

Gestion des transactions : Capture / Annulation / Remboursement

Cette page décrit les opérations possibles sur une transaction créée via Axepta BNP Paribas Online :

- Capture (automatique, différée, manuelle, partielle)
- Annulation (avant capture)
- Remboursement (après capture)

Contenu

- [Cas d'usage : Autorisation, Capture, Annulation, Remboursement](#)
 - [Autorisation](#)
 - [Capture](#)
 - [Annulation](#)
 - [Remboursement](#)
- [Capture](#)
 - [Capture automatique \(mode par défaut\)](#)
 - [Capture automatique différée \(DELAYED\)](#)
 - [Capture manuelle \(MANUAL\)](#)
- [Annulation](#)
- [Remboursement](#)
- [Dois-je annuler ou rembourser ma transaction ?](#)
 - [Etape 1: Consulter le statut de la transaction](#)
 - [Etape 2: Consulter les montants de la transaction](#)
 - [Etape 3a: Remboursement](#)
 - [Etape 3b: Annulation](#)

Cas d'usage : Autorisation, Capture, Annulation, Remboursement

Autorisation

Lorsqu'une transaction carte est créée, une autorisation est envoyée à la banque du porteur pour réserver le montant.

Tant que la capture n'est pas effectuée, les fonds ne sont pas débités.

Capture

La capture correspond à la demande d'envoi en remise bancaire.

C'est l'étape qui entraîne le débit réel du porteur.

Annulation

Une annulation est possible tant que la capture n'a pas été effectuée.

Elle libère immédiatement les fonds réservés.

Remboursement

Un remboursement s'effectue sur une transaction déjà capturée.

Il peut être total ou partiel, simple ou multiple.

Capture

Lors de l'initialisation de la transaction, vous pouvez contrôler le comportement via l'objet `captureMethod`.

```
"captureMethod": { "type": "AUTOMATIC" | "DELAYED" | "MANUAL" }
```



Si captureMethod est absent capture automatique par défaut.

Capture automatique (mode par défaut)

Fonctionnel

La demande de capture est effectuée automatiquement juste après l'autorisation.

Il n'y a aucune action à effectuer, la transaction partira en remise lors du prochain cycle d'encaissement (fin de journée).



Sauf si vous avez des cas d'usage particulier, c'est le mode à privilégier.

Cas d'usage recommandés

- Paiement classique e-commerce
- Processus sans gestion des stocks complexes
- Marchands n'ayant pas besoin de vérifier la disponibilité avant capture

Capture automatique différée (DELAYED)

Fonctionnel

Lors de la demande de transaction, il est possible préciser un délai (en heures) avant l'exécution de la demande de capture.

Cette option ouvre la possibilité de garantir que la transaction ne sera pas capturée pendant ce délai.

La remise peut être décalée d'une journée si le délai dépasse l'heure de remise quotidienne.



Cette capture doit être réalisée dans les 7 jours suivants l'autorisation, en accord avec le délai de garanti des fonds en banque.

Caractéristiques

- Le délai garantit qu'aucune capture n'aura lieu pendant X heures.
- Permet d'annuler la transaction facilement avant capture.
- Peut décaler la remise bancaire d'une journée si le délai dépasse l'heure de remise.

Cas d'usage

- Vérification de stock différée
- Vérifications manuelles avant expédition
- Anti-fraude nécessitant un délai avant capture

Intégration

L'endpoint [Capture](#) permet de capturer les fonds en utilisant le payId de la transaction d'origine.

(Montant optionnel capture partielle si différent du montant autorisé.)

captureMethod

```
...
    "captureMethod": {
      "type": "DELAYED",
      "delayed": {
        "delayedHours": 2
      }
    },
...
```

Capture manuelle (MANUAL)

Fonctionnel

La capture manuelle ouvre la possibilité de capture tout ou une partie seulement du montant autorisée.

Dans le cas où une partie du montant est capturé, on parle de capture partielle.

Donc, avec cette option, la capture n'est jamais exécutée automatiquement, il est impératif d'envoyer un ordre de capture pour que les fonds soient encaissés.

La date de remise bancaire de ces transactions sera celle de la demande de capture.



Cette capture doit être réalisée dans les 7 jours suivants l'autorisation, en accord avec le délai de garanti des fonds en banque.

Intégration

L'endpoint [Capture](#) permet de capturer les fonds en utilisant le payId de la transaction d'origine.

(Montant optionnel capture partielle si différent du montant autorisé.)

captureMethod

```
...
    "captureMethod": {
      "type": "MANUAL"
    },
...
```

Annulation

Fonctionnel

- L'annulation est possible uniquement si la transaction n'a pas été capturée (`capturedValue = 0`).
- Elle permet de libérer les fonds réservés sur la carte du client.

 L'annulation ne peut pas être partielle. Le montant annulé = montant autorisé.

Intégration

L'endpoint [Reverse](#) permet d'annuler une transaction (ou une étape de la transaction). Il est nécessaire de fournir le `payId` de la transaction d'origine.

 Le statut influence le nombre d'appels nécessaires.

Remboursement

Fonctionnel

Seule une transaction déjà capturée (`capturedValue > 0`) ou réglée (statut `Sale`) peut être remboursée.

Le remboursement peut-être total ou partiel.

Plusieurs remboursements successifs possibles jusqu'au montant capturé.

Intégration

L'endpoint [Refund](#) permet d'initier un remboursement sur une transaction déjà réglée. Il est nécessaire de fournir le `payId` de la transaction à rembourser.

Dois-je annuler ou rembourser ma transaction ?

Dans le cadre de la gestion des transactions, il est essentiel de distinguer clairement les situations nécessitant l'annulation d'un paiement de celles requérant un remboursement.

Cette section décrit les étapes clés pour déterminer la meilleure action à entreprendre lorsque une commande ne peut être honorée.

Que ce soit pour des raisons logistiques, techniques ou autres, comprendre le statut du paiement et les montants impliqués est crucial.

En vérifiant ces éléments, il sera possible de déterminer si le paiement doit être annulé (lorsque les fonds n'ont pas été réclamés par le porteur) ou remboursé (lorsque les fonds ont été effectivement réclamés). Cette démarche assure une gestion transparente et conforme aux attentes des clients, tout en optimisant les processus opérationnels.

 Si la transaction a été payée, alors le client a été débité. Dans ce cas, un remboursement doit être initié.
Si la transaction n'a pas été payée, alors le client n'a été débité. Dans ce cas, une annulation doit être initiée.

 Pour l'ensemble des opérations, il est nécessaire d'utiliser la référence de paiement '`payId`' retournée par la plateforme de paiement lors de la transaction.

Etape 1: Consulter le statut de la transaction

L'endpoint [GetByPayId](#) permet de récupérer les informations de la transaction en fournissant le `payId` correspondant.

Exemple réponse getByPayId

```
{
  "amount": {
    "currency": "EUR",
    "value": 1000,
    "capturedValue": 1000
  },
  "externalIntegrationId": "12345-abcde-67890-fghij",
  "language": "en",
  "payId": "ed7773c09f314c45b2a0de47a1994e1d",
  "xId": "86a8198cc9da4b4d9fbcdb15a5ddd6ca",
  "transId": "20240103-121913-554",
  "status": "OK",
  "responseCode": "00000000",
  "responseDescription": "success",
  "refNr": "refNr01",
  "metadata": {
    "userData": "my user data",
    "plain": "some plain text",
    "key1": "value1",
    "key2": "value2"
  },
  "paymentMethods": {
    "type": "CARD",
    "card": { ... }
  }
}
```

Afin de déterminer s'il est nécessaire d'annuler ou rembourser une transaction, il est nécessaire de consulter les propriétés '**status**' et '**responseCode**' de la transaction:

- la propriété '**status**' doit être égale à "OK" ou "AUTHORIZED" (à garder si besoin de faire une annulation)
- la propriété '**responseCode**' doit être égale à "00000000".
 - Se référer à [Codes réponse / Error codes \(1\)](#) pour le détail des codes réponse

```
"status": "OK",
"responseCode": "00000000",
```

Etape 2: Consulter les montants de la transaction

Les montants de la transaction sont détaillé dans l'objet JSON '**amount**'.

Objet 'amount'

```
"amount": {
  "currency": "EUR",
  "value": 1000,
  "capturedValue": 1000
},
```

Il est nécessaire de consulter les montants '**value**' et '**capturedValue**'

- **'value'** : Montant autorisé. il doit être supérieur à zéro.
S'il est égale à zéro, c'est que l'autorisation de la transaction à échoué, et que la transaction n'a pas aboutie.
- **'capturedValue'**: Montant de la transaction capturée.



'capturedValue' va déterminer si un remboursement ou une annulation doit être initiée.



Si capturedValue = 0 : la transaction n'est pas capturée => Annulation
Si capturedValue > 0 : la transaction est capturée => Remboursement

Etape 3a: Remboursement

L'endpoint [Refund](#) permet d'initier un remboursement sur une transaction déjà réglée. Il est nécessaire de fournir le payId de la transaction à rembourser.



Le montant à rembourser peut être tout ou une partie du montant de la transaction initiale.
On parle alors de remboursement partiel.
Il est possible de faire une succession de remboursement partiel jusqu'à concurrence du montant autorisé.

Etape 3b: Annulation

L'endpoint [Reverse](#) permet d'annuler une transaction (ou une étape de la transaction). Il est nécessaire de fournir le payId de la transaction d'origine.



Le montant de l'annulation doit être exactement égale au montant autorisé.
Il n'est pas possible de faire des annulations partielles.



L'annulation peut se faire en 1 ou 2 étapes, suivant si la demande de capture a déjà été faite ou pas.

L'endpoint [Reverse](#) permet de:

- Annuler la demande de capture et revenir à l'étape 'Authorized', si la transaction est en attente de capture.
- Annuler la demande d'autorisation, et définitivement annuler l'ensemble de la transaction, si la transaction est autorisée et que la demande de capture n'a pas été faite.

Pour savoir à quelle étape est la transaction, il faut regarder son 'status' :

- **"OK"**
La capture a été demandée mais n'est pas encore faite.
La transaction est en attente de capture.
L'appel à l'api de reversal va annuler la demande de capture, et passer le statut de la transaction à "AUTHORIZED".
- **"AUTHORIZED"**
La demande de capture n'a pas été faite (ou a été annulée).
L'appel à l'api de reversal va annuler l'autorisation.
Les fonds réservés lors de l'autorisation vont être libérés.
La transaction est définitivement annulée.



En conclusion:

- Pour annuler une transaction avec le statut égal à **"OK"**, 2 appels au endpoint [Reverse](#) sont nécessaires
 - un premier appel pour annuler la demande de capture, et un second pour annuler l'autorisation
- Pour annuler une transaction avec le statut égal à **"AUTHORIZED"**, 1 appel au endpoint [Reverse](#)
 - un unique appel pour annuler l'autorisation