  
}
```

```
"numAppart": "",
"x": 647122.3,
"y": 6851873.88
},
"itemList": [
  {
    "libelle": "GP-800Mb-COAX",
    "description": "offre COAX 800Mb/s",
    "prixHt": 13.99,
    "prixTtc": 16.79,
    "montantRemiseHt": 0,
    "montantRemiseTtc": 0,
    "reduction": 0,
    "recurrent": 1,
    "engagement": 12,
    "prorata": 1,
    "sortIndex": 1,
    "periodicite": "MOIS",
    "tva": 20
  }
]
}
```

## 3.2 Obtention des coordonnées bancaires

La méthode GET /clients/coordonneesBancaire permet d'obtenir les informations bancaires d'un utilisateur authentifié. Il n'y a pas de paramètre pour cette méthode, il faut uniquement transmettre token d'authentification.

Exemple d'appel :

```
curl -X 'GET' \
'https://sandbox-api.qotico-telecom.fr/sicomGP/api-v1/clients/coordonneesBancaire' \
-H 'accept: */*' \
-H 'Authorization: Bearer TokenAuthentificationOAuth2'
```

La réponse sera un JSON au format suivant :

```
{
  "iban": "q4Iuu4rGZHwt1ChtZ8MUNxU9t+h5ukKorRrzcLMUxE=",
  "codeBanque": null,
  "codeGuichet": null,
  "numCompte": null,
  "cleRib": null,
  "bic": "X9d861RPqTfaJTzD885hMg=="
}
```

Seul l'iban et le code bic seront fournis ici et seront cryptés en AES256 avec la clé de cryptage.

### 3.3 Modification des coordonnées bancaires

La méthode POST /clients/coordonneesBancaire permet de modifier les informations bancaires d'un utilisateur authentifié. En plus du token d'authentification du client, cette méthode prend en paramètre un JSON compos des attributs iban et bic.

Exemple d'appel :

```
curl -X 'POST' \
  'https://sandbox-api.qotico-telecom.fr/sicomGP/api-v1/clients/coordonneesBancaire' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer TokenAuthentificationOAuth2' \
  -H 'Content-Type: application/json' \
  -d '{
  "iban": "q4Iuu4rGZHwt1ChtZ8MUNxU9t+h5ukKorRrzcLMUxE=",
  "bic": "X9d861RPqTfaJTzD885hMg=="
}'
```

La réponse sera un JSON au format suivant si les coordonnées bancaires sont valides :

```
{
  "iban": "q4Iuu4rGZHwt1ChtZ8MUNxU9t+h5ukKorRrzcLMUxE=",
  "codeBanque": null,
  "codeGuichet": null,
  "numCompte": null,
  "cleRib": null,
  "bic": "X9d861RPqTfaJTzD885hMg=="
}
```

### 3.4 Obtention du mandat SEPA

La méthode GET /clients/urlMandat permet d'obtenir l'url du pdf du mandat SEPA d'un client authentifié. Seul le token d'authentification du client est nécessaire pour cette méthode.

Exemple d'appel :

```
curl -X 'GET' \
  'https://sandbox-api.qotico-telecom.fr/sicomGP/api-v1/clients/urlMandat' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer TokenAuthentificationOAuth2'
```

La réponse sera une url correspondant à la facture pdf du client :

```
https://.....
```

## 4. WebService Facture

Le Webservice facture se trouve aux adresses suivantes :

- Pour l'environnement de test : <https://sandbox-api.gotico-telecom.fr/sicomGP/api-v1/>
- Pour la production :

Il est composé de plusieurs méthodes :

- **GET /factures/client** : cette méthode permet d'obtenir la liste des factures d'un client authentifié
- **GET /factures/url/{ref}** : cette méthode permet d'obtenir l'url d'une facture d'un utilisateur authentifié

### 4.1 Obtention de la liste des factures d'un client

- La méthode GET /factures/client permet d'obtenir la liste des factures d'un client authentifié. Il n'y a pas de paramètre pour cette méthode, il faut uniquement transmettre le token d'authentification.

Exemple d'appel :

```
curl -X 'GET' \  
'https://sandbox-api.gotico-telecom.fr/sicomGP/api-v1/factures/client' \  
-H 'accept: */*' \  
-H 'Authorization: Bearer TokenObtenuParLeClientOAuth2'
```

La réponse sera un JSON correspondant à une liste de factures comme suit :

```
[  
{  
  "id": 1,  
  "ref": "FPQT202302-00002",  
  "dateGeneration": "2023-02-20T15:29:37.000+00:00",  
  "montantTtc": 16.788,  
  "montantHt": 13.99,  
  "mois": 2,  
  "annee": 2023,  
  "validate": true,  
  "prelevee": false,  
  "commentaire": null  
}  
]
```

Le retour fournit donc les informations importantes pour le portail client tel que le mois et l'année de la facture, le montant TTC et la **ref** de la facture qui permettra d'obtenir l'url du pdf.

## 4.2 Obtention d'une facture client

- La méthode GET /factures/client permet d'obtenir une facture spécifique d'un client authentifié. En plus du token d'authentification, la méthode prend en paramètre la référence de la facture.

Exemple d'appel :

```
curl -X 'GET' \  
  'https://sandbox-api.qotico-telecom.fr/sicomGP/api-  
v1/factures/prelevement/ur1/FPQT202302-00002' \  
  -H 'accept: */*' \  
  -H 'Authorization: Bearer TokenObtenuParLeClient0Auth2'
```

La réponse sera une url correspondant à la facture pdf du client :

```
https://.....
```

Si la facture n'existe pas pour la référence communiquée vous aurez le message d'erreur suivant :

```
{  
  "errorId": "ea56158e-5792-4c44-96eb-5d8b28b668bc",  
  "statusCode": 404,  
  "status": "404 NOT_FOUND \"Facture ref: FPQT202302-000ABC Not Found.\",",  
  "errors": [  
    "Facture ref: FPQT202302-000ABC Not Found."  
  ],  
  "path": "/api-v1/factures/ur1/FPQT202302-000ABC"  
}
```