



cma

# SPECIFICATIONS API REST

## Messagerie

Pour la Banque d'Algérie

Numéro de version :001

© 2024, CMA Small Systems AB

Aucune partie de cette publication ne peut être reproduite ou transmise sous quelque forme ou à quelque fin que ce soit sans l'autorisation expresse de CMA Small Systems AB.

Nom du projet : Mise à niveau ARTS - RTGS de la Banque d'Algérie

Nom du document : Spécifications API REST de Messagerie

## CONTENTS

<b>HISTORIQUE DES REVISIONS .....</b>	<b>5</b>
<b>1 VUE D'ENSEMBLE .....</b>	<b>6</b>
<b>2 AUTHENTIFICATION .....</b>	<b>7</b>
<b>3 RECEPTION DE MESSAGES.....</b>	<b>8</b>
3.1 'GET MX MESSAGE' .....	8
<b>4 ENVOI DE MESSAGES .....</b>	<b>11</b>
4.1 METHODE 'POST MX MESSAGE' .....	11
<b>5 SIGNATURE NUMERIQUE ET VERIFICATION DE SIGNATURES DE MESSAGES MX.....</b>	<b>14</b>
<b>6 CODES D'ERREUR HTTP COURANTS .....</b>	<b>15</b>
<b>7 REFERENCE API.....</b>	<b>16</b>
7.1 METHODE 'GET INFO'.....	16
7.1.1 DESCRIPTION DE LA METHODE .....	16
7.1.2 REQUÊTE .....	16
7.1.3 RESPONSE FIELDS .....	16
7.1.4 HTTP REQUEST .....	17
7.1.5 RÉPONSE HTTP .....	17
7.2 METHODE 'GET MX MESSAGE' .....	17
7.2.1 DESCRIPTION DE LA METHODE .....	17
7.2.2 VARIABLES D'URL .....	18
7.2.3 EN-TETES DE REQUETE .....	18
7.2.4 EXEMPLE DE REQUETE .....	19
7.2.5 EN-TETES DE REPONSE.....	19
7.2.6 EXEMPLE DE REPONSE DE REUSSITE.....	19
7.2.6.1 REQUÊTE HTTP .....	20
7.2.6.2 RÉPONSE HTTP .....	20
7.2.7 EXEMPLE REUSSI SANS MESSAGES : .....	20
7.2.7.1 REQUÊTE HTTP .....	21
7.2.7.2 RÉPONSE HTTP .....	21
7.2.8 EXEMPLE DE REPONSE D'ERREUR .....	21
7.2.8.1 ERREUR DU SERVEUR.....	21
7.2.8.2 ERREUR DE MAUVAISE REQUETE .....	22
7.2.8.3 DELAI D'ATTENTE DE RECUPERATION INCORRECT .....	23
7.2.8.4 TAILLE DE RECUPERATION INCORRECTE .....	24
7.2.8.5 STATUT INVALIDE.....	25
7.2.8.6 IDENTIFIANT DE DEMANDE INCORRECT .....	26
7.2.9 CODES DE REPONSE HTTP .....	27
7.2.10 PARAMETRES DE REPONSE DE REUSSITE.....	28
7.2.11 PARAMETRES DE REPONSE D'ERREURS .....	29

7.3 METHODE 'POST MX MESSAGE' .....	29
7.3.1 DESCRIPTION DE LA METHODE .....	29
7.3.2 VARIABLES D'URL .....	30
7.3.3 EN-TETES DE REQUETE .....	30
7.3.4 PARAMETRES DE REQUETE .....	32
7.3.5 REQUETE HTTP .....	33
7.3.6 EN-TETES DE REPONSE.....	34
7.3.7 EXEMPLE DE REPONSE DE REUSSITE.....	34
7.3.7.1 REQUETE HTTP .....	34
7.3.7.2 RÉPONSE HTTP .....	35
7.3.8 EXEMPLE DE REPONSE D'ERREUR .....	35
7.3.8.1 BAD REQUEST.....	35
7.3.8.2 INTERNAL SERVER ERROR.....	36
7.3.8.3 ERREUR DU SERVEUR.....	37
7.3.8.4 TRACEREFERENCE INCORRECTE .....	38
7.3.8.5 IDENTIFIANT DE DEMANDE INCORRECT .....	38
7.3.8.6 MESSAGE DUPLIQUE.....	39
7.3.8.7 XML NON AUTORISE (400) .....	40
7.3.8.8 VALIDATION DE SCHEMA XSD .....	41
7.3.8.9 VALIDATION DE LA SIGNATURE .....	42
7.3.9 CODES DE REPONSE HTTP .....	43
7.3.10 PARAMETRES DE REPONSE AUX ERREURS .....	44
7.4 LIMITATION DU NOMBRE DE REQUETES .....	45
7.4.1 NOMBRE DE REQUETES DEPASSEE .....	45
7.4.1.1 REQUETE HTTP .....	45
7.4.1.2 REPONSE HTTP .....	48
<b>8 REFERENCE API (A PARTIR DE YAML) .....</b>	<b>49</b>

## Historique des révisions

Version	Date	Auteur	Commentaires
001	17.06.2025	CMA Small Systems AB	Première version

## 1 Vue d'ensemble

Ce document décrit l'interface de programmation d'application (API) utilisée pour l'échange de messages MX entre les participants et les systèmes de CMA (RTS/X, BCS/X, IPS/X, DEPO/X).

**Note :**

Les systèmes répertoriés sont soumis à la disponibilité de cette interface de messagerie en fonction de la version actuelle et des spécifications préalablement convenues.

L'API de messagerie REST (interface basée sur les requêtes) permet aux participants :

- D'envoyer des messages MX (POST).
- De recevoir des messages MX (GET).
- D'obtenir des informations (GET).

Chaque participant devra implémenter l'API de son côté.

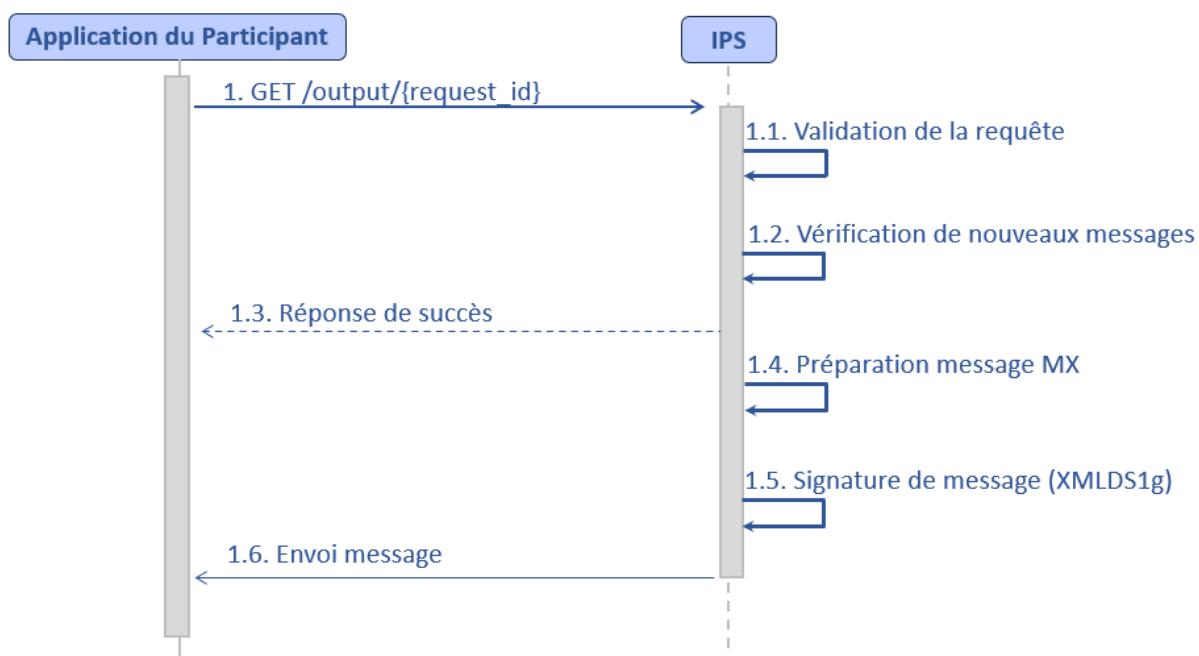
## 2 Authentification

Pour des informations détaillées sur l'authentification, voir le document "Spécification de l'API REST du service d'authentification".

## 3 Réception de messages

### 3.1 'Get MX message'

L'application du participant doit générer un ID de demande. Répéter instantanément la requête GET output message dans un thread séparé pour recevoir tous les messages en sortie. Il est recommandé de suivre le protocole RFC 4122 pour générer l'ID de demande. Plusieurs messages peuvent être reçus dans une seule réponse.



**Figure 3.1. Méthode Get MX message**

La méthode est idempotente. En cas de défaillance du réseau (interruption de la connexion, délai d'attente de connexion ou tout autre problème lié au réseau), l'application du participant est responsable de l'envoi répétitif de messages avec le même *request\_id* et le même contenu jusqu'à ce qu'à la réception l'une des réponses suivantes :

- Le participant peut ouvrir un thread et effectuer un appel API GET avec une valeur de délai d'attente spécifique (requête à pooling long, 10 secondes recommandées, X-Fetch-Timeout)
- Si un message est reçu sur le thread, il sera reçu par le participant et traité.
- Le participant ne doit pas lancer de nouvelle requête GET (requête avec un nouveau « *request\_id* ») dans le thread jusqu'à ce que la réponse à la précédente ne soit reçue.

Cette méthode est basée sur des requêtes de polling long. L'algorithme de l'application client pour le cas où **les messages sont renvoyés** (HTTP 200) est le suivant :

1. Ouvrir un thread et effectuez un appel GET « output » avec une valeur de délai d'attente spécifique (requête de pooling long, 10 secondes recommandées, X-Fetch-Timeout) et attendre la réponse.
2. La méthode répond immédiatement avec HTTP 200 avec un ensemble de messages pour un traitement ultérieur sur l'application client.
3. Passez à l'étape 1.

L'algorithme de l'application client pour le cas où **les messages ne sont pas renvoyés** (HTTP 204) est le suivant :

1. Ouvrir un thread et effectuez un appel GET « output » avec une valeur de délai d'attente spécifique (requête de pooling long, 10 secondes recommandées, X-Fetch-Timeout) et attendre la réponse.
2. La méthode répond avec HTTP 204 après X-Fetch-Timeout.
3. Passez à l'étape 1.

L'algorithme de l'application client pour le cas où **la réponse n'est ni 200 ni 204** est le suivant :

1. Ouvrir un thread et effectuez un appel GET « output » avec une valeur de délai d'attente spécifique (requête de pooling long, 10 secondes recommandées, X-Fetch-Timeout) et attendre la réponse.
2. La méthode répond avec un code HTTP différent de 200/204 après X-Fetch-Timeout.
3. Répéter l'appel à partir de l'étape 1 (avec la même « request\_id ») jusqu'à ce que 200/204 soit renvoyé.

L'algorithme de l'application client pour le cas où **la réponse n'est pas reçue** est le suivant :

1. Ouvrir un thread et effectuer un appel GET « output » avec une valeur de délai d'attente spécifique (requête de pooling long, 10 secondes recommandées, X-Fetch-Timeout) et attendre la réponse.
2. La connexion réseau a été fermée (délai d'expiration/problème de réseau).
3. Répéter l'appel à partir de l'étape 1 (avec le même « request\_id ») jusqu'à ce que 200/204 soit renvoyé.

L'application client n'arrête pas d'envoyer une nouvelle requête GET « output » (requête avec une nouvelle « *request\_id* ») à l'intérieur d'un thread jusqu'à ce que la réponse à la précédente soit reçue.

**Note :**

L'application du participant peut recevoir des messages dans un seul thread ou dans des threads parallèles. Dans ce dernier cas, l'application du participant est responsable du bon traitement des messages dans l'ordre chronologique (par exemple en raison du parallélisme, un message de statut de paiement peut arriver plus tôt que le message de paiement correspondant).

Description	Demander à recevoir un message
Méthode	GET
Client	Application du participant
Serveur	Serveur d'accès
URL	<a href="https://&lt;accessserver:port&gt;/output/{request_id}">https://&lt;accessserver:port&gt;/output/{request_id}</a>

Pour plus de détails, voir [7.2 Méthode 'Get MX message'](#).

## 4 Envoi de messages

### 4.1 Méthode 'Post MX message'

L'application du participant est responsable de la préparation et de la signature du message à envoyer. Une fois le message préparé et signé, l'application du participant doit générer un ID de demande de niveau transport pour ce message d'entrée et l'envoyer au serveur à l'aide de la méthode POST de l'API REST.

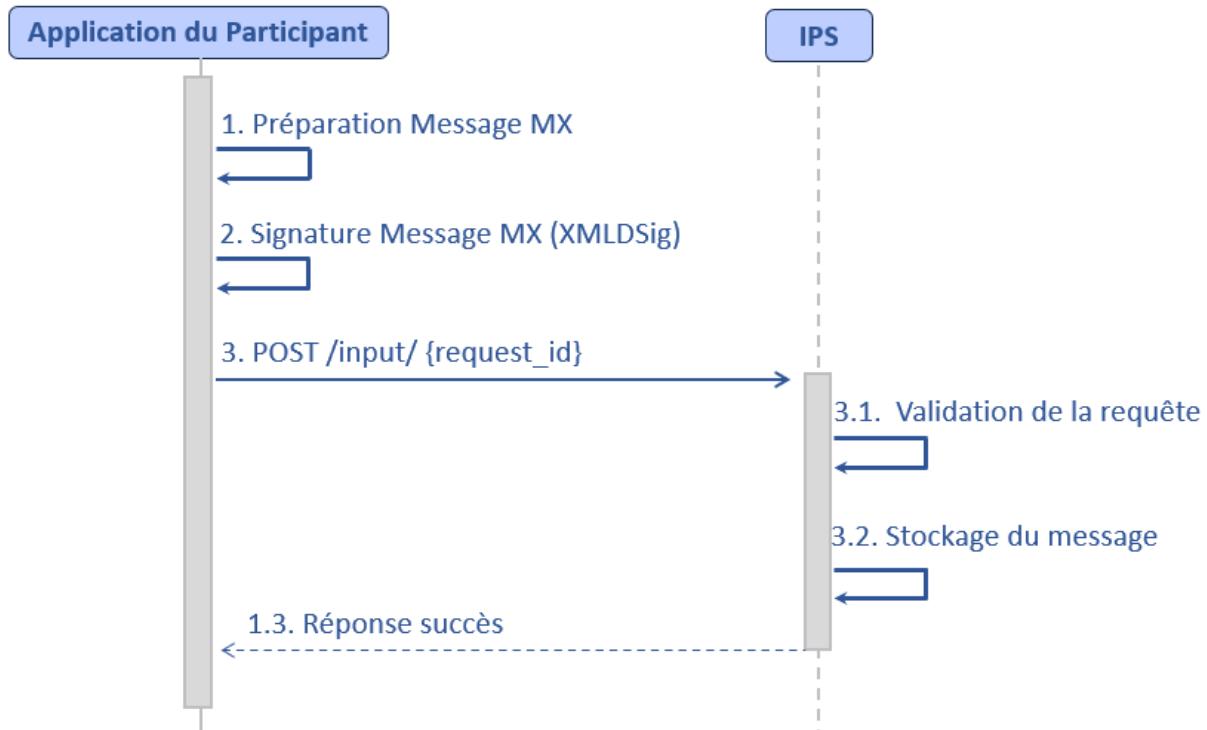
Il est recommandé de suivre le protocole RFC 4122 pour générer l'ID de demande.

Un seul message peut être envoyé dans une demande. Plusieurs demandes d'envoi simultanées sont autorisées.

#### Note :

#### Minification des Messages en XML

Avant d'envoyer et de signer vos messages, veuillez les minifier. La minification XML consiste à supprimer les espaces inutiles, les sauts de ligne et d'autres éléments non essentiels des messages XML. Ce processus rend les messages plus compacts et réduit la bande passante nécessaire pour leur transfert. Par conséquent, la minification améliore les performances d'échange et de traitement des messages.



**Figure 4.1. Méthode Post de message MX**

La méthode est idempotente. En cas de défaillance du réseau (interruption de connexion, délai d'attente de connexion ou tout autre problème lié au réseau), l'application du participant est responsable des tentatives ultérieures d'envoi du message avec le même `request_id` et le même contenu jusqu'à la réception de réponse « opération réussie » (HTTP code d'état 200) ou de la réponse « demande incorrecte » (code d'état HTTP 400).

L'algorithme de l'application client pour le cas **où la méthode renvoie le code HTTP 200/400** est le suivant :

1. Ouvrir un thread, effectuer un appel POST « input » et attendre la réponse.
2. La méthode répond avec HTTP 200/400.
3. Passer à l'étape 1 (nouveau « `request_id` », nouveau contenu).

L'algorithme de l'application client pour le cas **où la réponse n'est pas reçue** est le suivant :

1. Ouvrir un thread, effectuer un appel POST « input » et attendre la réponse.
2. La méthode renvoie un code HTTP différent de 200/400.

3. Répétez l'appel à partir de l'étape 1 (même « request\_id », même contenu) jusqu'à ce que le code HTTP soit reçu.

Description	Demande d'envoi d'un message MX
Méthode	POST
Client	Application du participant
Serveur	Serveur d'accès
URL	<a href="https://&lt;access-server:access-server-port&gt;/input/{request_id}">https://&lt;access-server:access-server-port&gt;/input/{request_id}</a>

Pour plus de détails, voir [7.3 'Méthode 'Post MX message'](#).

## **5 Signature numérique et vérification de signatures de messages MX**

Pour les détails d'information sur la signature et la vérification des signatures des messages MX, consultez le document « Guide de signature des messages MX ».

## 6 Codes d'Erreur HTTP Courants

Error code	Nom	Description
<b>Code d'erreur</b>	Non Autorisé	Erreur d'authentification (token manquant, signature de token invalide, etc.).
400	Mauvaise Requête	NAK métier pour les messages entrants ou réponse incorrecte en cas d'erreurs lorsque le message est invalide et ne peut pas être traité.
405	Méthode Autorisée	Non Le client a utilisé une méthode HTTP incorrecte, par exemple, GET au lieu de POST.
406	Non Acceptable	Le client a utilisé un en-tête Accept incorrect, empêchant le serveur de préparer une réponse appropriée.
409	Conflit	Le client a effectué une demande de récupération répétée alors que le traitement de la demande initiale était toujours en cours.
415	Type de Média Non Pris en Charge	Le client a utilisé un Content-Type incorrect, tel que application/xml alors que application/json est requis, etc.
429	Trop Demandes	de La limite de demandes au serveur a été dépassée dans une certaine période (généralement une minute ou une heure).
500	Erreur Interne du Serveur	Erreur du serveur lors du traitement de la demande.
503	Service Indisponible	Délai d'attente temporaire ou indisponibilité du serveur.

## 7 Référence API

### 7.1 Méthode 'Get Info'

#### 7.1.1 Description de la méthode

Description	Requête pour obtenir des informations supplémentaires : Bic du système central, format du message, etc.
Méthode	GET
Client	Application du participant
Serveur	Serveur d'authentification
URL	<a href="https://&lt;auth-server-host:auth-server-port&gt;/info">https://&lt;auth-server-host:auth-server-port&gt;/info</a>

#### 7.1.2 Requête

```
$ curl 'https://asrv:23432/info' -i -X GET \
-H 'Accept: application/json' \
-H 'Authorization: Bearer eyJ4NXQi...'
```

#### 7.1.3 Response fields

Chemin	Type	Description
messageReceiver	String	Destinataire du message (BIC du système central)
messageFormat	String	Format du message
projectCode	String	Code du projet
bizSvc	String	Business Service

## 7.1.4 HTTP request

```
GET /info HTTP/1.1
Accept: application/json
Authorization: Bearer eyJ4NXQ...
Host: asrv:23432
```

## 7.1.5 Réponse HTTP

```
HTTP/1.1 200 OK
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
Content-Length: 112
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
{
    "messageReceiver" : "SBPPPKKAXIPS",
    "messageFormat" : "MX",
    "projectCode" : "IPSPK",
    "bizSvc" : "SN"
}
```

## 7.2 Méthode 'Get MX message'

### 7.2.1 Description de la méthode

Description	Demander à recevoir un message
Méthode	GET
Client	Application du participant
Serveur	Serveur d'accès
URL	<a href="https://&lt;accessserver:port&gt;/output/{request_id}">https://&lt;accessserver:port&gt;/output/{request_id}</a>

## 7.2.2 Variables d'URL

Variable	Description
request_id	Référence unique du message en sortie. Elle doit comporter moins de 64 caractères et contenir uniquement les symboles autorisés, listés dans l'expression suivante :  [0-9a-zA-Z-._] {1, 64}

## 7.2.3 En-têtes de requête

Accept	Définissez cet en-tête sur "application/json".	Requis
X-Fetch-Timeout	<p>Délai d'expiration de la demande, en ms.</p> <p>Si aucune réponse n'est reçue immédiatement, l'expéditeur attendra la réponse jusqu'à l'expiration de ce délai.</p> <p>Par défaut, 5000 ms ; Maximum – 30 000 ms.</p> <p>Si la valeur est inférieure à la valeur par défaut, la requête est considérée comme « mauvaise requête » (http/400)</p>	Facultatif
X-Fetch-Size	<p>Nombre maximum de messages autorisés par réponse.</p> <p>Il s'agit d'un paramètre système modifiable par l'opérateur.</p> <p>La valeur maximale est 10.</p>	Facultatif
X-Timestamp	Horodatage de la demande, au format ISO 8601. Le système ne valide pas ce paramètre mais l'audite.	Facultatif

Authorization	Contient le jeton d'accès utilisé pour authentifier le Requis user-agent auprès du serveur	
Host	Adresse et port de l'hôte.	Requis

## 7.2.4 Exemple de requête

```
GET /output/0eeca02-2301-4638-bb96-b67973c57943 HTTP/1.1
Accept: application/json
X-Fetch-Timeout: 15000
X-Fetch-Size: 5
X-Timestamp: 2018-08-13T12:15:54.651Z
Authorization: Bearer eyJ...iJ9.eyJ...jJ9.Ac-...0MSw
Host: asrv:23432
```

## 7.2.5 En-têtes de réponse

Content-type	Type de média de la réponse.
Content-length	Longueur totale des messages en réponse, en octets.
X-Request-ID	Doit être {request_id} identique à la requête ci-dessus.
X-Timestamp	Horodatage de la réponse, au format ISO 8601.
X-Fetch-Count	Nombre de messages dans la réponse.
Server-Timing	Statistiques détaillées du temps serveur, passé à traiter la requête et à préparer la réponse.

La réponse peut également contenir un ensemble d'en-têtes techniques et de sécurité générés automatiquement. Ces en-têtes ne sont pas répertoriés ici.

## 7.2.6 Exemple de réponse de réussite

Le client reçoit une réponse avec le code 200 et un message dans le corps de la réponse en cas de succès, s'il y a des messages dans le système central pour l'utilisateur.

### 7.2.6.1 Requête HTTP

```
GET /output/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
Accept: application/json
X-Fetch-Timeout: 15000
X-Fetch-Size: 1
X-Timestamp: 2024-07-19T15:36:47.584763375Z
Authorization: Bearer eyJ4NXQ...
Host: asrv:23432
```

### 7.2.6.2 Réponse HTTP

```
HTTP/1.1 200 OK
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Request-ID: 0eecaf02-2301-4638-bb96-b67973c57943
X-Timestamp: 2024-07-19T15:36:47.814854558Z
X-Fetch-Count: 1
Server-Timing: app;dur=120, wait;dur=100, queue;dur=0, acc;dur=222, acc-app;dur=0
Content-Type: application/json
Content-Length: 4091
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
[
  {
    "traceReference": "CKvOI85gv0SgNKqLAXBpwQ.0",
    "type": "pacs.002.001.09",
    "sender": "SYSTECAUSR",
    "receiver": "PARTIBICXUSR",
    "document": "<DataPDU ...>...</DataPDU>"
  },
  {
    "traceReference": "IgULMaA3a0W4bksqhIrQLg.0",
    "type": "pacs.002.001.09",
    "sender": " SYSTECAUSR ",
    "receiver": " PARTIBICXUSR ",
    "document": "<DataPDU ...>...</DataPDU>"
  }
]
```

### 7.2.7 Exemple réussi sans messages :

Le client reçoit une réponse avec le code 204 et un corps de réponse vide, en cas de succès, mais sans messages pour l'utilisateur dans le système central.

### 7.2.7.1 Requête HTTP

```
GET /output/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
Accept: application/json
X-Fetch-Timeout: 15000
X-Fetch-Size: 1
X-Timestamp: 2024-07-19T15:36:48.032958023Z
Authorization: Bearer eyJ4NXQ...
Host: asrv:23432
```

### 7.2.7.2 Réponse HTTP

```
HTTP/1.1 204 No Content
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Request-ID: 0eecaf02-2301-4638-bb96-b67973c57943
X-Timestamp: 2024-07-19T15:36:48.037646935Z
X-Fetch-Count: 0
Server-Timing: acc;dur=2, acc-app;dur=0
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
```

## 7.2.8 Exemple de réponse d'erreur

### 7.2.8.1 Erreur du serveur

Le client reçoit une erreur avec le code 500 si une erreur du serveur se produit.

#### 7.2.8.1.1 Requête HTTP

```
GET /output/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
Accept: application/json
X-Fetch-Timeout: 15000
X-Fetch-Size: 1
X-Timestamp: 2024-07-19T15:36:48.073168283Z
Authorization: Bearer eyJ4NXQ...
Host: asrv:23432
```

#### 7.2.8.1.2 Réponse HTTP

```
HTTP/1.1 500 Internal Server Error
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
```

```
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Content-Length: 236
{
    "timestamp" : "2024-07-19T15:36:48.079+00:00",
    "status" : 500,
    "error" : "Internal Server Error",
    "message" : "Error while calling application",
    "path" : "/output/0eecaf02-2301-4638-bb96-b67973c57943",
    "errorCode" : "GE"
}
```

## 7.2.8.2 Erreur de mauvaise requête

Le client reçoit une erreur avec le code 400 si une "erreur de mauvaise requête" se produit.

### 7.2.8.2.1 Requête HTTP

```
GET /output/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
Accept: application/json
X-Fetch-Timeout: 15000
X-Fetch-Size: 1
X-Timestamp: 2024-07-19T15:36:47.300775258Z
Authorization: Bearer eyJ4NXQ...
Host: asrv:23432
```

### 7.2.8.2.2 Réponse HTTP

```
HTTP/1.1 400 Bad Request
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Content-Length: 225
{
    "timestamp" : "2024-07-19T15:36:47.341+00:00",
    "status" : 400,
    "error" : "Bad Request",
    "message" : "Wrong UserCode: CBOMOMRUWRPG",
    "path" : "/output/0eecaf02-2301-4638-bb96-b67973c57943",
    "errorCode" : "EA33"
}
```

### 7.2.8.3 Délai d'attente de récupération incorrect

Si l'en-tête HTTP X-Fetch-Timeout contient une valeur incorrecte, une réponse avec le code HTTP 400 sera renvoyée. L'exemple ci-dessous montre le cas où la valeur est inférieure au minimum de 5000 ms :

#### 7.2.8.3.1 Requête HTTP

```
GET /output/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
Accept: application/json
X-Fetch-Timeout: 4500
X-Fetch-Size: 1
X-Timestamp: 2024-07-19T15:36:47.411362200Z
Authorization: Bearer eyJ4NXQ...
Host: asrv:23432
```

#### 7.2.8.3.2 Réponse HTTP

```
HTTP/1.1 400 Bad Request
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Content-Length: 265
{
    "timestamp" : "2024-07-19T15:36:47.419+00:00",
    "status" : 400,
    "error" : "Bad Request",
    "message" : "Wrong data in field: Fetch timeout is less than min value
of 5000 ms",
    "path" : "/output/0eecaf02-2301-4638-bb96-b67973c57943",
    "errorCode" : "EA32"
}
```

L'exemple ci-dessous illustre le cas où la valeur dépasse le maximum. La valeur maximale est définie comme le produit du coefficient 0,8 et de la valeur de la configuration access-server.internalservices.httpClientProps.socketTimeout en ms (pour un socketTimeout par défaut de 60 ms, le maximum est de 48 ms ; pour un socketTimeout de 45 ms, le maximum est de 36 ms, etc.) :

### 7.2.8.3.3 Requête HTTP

```
GET /output/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
Accept: application/json
X-Fetch-Timeout: 48500
X-Fetch-Size: 1
X-Timestamp: 2024-07-19T15:36:47.954448098Z
Authorization: Bearer eyJ4NXQ...
Host: asrv:23432
```

### 7.2.8.3.4 Réponse HTTP

```
HTTP/1.1 400 Bad Request
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Content-Length: 269
{
    "timestamp" : "2024-07-19T15:36:47.959+00:00",
    "status" : 400,
    "error" : "Bad Request",
    "message" : "Wrong data in field: Fetch timeout is greater than max
value of 48000 ms",
    "path" : "/output/0eecaf02-2301-4638-bb96-b67973c57943",
    "errorCode" : "EA32"
}
```

### 7.2.8.4 Taille de récupération incorrecte

Si l'en-tête HTTP X-Fetch-Size contient une valeur incorrecte, une réponse avec le code HTTP 400 sera renvoyée. L'exemple ci-dessous illustre le cas où la valeur dépasse le maximum de 50 :

#### 7.2.8.4.1 Requête HTTP

```
GET /output/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
Accept: application/json
X-Fetch-Timeout: 15000
X-Fetch-Size: 51
X-Timestamp: 2024-07-19T15:36:47.916194909Z
Authorization: Bearer eyJ4NXQ...
Host: asrv:23432
```

### 7.2.8.4.2 Réponse HTTP

```
HTTP/1.1 400 Bad Request
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Content-Length: 260
{
    "timestamp" : "2024-07-19T15:36:47.920+00:00",
    "status" : 400,
    "error" : "Bad Request",
    "message" : "Wrong data in field: Fetch size is greater than max value
of 50",
    "path" : "/output/0eecaf02-2301-4638-bb96-b67973c57943",
    "errorCode" : "EA32"
}
```

### 7.2.8.5 Statut invalide

Si le client tente d'envoyer une demande parallèle avec un request\_id qui est déjà en cours de traitement, le client recevra une réponse HTTP 409 avec l'erreur "Statut invalide" et le code EP169.

#### 7.2.8.5.1 Requête HTTP

```
GET /output/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
Accept: application/json
X-Fetch-Timeout: 15000
X-Fetch-Size: 1
X-Timestamp: 2024-07-19T15:36:47.465552991Z
Authorization: Bearer eyJ4NXQ...
Host: asrv:23432
```

#### 7.2.8.5.2 Réponse HTTP

```
HTTP/1.1 409 Conflict
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
```

```
X-Frame-Options: DENY
Content-Length: 209
{
    "timestamp" : "2024-07-19T15:36:47.471+00:00",
    "status" : 409,
    "error" : "Conflict",
    "message" : "Invalid status",
    "path" : "/output/0eecaf02-2301-4638-bb96-b67973c57943",
    "errorCode" : "EP169"
}
```

## 7.2.8.6 Identifiant de demande incorrect

### 7.2.8.6.1 Requête HTTP

```
GET /output/%5E-%5E HTTP/1.1
Accept: application/json
X-Fetch-Timeout: 15000
X-Fetch-Size: 1
X-Timestamp: 2024-07-19T15:36:45.412648913Z
Authorization: Bearer eyJ4NXQ...
Host: asrv:23432
```

### 7.2.8.6.2 Réponse HTTP

```
HTTP/1.1 400 Bad Request
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Content-Length: 190
{
    "timestamp" : "2024-07-19T15:36:45.743+00:00",
    "status" : 400,
    "error" : "Bad Request",
    "message" : "RequestId has bad format",
    "path" : "/output/%5E-%5E",
    "errorCode" : "GE"
}
```

## 7.2.9 Codes de réponse HTTP

<b>Code</b>	<b>Description</b>	<b>Action</b>
200	Opération réussie	Envoyer une nouvelle demande
204	Aucun nouveau message à traiter	Envoyer une nouvelle demande
400	Mauvaise requête	Envoyer une nouvelle demande
401	Non autorisé	Procéder à l'autorisation  Obtenir un nouveau jeton  Répétez la requête
409	Conflit	Le client a envoyé une requête parallèle avec la même valeur <code>request_id</code> alors que le traitement de la requête précédente avec ce <code>request_id</code> n'est pas encore terminé.  Les requêtes parallèles avec le même <code>request_id</code> ne sont pas autorisées <a href="#">7.2.8.5 Invalid status</a> .
429	Trop de demandes (répéter la même demande plus tard) <a href="#">7.4 Rate limit</a> .	Répétez la même demande après que la valeur 'Retry-After' (secondes) soit reçue dans l'en-tête de réponse.
500	Internal server error (stop sending requests until resolved)	Repeat the same request until resolved. One request per second
503	Erreur interne du serveur (arrêter d'envoyer des requêtes jusqu'à ce qu'elle soit résolue)	Répétez la même demande jusqu'à ce qu'elle soit résolue. Une requête par seconde

### 7.2.10 Paramètres de réponse de réussite

<b>Path</b>	<b>Type</b>	<b>Requis</b>	<b>Description</b>
[ ]	Array	Vrai	<p>Le nombre de messages est spécifié dans le champ X-Fetch-Count.</p>
[ ].traceReference	String	FAUX	<p>ID de message généré par l'expéditeur.</p> <p>traceReference est généré pour le message initial dans un flux de messages ; tous les autres messages du flux doivent reproduire la traceReference initiale.</p> <p>Si une valeur vide est reçue, les messages suivants du flux doivent être envoyés (postMX) avec un nouveau traceReference.</p> <p>Les messages, qui sont générés par le système lui-même, c'est-à-dire certains rapports, notifications, etc. contiennent la valeur traceReference 'NOTPROVIDED'.</p>
[ ].type	String	Vrai	Type de message. pacs.002.001.09.
[ ].sender	String	Vrai	Code utilisateur participant de l'expéditeur, composé de 12 caractères. Ce code correspond au code participant dans le système avec la lettre « A » en 9ème position.

[].receiver	String	Vrai	Code participant du destinataire, composé de 12 caractères.	
[].document	String	Vrai	Texte du message, au format XML.	

### 7.2.11 Paramètres de réponse d'erreurs

Field	Type	Requis	Description
timestamp	String	Vrai	Horodatage de la réponse.
status	String	Vrai	Code d'état de la réponse HTTP.
errorCode	String	Vrai	Code erreur.
error	String	Vrai	Nom de l'erreur.
message	String	Vrai	Message décrivant l'erreur.
path	String	Vrai	Chemin de l'URL.

## 7.3 Méthode 'Post MX message'

### 7.3.1 Description de la méthode

Description	Demande d'envoi d'un message MX
Méthode	POST
Client	Application du participant

Serveur	Serveur d'accès
URL	<a href="https://&lt;access-server:access-server-port&gt;/input/{request_id}">https://&lt;access-server:access-server-port&gt;/input/{request_id}</a>

**Note :**

**Minifiez vos messages (au format XML) avant de les envoyer et de les signer.**

La minification XML consiste à supprimer les espaces inutiles, les sauts de ligne et d'autres éléments non essentiels des messages XML. Ce processus les rend plus compacts et réduit la bande passante nécessaire pour leur transfert. Par conséquent, la minification améliore les performances d'échange et de traitement des messages.

### 7.3.2 Variables d'URL

Variable	Description
request_id	Référence unique pour un message d'entrée. Elle doit comporter moins de 64 caractères et contenir uniquement les symboles autorisés, listés dans l'expression suivante :  [0-9a-zA-Z-_]{1,64}

### 7.3.3 En-têtes de requête

X-Timestamp	Horodatage de la demande, au format ISO 8601. Le Facultatif système ne valide pas ce paramètre mais l'audite.	
Content-type	Définir cet en-tête sur "application/json".	Requis
Accept	Définir cet en-tête sur "application/json".	Requis

Authorization	Contient le jeton d'accès utilisé pour authentifier le Requis user-agent auprès du serveur.	
Content-length	Taille du corps de la requête, en octets.	Facultatif
Host	Adresse et port de l'hôte.	Requis

### 7.3.4 Paramètres de requête

<b>Champ</b>	<b>Type</b>	<b>Requis</b>	<b>Description</b>
traceReference	String	Vrai	<p>Référence unique générée par l'expéditeur pour le message initial dans un flux.</p> <p>L'expéditeur du message initial dans le flux de messages doit générer traceReference. Tous les messages suivants dans le flux doivent reproduire la traceReference initiale reçue dans la méthode « Get MX message ».</p> <p>Par exemple, lors de l'envoi d'un message pacs.008, l'expéditeur génère une nouvelle traceReference. Lors de l'envoi d'un message pacs.002 en réponse au pacs.008, l'expéditeur utilise la traceReference du pacs.008 d'origine.</p> <p>Il est recommandé d'utiliser le GUID comme traceReference pour le message initial dans un flux de messages. Se référer à la RFC 4122.</p> <p>Le champ traceReference ne doit pas être utilisé par la logique de l'application (par exemple pour la liaison de messages).</p> <p>Messages provenant du système à relier par des balises XML.</p> <p>La valeur doit être inférieure à 64 symboles et contenir uniquement les symboles autorisés, listés dans l'expression suivante :</p> <pre>[0-9a-zA-Z/\-?:().,+]{1,64}</pre>
type	String	Vrai	Type de message.

sender	String	Vrai	<p>Code utilisateur participant de l'expéditeur, composé de 12 caractères.</p> <p>Il s'agit de l'identifiant utilisateur du Participant (basé sur le BIC du Participant) dans la méthode POST.</p>
receiver	String	Vrai	<p>Code participant du destinataire, composé de 12 caractères.</p> <p>Il s'agit du code identifiant du système (basé sur le code BIC du système avec « X » en 9ème position)</p>
document	String	Vrai	<p>Texte du message, au format XML.</p> <p>Ce paramètre ne doit pas commencer par la déclaration XML pour exclure le message d'erreur « Données erronées dans le champ » (EA32).</p> <p>Supprimez les espaces inutiles, les sauts de ligne et d'autres éléments non essentiels de l'XML pour minifier vos messages avant de les envoyer et de les signer.</p>

### 7.3.5 Requête HTTP

```

POST /input/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
X-Timestamp: 2018-08-13T12:15:54.651Z
Content-Type: application/json
Accept: application/json
Authorization: Bearer eyJ...iJ9.eyJ...jJ9.Ac-...0MSw
Content-Length: 5634
Host: asrv:23432

{
  "traceReference": "CKvOI85gv0SgNKqLAvBpwQ",
  "type": "pacs.008.001.08",
  "sender": "PARTIBICXUSR",
  "receiver": "SYSTEBICXUSR",
  "document": "<DataPDU ...>...</DataPDU>"
}
  
```

### 7.3.6 En-têtes de réponse

X-Request-ID	{request_id} identique à la requête ci-dessus. Il peut être utilisé pour relier une demande et une réponse.
X-Timestamp	Horodatage de la réponse, au format ISO 8601.
Server-Timing	Statistiques détaillées du temps serveur, passé à traiter la requête et à préparer la réponse.

La réponse peut également contenir un ensemble d'en-têtes techniques et de sécurité générés. Ces en-têtes ne sont pas répertoriés ici.

### 7.3.7 Exemple de réponse de réussite

#### 7.3.7.1 Requête HTTP

```
POST /input/0eeca02-2301-4638-bb96-b67973c57943 HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: application/json
X-Timestamp: 2024-07-19T15:36:52.246256721Z
Authorization: Bearer eyJ4NXQ...
Content-Length: 4089
Host: asrv:23432
{
  "traceReference" : "CKvOI85gv0SgNKqLAXBpwQ.0",
  "type" : "pacs.002.001.10",
  "sender" : "NRBLNPKAXOM1",
  "receiver" : "NRBLNPKAARTS",
  14
  "document" : "<DataPDU xmlns=\"urn:cma:stp:xsd:stp.1.0\">...</DataPDU>"
}
```

### 7.3.7.2 Réponse HTTP

```
HTTP/1.1 200 OK
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Request-ID: 0eeca02-2301-4638-bb96-b67973c57943
X-Timestamp: 2024-07-19T15:36:52.254065510Z
Server-Timing: app;dur=120, acc;dur=3, acc-app;dur=0
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
```

## 7.3.8 Exemple de réponse d'erreur

### 7.3.8.1 Bad request

A client receives an error with the code 400 (Bad request), if the request is incorrect.

#### 7.3.8.1.1 Requête HTTP

```
POST /input/0eeca02-2301-4638-bb96-b67973c57943 HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: application/json
X-Timestamp: 2024-07-19T15:36:52.548454275Z
Authorization: Bearer eyJ4NXQ...
Content-Length: 4089
Host: asrv:23432
{
    "traceReference" : "CKvOI85gv0SgNKqLAXBpwQ.0",
    "type" : "pacs.002.001.10",
    "sender" : "NRBLNPKAXOM1",
    "receiver" : "NRBLNPKAARTS",
    "document" : "<DataPDU xmlns=\"urn:cma:stp:xsd:stp.1.0\">...</DataPDU>"
}
```

### 7.3.8.1.2 Réponse HTTP

```
HTTP/1.1 400 Bad Request
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: In case the message contains a traceReference in an incorrect
format, the client will receive an error with code 400. The traceReference
field must match the regular expression [0-9a-zA-Z/\-?:()..+ ]{1,64},
meaning it should be no more than 64 characters and can contain the
specified characters in the regular expression.
```

### 7.3.8.2 Internal server error

The client receives an error with code 500 (Internal server error), if an internal server error occurs while receiving the message.

#### 7.3.8.2.1 Requête HTTP

```
POST /input/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: application/json
X-Timestamp: 2024-07-19T15:36:52.600902391Z
Authorization: Bearer eyJ4NXQ...
Content-Length: 4089
Host: asrv:23432
{
  "traceReference" : "CKvOI85gv0SgNKqLAXBpwQ.0",
  "type" : "pacs.002.001.10",
  "sender" : "NRBLNPKAXOM1",
  "receiver" : "NRBLNPKAARTS",
  "document" : "<DataPDU xmlns=\"urn:cma:stp:xsd:stp.1.0\">...</DataPDU>"
}
```

#### 7.3.8.2.2 Réponse HTTP

```
HTTP/1.1 500 Internal Server Error
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Content-Length: 227
{
  "timestamp" : "2024-07-19T15:36:52.609+00:00",
```

```
"status" : 500,  
"error" : "Internal Server Error",  
"message" : "No space left on device",  
"path" : "/input/0eecaf02-2301-4638-bb96-b67973c57943",  
"errorCode" : "GE"  
}
```

### 7.3.8.3 Erreur du serveur

Le client reçoit une erreur avec le code 500 si une erreur du serveur se produit.

#### 7.3.8.3.1 Requête HTTP

```
POST /input/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1  
Content-Type: application/json; charset=UTF-8  
Accept: application/json  
X-Timestamp: 2024-07-19T15:36:52.185033659Z  
Authorization: Bearer eyJ4NXQ...  
Content-Length: 4089  
Host: asrv:23432  
{  
    "traceReference" : "CKvOI85gv0SgNKqLAXBpwQ.0",  
    "type" : "pacs.002.001.10",  
    "sender" : "NRBLNPKAXOM1",  
    "receiver" : "NRBLNPKAARTS",  
    "document" : "<DataPDU xmlns=\\\"urn:cma:stp:xsd:stp.1.0\\\">...</DataPDU>"  
}
```

#### 7.3.8.3.2 Réponse HTTP

```
HTTP/1.1 503 Service Unavailable  
Vary: Origin  
Vary: Access-Control-Request-Method  
Vary: Access-Control-Request-Headers  
Content-Type: application/json  
22  
X-Content-Type-Options: nosniff  
X-XSS-Protection: 0  
Cache-Control: no-cache, no-store, max-age=0, must-revalidate  
Pragma: no-cache  
Expires: 0  
Strict-Transport-Security: max-age=31536000 ; includeSubDomains  
X-Frame-Options: DENY  
Content-Length: 228  
{  
    "timestamp" : "2024-07-19T15:36:52.194+00:00",  
    "status" : 503,  
    "error" : "Service Unavailable",  
    "message" : "Application is unavailable",  
    "path" : "/input/0eecaf02-2301-4638-bb96-b67973c57943",  
    "errorCode" : "GE"  
}
```

### 7.3.8.4 TraceReference incorrecte

Le client reçoit une erreur avec le code 400 (Référence de trace incorrecte) si le message contient une traceReference dans un format incorrect. Le champ traceReference doit correspondre à l'expression régulière [0-9a-zA-Z/-?:().,+]{1,64}, ce qui signifie qu'il ne doit pas dépasser 64 caractères et peut contenir uniquement les caractères spécifiés dans l'expression régulière.

#### 7.3.8.4.1 Requête HTTP

```
POST /input/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: application/json
X-Timestamp: 2024-07-19T15:36:52.654366523Z
Authorization: Bearer eyJ4NXQ...
Content-Length: 4115
Host: asrv:23432
{
    "traceReference" : "`select * from all-your_base-are_belong-to_us`<",
    "type" : "pacs.002.001.10",
    "sender" : "NRBLNPKAXOM1",
    "receiver" : "NRBLNPKAARTS",
    "document" : "<DataPDU xmlns=\"urn:cma:stp:xsd:stp.1.0\">...</DataPDU>"
}
```

#### 7.3.8.4.2 Réponse HTTP

```
HTTP/1.1 400 Bad Request
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Content-Length: 248
{
    "timestamp" : "2024-07-19T15:36:52.661+00:00",
    "status" : 400,
    "error" : "Bad Request",
    "message" : "Wrong data in field: Wrong symbols in traceReference",
    "path" : "/input/0eecaf02-2301-4638-bb96-b67973c57943",
    "errorCode" : "EA32"
}
```

### 7.3.8.5 Identifiant de demande incorrect

Le client reçoit une erreur avec le code 400 (Identifiant de demande incorrect) si la requête contient un request\_id dans un format incorrect. Le champ request\_id

doit correspondre à l'expression régulière [0-9a-zA-Z-.]{1,64}, ce qui signifie qu'il ne doit pas dépasser 64 caractères et peut contenir uniquement les caractères spécifiés dans l'expression régulière.

### 7.3.8.5.1 Requête HTTP

```
POST /input/%5E_%5E HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: application/json
X-Timestamp: 2024-07-19T15:36:51.990707919Z
Authorization: Bearer eyJ4NXQ...
Content-Length: 4089
Host: asrv:23432
{
  "traceReference" : "CKvOI85gv0SgNKqLAXBpwQ.0",
  "type" : "pacs.002.001.10",
  "sender" : "NRBLNPKAXOM1",
  "receiver" : "NRBLNPKAARTS",
  "document" : "<DataPDU xmlns=\"urn:cma:stp:xsd:stp.1.0\">...</DataPDU>"
}
```

### 7.3.8.5.2 Réponse HTTP

```
HTTP/1.1 400 Bad Request
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Content-Length: 189
{
  "timestamp" : "2024-07-19T15:36:52.042+00:00",
  "status" : 400,
  "error" : "Bad Request",
  "message" : "RequestId has bad format",
  "path" : "/input/%5E_%5E",
  "errorCode" : "GE"
}
```

### 7.3.8.6 Message dupliqué

Le client reçoit une réponse HTTP 400 avec l'erreur "Message dupliqué" et le code EA5, si la requête contient un request\_id qui a déjà été utilisé pour envoyer un autre message.

### 7.3.8.6.1 Requête HTTP

```
POST /input/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: application/json
X-Timestamp: 2024-07-19T15:36:52.391271624Z
Authorization: Bearer eyJ4NXQ...
Content-Length: 4089
Host: asrv:23432
{
  "traceReference" : "CKvOI85gv0SgNKqLAXBpwQ.0",
  "type" : "pacs.002.001.10",
  "sender" : "NRBLNPKAXOM1",
  "receiver" : "NRBLNPKAARTS",
  "document" : "<DataPDU xmlns=\"urn:cma:stp:xsd:stp.1.0\">...</DataPDU>"
}
```

### 7.3.8.6.2 HTTP response

```
HTTP/1.1 400 Bad Request
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Content-Length: 216
{
  "timestamp" : "2024-07-19T15:36:52.398+00:00",
  "status" : 400,
  "error" : "Bad Request",
  "message" : "Message is duplicated",
  "path" : "/input/0eecaf02-2301-4638-bb96-b67973c57943",
  "errorCode" : "EA5"
}
```

### 7.3.8.7 XML non autorisé (400)

Le client recevra une erreur 400 "Données incorrectes dans le champ" avec le code EA32, si le document commence par une déclaration XML.

### 7.3.8.7.1 Requête HTTP

```
POST /input/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: application/json
X-Timestamp: 2024-07-19T15:36:52.449621694Z
Authorization: Bearer eyJ4NXQ...
Content-Length: 4149
Host: asrv:23432
{
  "traceReference" : "CKvOI85gv0SgNKqLAXBpwQ.0",
  "type" : "pacs.002.001.10",
  "sender" : "NRBLNPKAXOM1",
  "receiver" : "NRBLNPKAARTS",
  "document" : "<?xml version=\"1.0\" encoding=\"UTF-8\" standalone=\"no\"?><DataPDU
  xmlns=\"urn:cma:stp:xsd:stp.1.0\">...</DataPDU>"
}
```

### 7.3.8.7.2 HTTP response

```
HTTP/1.1 400 Bad Request
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Content-Length: 292
{
  "timestamp" : "2024-07-19T15:36:52.501+00:00",
  "status" : 400,
  "error" : "Bad Request",
  "message" : "Wrong data in field: The processing instruction target matching \"[xX] [mM] [lL]\" is not allowed.",
  "path" : "/input/0eecaf02-2301-4638-bb96-b67973c57943",
  "errorCode" : "EA32"
}
```

### 7.3.8.8 Validation de schéma XSD

S'il y a des erreurs de validation du document entrant par rapport au schéma XSD, le serveur d'accès envoie immédiatement une erreur de validation au client. Le client reçoit une erreur 400 "Le bloc de texte a un format invalide" avec le code EA1.

### 7.3.8.8.1 Requête HTTP

```
POST /input/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: application/json
X-Timestamp: 2024-07-19T15:36:52.703604573Z
Authorization: Bearer eyJ4NXQ...
Content-Length: 4089
Host: asrv:23432
{
  "traceReference" : "CKvOI85gv0SgNKqLAXBpwQ.0",
  "type" : "pacs.002.001.10",
  "sender" : "NRBLNPKAXOM1",
  "receiver" : "NRBLNPKAARTS",
  "document" : "<DataPDU xmlns=\"urn:cma:stp:xsd:stp.1.0\">...</DataPDU>"
}
```

### 7.3.8.8.2 Réponse HTTP

```
HTTP/1.1 400 Bad Request
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Content-Length: 497
{
  "timestamp" : "2024-07-19T15:36:52.710+00:00",
  "status" : 400,
  "error" : "Bad Request",
  "message" : "Text block has invalid format: Invalid MX input: \n [80, 32]
ext-rulecheck:
/DataPDU/Body/Document/FIToFICstmrCdtTrft:FIToFICustomerCreditTransferV08\n
\tFalseBatchBookingLocalInstrumentRule: For single instruction payments
Local Instrument must be equal to CTAA, CTWA, CTAW, CTWW, PMCT, MSP2M or
M2MSP",
  "path" : "/input/0eecaf02-2301-4638-bb96-b67973c57943",
  "errorCode" : "EA1"
}
```

### 7.3.8.9 Validation de la signature

S'il y a des erreurs de validation de la signature du document entrant, le serveur d'accès envoie immédiatement une erreur de vérification de signature au client. Le client reçoit une erreur 400 "Échec de la vérification de la signature" avec le code SC312.

### 7.3.8.9.1 Requête HTTP

```
POST /input/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: application/json
X-Timestamp: 2024-07-19T15:36:52.120212951Z
Authorization: Bearer eyJ4NXQ...
Content-Length: 4089
Host: asrv:23432
{
  "traceReference" : "CKvOI85gv0SgNKqLAXBpwQ.0",
  "type" : "pacs.002.001.10",
  "sender" : "NRBLNPKAXOM1",
  "receiver" : "NRBLNPKAARTS",
  "document" : "<DataPDU xmlns=\"urn:cma:stp:xsd:stp.1.0\">...</DataPDU>"
}
```

### 7.3.8.9.2 Réponse HTTP

```
HTTP/1.1 400 Bad Request
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Content-Length: 254
{
  "timestamp" : "2024-07-19T15:36:52.130+00:00",
  "status" : 400,
  "error" : "Bad Request",
  "message" : "Signature check fail: Signature of MX Document is missing",
  "path" : "/input/0eecaf02-2301-4638-bb96-b67973c57943",
  "errorCode" : "SC312"
}
```

## 7.3.9 Codes de réponse HTTP

<b>Code Description</b>	<b>Action</b>
200 Opération réussie	Envoyer le prochain message

400	Mauvaise requête	Supprimer le message, Envoyer le prochain message
401	Non autorisé	Procéder à l'autorisation Obtenir un nouveau jeton Répéter la même requête
429	Trop de demandes (envoyer la même demande plus tard)	Répéter la même demande après que la valeur 'Retry-After' (secondes) soit reçue dans l'en-tête de réponse.
500	Erreur interne du serveur (arrêter d'envoyer des requêtes jusqu'à ce qu'elle soit résolue)	Répétez la même demande jusqu'à ce qu'elle soit résolue. Une requête par seconde
503	Le service n'est pas disponible (arrêtez d'envoyer des demandes jusqu'à ce qu'ils soient résolus)	Répétez la même demande jusqu'à ce qu'elle soit résolue. Une requête par seconde

### 7.3.10 Paramètres de réponse aux erreurs

Field	Type	Requis	Description
timestamp	String	Vrai	Horodatage de la réponse.
status	String	Vrai	Code d'état de la réponse HTTP.
errorCode	String	Vrai	Code d'erreur.
error	String	Vrai	Nom de l'erreur.
message	String	Vrai	Message décrivant la réponse.

path	String	Vrai	Chemin de l'URL.
------	--------	------	------------------

## 7.4 Limitation du nombre de requêtes

Pour limiter le nombre de requêtes par les participants, les paramètres suivants peuvent être utilisés : API\_RATE\_LIMIT.<URL>\_PER\_MINUTE and API\_RATE\_LIMIT.<URL>\_PER\_HOUR, where <URL> est l'URL de la requête, réduite en majuscules, et où les caractères / sont remplacés par \_.

Exemples de paramètres :

- API\_RATE\_LIMIT.OUTPUT\_PER\_MINUTE
- API\_RATE\_LIMIT.OUTPUT\_PER\_HOUR
- API\_RATE\_LIMIT.INPUT\_PER\_MINUTE
- API\_RATE\_LIMIT.INPUT\_PER\_HOUR

### 7.4.1 Nombre de requêtes dépassée

Si la limite est dépassée, l'erreur 429 sera renvoyée. La réponse peut également contenir l'en-tête Retry-After, qui indique après combien de secondes l'utilisateur peut réessayer.

#### 7.4.1.1 Requête HTTP

```
POST /input/0eecaf02-2301-4638-bb96-b67973c57943 HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: application/json
X-Timestamp: 2024-07-19T15:36:52.339174812Z
Authorization: Bearer
eyJ4NXQiOj0cHNI0cDVRNS0zM3pRRG5KQkhJeG10TzgiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJDQk9NT01SVVdsUEciLCJhc3J2X3R5cGUioiJhY2N1c3MiLCJleHAiOjQ3NDA5MjI1MDYsImhdCI6MTYxODg1ODUwNiwianRpIjoiYmZyeGJSa0hSei1QVk5LvvFoTHh1dyJ9.j0k6cZtGYb41DoTipYxgWcHQugEkG1dT2SE43c4jNUQFHgCfCa09d8hjJQp09bzG14wCuj_asNR7j-kQSwt0g4doMwcxGVsQRnhStnrmvlGGnGxGGcdAb5k_PJDM9MPOYe2tXKo0Cg0YIBeEUyrlwU5AnoVGoBLP75vV9x
iTwpS_61_PHSUy8q6X-LqHYOVnduDTR5uP0ViK0C9RHsn3lNWCw-fhXxkbGpF7aJJRKAGhFaR_wzdFhjpYgqCP605wkYRN5n8FoCmyyiHODE8V3ThV5imYRmWoJ1Yrm
o8z_roy4VB9HUHw44YRQXma4YfIfSGkvSq96I13JUb0hkA
Content-Length: 4089
Host: asrv:23432
{
  "traceReference" : "CKvOI85gv0SgNKqLAxBpwQ.0",
}
```

## Spécifications API REST de Messagerie

```
"type" : "pacs.002.001.10",
"sender" : "NRBLNPKAXOM1",
"receiver" : "NRBLNPKAARTS",
"document" : "<DataPDU xmlns=\\"urn:cma:stp:xsd:stp.1.0\\"><Body><AppHdr

xmlns=\\"urn:iso:std:iso:20022:tech:xsd:head.001.001.01\\"><Fr><FIId><FinInstnId><ClrSys

MmbId><MmbId>AZAZPKPK</MmbId></ClrSysMmbId></FinInstnId></FIId></Fr><To><FI

Id><FinInstn

nId><ClrSysMmbId><MmbId>SBPPPKAXIPS</MmbId></ClrSysMmbId></FinInstnId></FI

Id></To><Bi

zMsgIdr>AZAZPKPK8823180921.951</BizMsgIdr><MsgDefIdr>pacs.002.001.10</MsgDe

fIdr><BizSv

c>ACH</BizSvc><CreDt>2018-09-
28T18:47:04.101Z</CreDt><Sgntr><ds:Signature

xmlns:ds=\\"http://www.w3.org/2000/09/xmldsig#\\\" Id=\\"_75013d0b-c787-4d65-b038-
2a3c6241e8f7\\\"><ds:SignedInfo><ds:CanonicalizationMethod

Algorithm=\\"http://www.w3.org/2001/10/xml-exc-c14n#\\"/><ds:SignatureMethod

Algorithm=\\"http://www.w3.org/2001/04/xmldsig-more#rsa-sha256\\\"/><ds:Reference

URI=\\"#_f1311cda-ac0c-4941-b605-0dca172599f7-ki\\\"><ds:Transforms><ds:Transform

Algorithm=\\"http://www.w3.org/2001/10/xml-exc-c14n#\\"/></ds:Transforms><ds:DigestMethod

Algorithm=\\"http://www.w3.org/2001/04/xmlenc#sha256\\\"/><ds:DigestValue>Mmq19UU7naR3KTM

7OXN7+0wi+Efx4gz7zjEcCoKei/E=</ds:DigestValue></ds:Reference><ds:Reference

Type=\\"http://uri.etsi.org/01903/v1.3.2#SignedProperties\\"
URI=\\"#_6eac1ad9-7f5e-402f8e3f-186e969d8ba8-sp\\\"><ds:Transforms><ds:Transform

Algorithm=\\"http://www.w3.org/2001/10/xml-exc-c14n#\\"/></ds:Transforms><ds:DigestMethod

Algorithm=\\"http://www.w3.org/2001/04/xmlenc#sha256\\\"/><ds:DigestValue>WfE5si5Q8AhfOn8

+DYHxVfd0aHQJks8DHmLGP8KP/eQ=</ds:DigestValue></ds:Reference><ds:Reference>

<ds:Transforms><ds:Transform Algorithm=\\"http://www.w3.org/2001/10/xml-exc-c14n#\\"/></ds:Transforms><ds:DigestMethod

Algorithm=\\"http://www.w3.org/2001/04/xmlenc#sha256\\\"/><ds:DigestValue>sZMagsbXl7o93J1g

mkMmbJSTH8NEyXKt3DetnGgGaTcI=</ds:DigestValue></ds:Reference></ds:SignedInfo><ds:Signa

tureValue>AMy4fY+rI3sseHR+PATvz32g58BaSBSvjDz/B7091DxkFhDzmdK+7zOoeWG4ZHEX/
```

Spécifications API REST de Messagerie

QwVV1s0KvH8

&#13; \nX40so8eCQbG8ZLBdPuaI5kpCuFjIK0dmTWb+d4WbRcTgqjiUQ4ZaxQP/OsjFgciWxnJ6  
AwYia82g&#13;

3; \nNtRwl8OKCLGKyh1UKVepHWE8MBv2s2NhUxwc5YYOxeeRlKOlnHNY/k2fszIGTemkdSecywy  
K7X/i&#13; ;\n

nRUzKPVj6007HIQnGdYs/sVnLVHS2Sy3/Zfx50m7+YV2QL2EE0frn8gE+TTKbMHTAf10i65zv4r  
kn&#13; ;\nFY  
wQn9ZqqhlU56EZyDGGVFFD9sUA1zgILkTPeQ==</ds:SignatureValue><ds:KeyInfo  
Id=\\"\_f1311cdaac0c-4941-b605-0dca172599f7-  
ki\\"><ds:X509Data><ds:X509IssuerSerial><ds:X509IssuerName>CN=access-  
server-test,  
O=cma,

C=ru</ds:X509IssuerName><ds:X509SerialNumber>1565363905</ds:X509SerialNumbe  
r></ds:X509

IssuerSerial></ds:X509Data></ds:KeyInfo><ds:Object><xades:QualifyingPropert  
ies  
xmlns:xades=\\"http://uri.etsi.org/01903/v1.3.2#\\"  
Target=\\"#\_75013d0b-c787-4d65-b038-  
2a3c6241e8f7\\><xades:SignedProperties Id=\\"\_6eac1ad9-7f5e-402f-8e3f-  
186e969d8ba8-  
sp\\><xades:SignedSignatureProperties><xades:SigningTime>2024-07-

19T18:36:52+03:00</xades:SigningTime></xades:SignedSignatureProperties></xa  
des:SignedP

roperties></xades:QualifyingProperties></ds:Object></ds:Signature></Sgntr><  
/AppHdr><Do  
cument

xmlns=\\"urn:iso:std:iso:20022:tech:xsd:pacs.002.001.10\\><FIToFIPmtStsRpt><  
GrpHdr><Msg  
Id>AZAZPKPK8823180921.951</MsgId><CreDtTm>2017-09-

13T18:18:00</CreDtTm><InstgAgt><FinInstnId><ClrSysMmbId><MmbId>AZAZPKPK</M  
bId></ClrSy

sMmbId></FinInstnId></InstgAgt><InstdAgt><FinInstnId><ClrSysMmbId><MmbId>NI  
BPPKKA</Mmb

Id></ClrSysMmbId></FinInstnId></InstdAgt></GrpHdr><OrgnlGrpInfAndSts><Orgnl  
MsgId>NIBPP

KKAA9000180919.609</OrgnlMsgId><OrgnlMsgNmId>pacs.008.001.08</OrgnlMsgNmId><  
OrgnlCreDtT  
m>2017-09-

13T18:18:00</OrgnlCreDtTm><StsRsnInf><Rsn><Prtry>AUTH</Prtry></Rsn></StsRsn  
Inf></Orgnl

GrpInfAndSts><TxInfAndSts><OrgnlInstrId>NIBPPKKA9000180919.609</OrgnlInstrI

```
d><OrgnlEnd
    ToEndId>3d67e297-c2a7-1-
afad05e458c6c51</OrgnlEndToEndId><OrgnlTxId>NIBPPKKA9000180919.609</OrgnlTx
Id><StsRsnInf><
Rsn><Prtry>AUTH</Prtry></Rsn></StsRsnInf><InstgAgt><FinInstnId><ClrSysMmbId
><MmbId>NIB
PPKKA</MmbId></ClrSysMmbId></FinInstnId></InstgAgt><OrgnlTxRef><IntrBkSttlm
Dt>2021-03-
26</IntrBkSttlmDt></OrgnlTxRef></TxInfAndSts></FIToFIPmtStsRpt></Document><
/Body></Dat
    aPDU>""
}
```

#### 7.4.1.2 Réponse HTTP

```
HTTP/1.1 429 Too Many Requests
Retry-After: 59
```

## 8 Référence API (à partir de YAML)