



Hosted Fields

Documentation d'intégration



SOMMAIRE

1.	<i>Introduction</i>	3
2	<i>Documentation de référence</i>	4
3	<i>Prérequis</i>	5
3.1	Configuration du TPEv (Terminal de Paiement Electronique virtuel)	5
3.2	Parcours d'enregistrement et de réutilisation de carte	5
4	<i>Scenario</i>	6
5	<i>Cas d'utilisation et parcours</i>	8
5.1	Demande de paiement et demande de renseignement	8
5.2	Enregistrement et réutilisation de carte	8
6	<i>Intégration</i>	9
6.1	Etape 1 : intégration de l'API « PaymentMeanTokenGenerator » (Back-end)	9
6.2	Etape 2 : intégration du SDK « hostedfields.js » (Front-end)	9
6.3	Etape 3 : intégration de l'API « PaymentService » (Back-End)	10
6.3.1	Appel au service	10
6.3.2	Retour du service	11
6.3.3	Exemples d'appel et de retour du service	13
7	<i>Environnements</i>	24
7.1	Environnement de test dit « sandbox »	24
7.2	Environnement d'exploitation dit « live »	25

1. Introduction

Le but du système *Hosted Fields* de Monetico Paiement est de permettre aux commerçants de générer au sein de leur site e-commerce des parcours de paiement et/ou d'enregistrement de carte de paiement qui soient parfaitement intégrés à leur site et sécurisés.

Un *hosted field* est un composant web indépendant permettant de saisir une seule information du moyen de paiement que le consommateur souhaite utiliser pour effectuer son ou ses achats. Il existe en tout quatre types basiques de *hosted fields* correspondant chacun aux éléments suivants d'un moyen de paiement :

- le numéro de la carte de paiement ou *Personal Account Number (PAN)*
- la date d'expiration de la carte de paiement
- le cryptogramme visuel de la carte de paiement
- le nom du porteur de la carte de paiement

Lors de la mise en place d'un parcours utilisant les *Hosted Fields*, le commerçant a la charge de générer sur sa ou ses pages web l'ensemble des *hosted fields* qui permettront à son client de saisir son moyen de paiement.

Pour chaque parcours *Hosted Fields* instancié, le commerçant a également la charge d'effectuer au préalable auprès de Monetico Paiement une demande de génération d'identifiant unique appelé « jeton de saisie temporaire ».

Cet identifiant permet à Monetico Paiement de faire le lien entre les différents *hosted fields* générés par le commerçant pour un seul et même parcours.

Une fois les champs instanciés sur la page web du commerçant, Monetico Paiement se charge de sauvegarder temporairement les données sensibles saisies à l'intérieur des champs et de les associer à ce jeton sans action particulière du commerçant.

Lorsque la saisie est terminée, Monetico Paiement notifie le commerçant de cet événement. Celui-ci peut alors effectuer une demande de paiement ou d'enregistrement de carte auprès de Monetico Paiement en indiquant la valeur du jeton de saisie temporaire. Monetico Paiement récupère alors les données sensibles du moyen de paiement enregistrées plus tôt lors de la saisie du moyen de paiement par le consommateur via les *hosted fields*.

La suite du parcours se déroule comme un parcours de paiement ou d'enregistrement de carte de paiement par API classique.

Une fois le parcours finalisé, le jeton de saisie temporaire est considéré comme consommé et n'est plus utilisable pour une autre demande de paiement ou d'enregistrement de carte. La notion d'enregistrement de la carte en fin de parcours est associée à un autre identifiant appelé « identifiant de *wallet* » qui lui sera réutilisable pour des parcours de paiement ultérieurs dans le cas où l'enregistrement a été effectué avec succès.

Dans le cas particulier d'un parcours de paiement avec réutilisation d'une carte enregistrée, Monetico Paiement met à disposition deux types de champs *hosted field* supplémentaires :

- un champ de restitution des informations de la carte enregistrée :
 - le numéro de la carte de paiement au format PCIDSS (ex : 497783XXXXXX0005)
 - la date d'expiration de la carte de paiement
 - le nom du porteur de la carte de paiement
- un champ équivalent à un bouton permettant au porteur de saisir une autre de carte pour effectuer son paiement que celle enregistrée

2 Documentation de référence

[1] [Documentation technique « Monetico Paiement – Demande de paiement par API », version 1.0, septembre 2023. Euro Information.](#)

[2] [Documentation technique « Monetico Paiement – Demande de jeton de saisie temporaire de moyen de paiement », version 3, décembre 2025. Euro Information.](#)

[3] [Documentation technique « Monetico Paiement – SDK Hosted Fields », version 3, décembre 2025. Euro Information.](#)

3 Prérequis

3.1 Configuration du TPEv (Terminal de Paiement Electronique virtuel)

L'intégration du système *Hosted Fields* de Monetico Paiement nécessite d'avoir souscrit au préalable à un pack « expert » Monetico.

Une fois la souscription finalisée, l'intégrateur doit être en possession des informations techniques suivantes :

- le numéro de contrat TPEv (Terminal de Paiement Electronique virtuel)
- la clé de sécurité du TPEv
- le nom du « code site » ou « code société » associé à la **configuration pour les appels aux API** de Monetico Paiement (par opposition à la configuration dédiée à l'utilisation de la page de paiement déportée de Monetico)
- le code langue associé au « code site » décrit ci-dessus

Une demande doit également être faite auprès du support technique pour activer l'option « **tokenisation** » sur le TPEv et le « code site » utilisés.

3.2 Parcours d'enregistrement et de réutilisation de carte

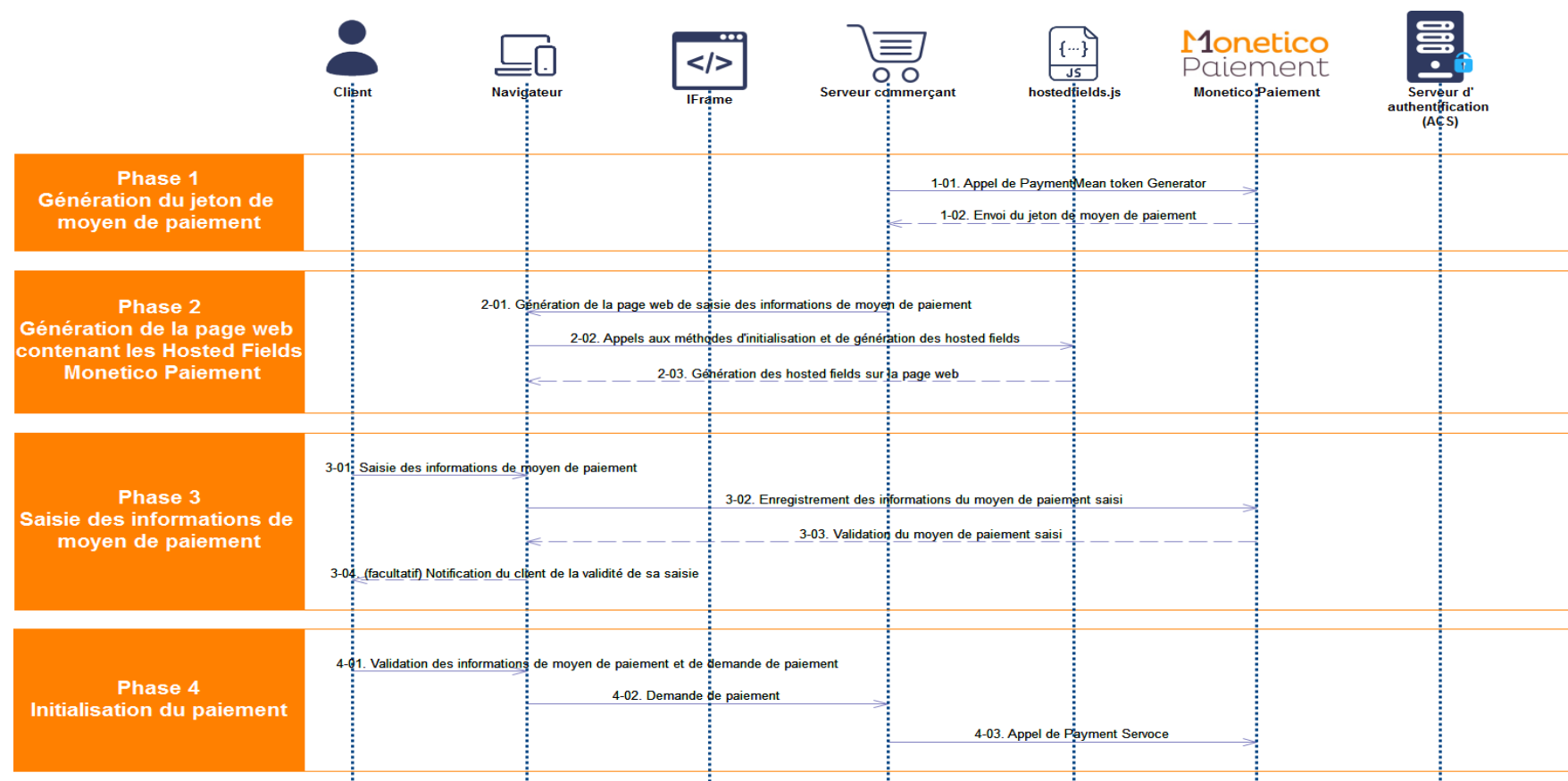
L'implémentation de parcours **d'enregistrement et de réutilisation de carte** nécessite l'activation de l'option « **paiement express** » et doit également faire l'objet d'une demande au support technique.

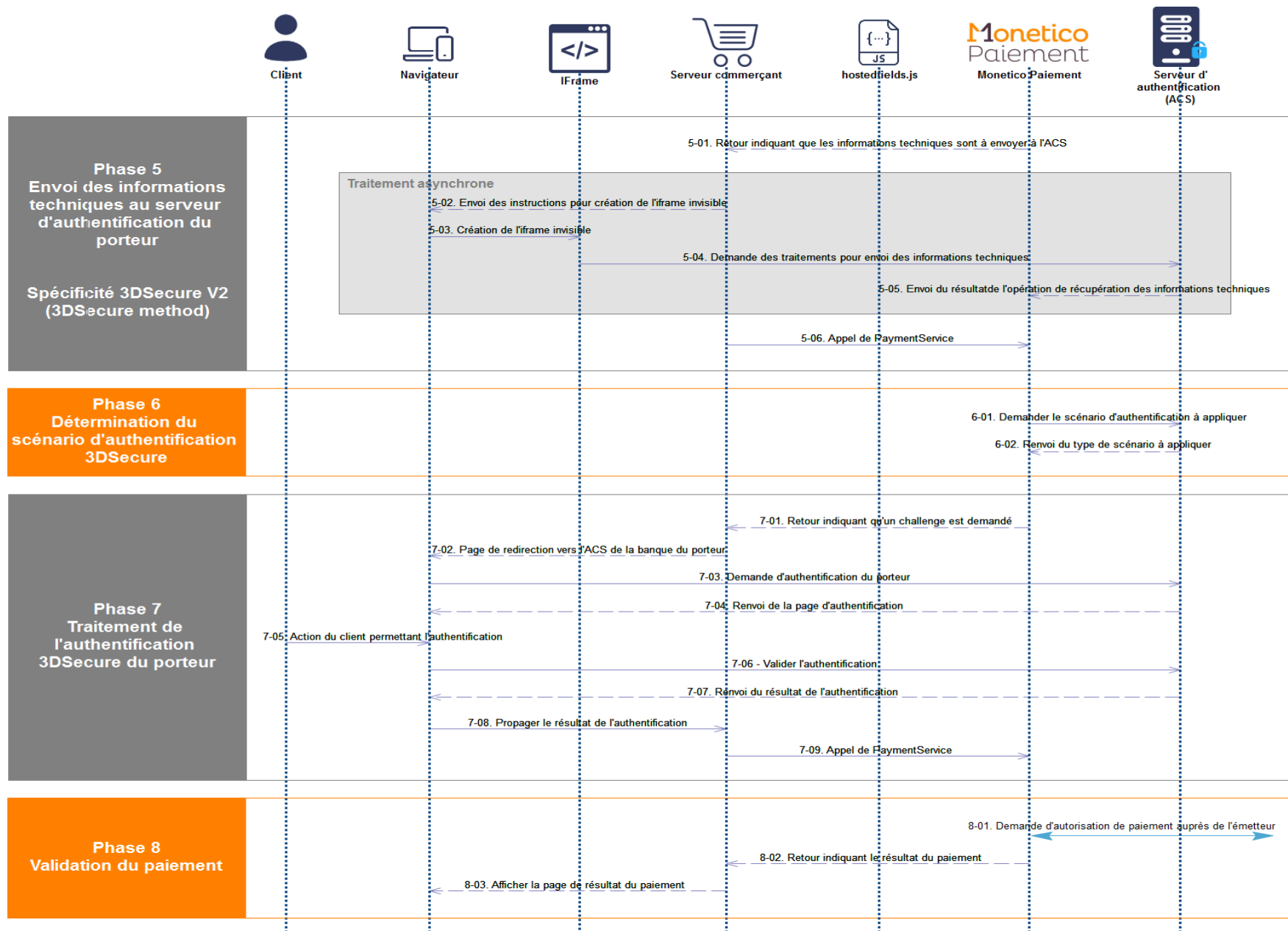
Pour plus de détails sur les différents parcours de paiement ou d'enregistrement de carte, se référer à la [section 5](#).

4 Scenario

Le diagramme ci-dessous décrit le scénario d'une demande de paiement ou d'enregistrement de carte intégrant le système des *Hosted Fields*. En orange, les étapes obligatoires. En gris, les étapes facultatives dépendantes du scénario d'authentification du porteur (3DSecure) devant être appliqué sur l'opération.

A partir de la Phase 5, ce diagramme reprend exactement les éléments décrits dans la section 2 de [\[1\]](#).





5 Cas d'utilisation et parcours

5.1 Demande de paiement et demande de renseignement

Il existe plusieurs types de demande réalisable auprès des API de Monetico Paiement :

- la demande de paiement pour un montant supérieur à 0€ donne lieu à une transaction du montant correspondant
- la demande de renseignement pour un montant égal à 0€ donne lieu à une vérification des informations de la carte de paiement utilisée

Le système de saisie de la carte de paiement via *Hosted Fields* est utilisable pour ces deux types de demande sans distinction.

5.2 Enregistrement et réutilisation de carte

En fonction de la configuration du TPEv, Monetico Paiement peut proposer au porteur de la carte d'enregistrer dans son système d'information les données de celle-ci. Une fois enregistrée, le porteur pourra réutiliser sa carte lors de transactions ultérieures chez le même commerçant sans avoir besoin de ressaisir toutes les informations.

Si la configuration du TPEv le permet, la demande d'enregistrement d'une carte de paiement se fait soit via une demande de paiement, soit via une demande de renseignement, en fournissant à Monetico Paiement un paramètre supplémentaire appelé « *wallet_id* » lors de l'initialisation du parcours.

L'enregistrement de la carte est alors effectué automatiquement par Monetico Paiement en fin de parcours selon les conditions suivantes :

- L'option « paiement express » est active sur le TPEv utilisé pour le parcours
- Un identifiant unique appelé « *wallet_id* » est fourni à Monetico Paiement en début de parcours
- Le porteur a donné son consentement pour l'enregistrement de sa carte au moment de la saisie et le consentement a été recueilli avec succès par Monetico Paiement
- Le porteur est authentifié de manière forte (3DSecure) avec succès au cours de la transaction
- L'opération est autorisée par la banque émettrice de la carte

Une fois l'enregistrement effectué avec succès, la réutilisation de la carte sera possible en fournissant à nouveau l'identifiant « *wallet_id* » à Monetico Paiement en début de parcours.

La mise en place de ce mécanisme est possible via le système *Hosted Fields* de Monetico Paiement. Dans ce cas, l'identifiant unique pour le commerçant doit être défini par celui-ci au début du parcours, et fourni à Monetico Paiement en entrée de l'API « *PaymentMeanTokenGenerator* ».

Attention : il faut bien distinguer l'« identifiant de *wallet* » associé à une carte de paiement du « jeton de saisie temporaire ».

L'« identifiant de *wallet* » est associé à une carte de paiement de manière pérenne et permet de réutiliser celle-ci lors de parcours de paiement ultérieurs.

Le « jeton de saisie temporaire » permet lui de faire le lien entre les informations d'une carte saisie via les différents champs « *hosted fields* » seulement le temps d'un seul parcours de paiement, il n'est plus utilisable une fois le parcours terminé.

Dans le cas d'un parcours impliquant l'enregistrement ou la réutilisation de carte via le système *Hosted Fields*, les deux identifiants devront être utilisés simultanément.

6 Intégration

6.1 Etape 1 : intégration de l'API « PaymentMeanTokenGenerator » (Back-end)

La 1^{ère} étape d'un parcours de paiement intégrant les Hosted Fields Monetico consiste à générer un jeton de saisie de carte temporaire via l'API « PaymentMeanTokenGenerator ». Ce jeton sera valable 1h, temps maximal pour compléter la saisie de la carte et la demande de paiement.

Une fois le jeton généré avec succès, il devra être fourni au SDK « hostedfields.js » pour permettre de générer les différents champs de saisie de la carte (cf. section 6.2 de ce document).

Lors de l'appel à l'API « PaymentMeanTokenGenerator, des informations sur le contexte du parcours de paiement peuvent être fournies afin d'adapter les contrôles qui seront effectués lors de la saisie du moyen de paiement via les champs générés.

La liste des champs de saisie à générer sera différente en fonction du type de parcours de paiement mis en place. L'API « PaymentMeanTokenGenerator » précise donc cette liste en fonction du cas d'utilisation fourni en entrée du service.

Pour plus de détails sur l'intégration du service « PaymentMeanTokenGenerator », se référer au document [\[2\]](#).

6.2 Etape 2 : intégration du SDK « hostedfields.js » (Front-end)

Une fois le jeton de saisie de carte temporaire généré, le SDK « hostedfields.js » peut être initialisé en renseignant la valeur du jeton lors de l'appel à la méthode **initialize()**. Cette action permet à Monetico de faire le lien entre le jeton généré précédemment et les données de carte qui seront saisies par le client à travers les champs Hosted Fields.

Une fois l'initialisation réalisée avec succès, les différents champs Hosted Fields peuvent être configurés et générés via les méthodes **createField()** et **generate()** du SDK.

La méthode **addEventListener()** permet quant à elle d'être informé des actions effectuées par le porteur au moment de la saisie de la carte de paiement, notamment lorsque cette saisie est considérée comme terminée et valide, ou bien lorsque des erreurs de saisie sont détectées.

La liste des champs Hosted Fields à générer sera différente en fonction du type de parcours de paiement mis en place :

Champ	Demande de paiement / renseignement sans enregistrement de carte	Demande de paiement / renseignement avec enregistrement de carte	Demande de paiement avec réutilisation de carte
cardNumber	Obligatoire	Obligatoire	
cardExpDate	Obligatoire	Obligatoire	
cardCvx	Obligatoire	Obligatoire	Conditionnelle
cardHolderName	Obligatoire	Obligatoire	
cardPreview			Optionnelle
cardSwitch			Optionnelle
cardRegistrationConsent		Obligatoire	

Note : pour une demande de paiement avec réutilisation de carte, le champs « **cardCvx** » n'a pas besoin d'être instancié si le contrat possède l'option « oneclick » pour la configuration du contrat précisée lors de la génération du jeton

Note 2 : une carte de paiement ne peut être enregistrée en fin de parcours que si le porteur a donné explicitement son consentement, la carte ne pourra donc pas être enregistrée si le champ « **cardRegistrationConsent** » n'a pas été instancié sur la page

Pour plus de détails sur l'intégration du SDK « hostedfields.js », se référer au document [\[3\]](#).

6.3 Etape 3 : intégration de l'API « PaymentService » (Back-End)

Lorsque la saisie de la carte de paiement est finalisée avec succès, la demande de paiement et/ou d'enregistrement de carte peut être effectuée auprès de l'API « Payment Service ». Le jeton de saisie de carte temporaire doit alors être fourni pour permettre à Monetico de récupérer les données de carte de paiement précédemment saisies via les Hosted Fields.

6.3.1 Appel au service

Le jeton de saisie de carte temporaire doit être fourni en lieu et place des données de carte du porteur à l'appel de « Payment Service ». Le service se charge alors de récupérer les données de carte associées au jeton.

Pour ce faire, l'objet JSON « **payment_mean** » (section 3.5.1.2.3.1 de [\[1\]](#)) est augmenté du champ suivant :

Champ	payment_mean_token
Présence	Obligatoire si les champs « account_number », « expiry_date », « cardholdername », « birth_date », « cvx », « scheme » et « default_scheme » ne sont pas fournis Nécessite la mise en place de l'option « tokenisation » sur le TPE
Description	Jeton temporaire de saisie de carte de paiement Remplace les informations de carte de paiement à fournir à l'initialisation du paiement Doit être généré en amont de la demande de paiement via le service « Payment Mean Token Generator » et complété avec les informations de la carte de paiement par le porteur dans le cadre d'un parcours intégrant le SDK javascript « hostedfields.js »
Format	UUID (RFC 4122)
Valeur(s) possible(s)	
Exemple(s)	eb2294de-a864-4a1f-abda-1b568bf7ca9a

Le jeton n'est à fournir que lors de la **phase d'initialisation de la demande de paiement** (ou de renseignement). Une fois la demande initialisée, il n'est plus nécessaire de renseigner le jeton lors des appels à « Payment Service ».

Note : le jeton remplace l'intégralité des données contenues dans l'objet « **payment_mean** », y compris le paramètre « wallet_id ».

Ci-dessous la description de la structure de l'objet « **payment_mean** » en fonction des cas d'utilisation.

6.3.1.1 Demande de paiement/renseignement avec une carte de paiement

Champ	Parcours standard	Parcours Hosted Fields
wallet_id		

hpan		
name		
account_number	Obligatoire	
expiry_date	Obligatoire	
cardholdername	Obligatoire	
birth_date	Conditionnelle	
cvx	Conditionnelle	
scheme	Obligatoire	
default_scheme	Obligatoire	
payment_mean_token		Obligatoire

6.3.1.2 Demande d'enregistrement d'une carte de paiement

Champ	Parcours standard	Parcours Hosted Fields
wallet_id	Obligatoire	
hpan		
name	Optionnelle	
account_number	Obligatoire	
expiry_date	Obligatoire	
cardholdername	Obligatoire	
birth_date	Conditionnelle	
cvx	Conditionnelle	
scheme	Obligatoire	
default_scheme	Obligatoire	
payment_mean_token		Obligatoire

6.3.1.3 Demande de paiement avec réutilisation d'une carte de paiement

Champ	Parcours standard	Parcours Hosted Fields
wallet_id	Obligatoire	
hpan	Optionnelle	
name	Optionnelle	
account_number		
expiry_date		
cardholdername		
birth_date		
cvx		
scheme	Optionnelle	
default_scheme	Optionnelle	
payment_mean_token		Obligatoire

Pour plus de détails sur les champs décrits ci-dessus, se référer aux sections 3.5.1 et 3.5.1.2.3.1 du document [1].

6.3.2 Retour du service

À la validation du paiement, l'objet JSON « **payment_mean** » retourné présente les informations de cartes au format PCI-DSS. De plus, si un jeton de carte a été utilisé à l'appel du service, le champ « **payment_mean_token** » est également retourné.

Le champ « **return_code** » du JSON de retour du service est également augmenté d'une valeur indiquant une erreur liée au jeton de moyen de paiement.

Valeur	Description	Commentaire
-29	Jeton de moyen de paiement invalide	<p>Le jeton de moyen de paiement fourni via le paramètre « payment_mean_token » est invalide, pour l'une des raisons suivantes :</p> <ul style="list-style-type: none"> le jeton n'existe pas le jeton est expiré le jeton n'est pas associé à une carte de paiement ou les informations de carte saisies sont incomplètes le jeton a déjà été utilisé pour une autre demande de paiement et la demande a été acceptée le jeton a déjà été utilisé pour 5 tentatives de paiement refusées le jeton a été généré pour une autre configuration marchand

Ci-dessous la description de la structure de l'objet « **payment_mean** » en fonction des cas d'utilisation.

6.3.2.1 Demande de paiement/renseignement avec une carte de paiement

Champ	Parcours standard	Parcours Hosted Fields
card_acquired		
wallet_id		
hpan	Obligatoire	Obligatoire
name		
cardholdername	Obligatoire	Obligatoire
birth_date	Conditionnelle	
usage_code	Optionnelle	Optionnelle
account_type	Optionnelle	Optionnelle
origin	Optionnelle	Optionnelle
ecard	Optionnelle	Optionnelle
scheme	Obligatoire	Obligatoire
expiry_date	Obligatoire	Obligatoire
masked_account_number	Obligatoire	Obligatoire
payment_mean_token		Obligatoire

6.3.2.2 Demande d'enregistrement d'une carte de paiement

Champ	Parcours standard	Parcours Hosted Fields
card_acquired	Conditionnelle	Conditionnelle
wallet_id	Obligatoire	Obligatoire
hpan	Obligatoire	Obligatoire
name	Optionnelle	
cardholdername	Obligatoire	Obligatoire
birth_date	Conditionnelle	
usage_code	Optionnelle	Optionnelle
account_type	Optionnelle	Optionnelle
origin	Optionnelle	Optionnelle

ecard	Optionnelle	Optionnelle
scheme	Obligatoire	Obligatoire
expiry_date	Obligatoire	Obligatoire
masked_account_number	Obligatoire	Obligatoire
payment_mean_token		Obligatoire

6.3.2.3 Demande de paiement avec réutilisation d'une carte de paiement

Champ	Parcours standard	Parcours Hosted Fields
card_acquired	Conditionnelle	Conditionnelle
wallet_id	Obligatoire	Obligatoire
hpan	Obligatoire	Obligatoire
name	Optionnelle	
cardholdername	Obligatoire	Obligatoire
birth_date	Optionnelle	
usage_code	Optionnelle	Optionnelle
account_type	Optionnelle	Optionnelle
origin	Optionnelle	Optionnelle
ecard	Optionnelle	Optionnelle
scheme	Obligatoire	Obligatoire
expiry_date	Obligatoire	Obligatoire
masked_account_number	Obligatoire	Obligatoire
payment_mean_token		Obligatoire

Note : le champ « **card_acquired** » ne sera valorisé que si la carte est enregistrée avec succès dans le SI de Monetico Paiement, il ne sera donc présent que dans le retour du dernier appel à l'API « Payment Service » et vaudra alors « **true** ».

Pour plus de détails sur les champs décrits ci-dessus, se référer aux sections 3.6.1 et 3.6.1.3.1 du document [\[1\]](#).

6.3.3 Exemples d'appel et de retour du service

6.3.3.1 Demande de paiement/renseignement avec une carte de paiement

Corps d'une requête d'initialisation de paiement (phase 1)
<pre>{ "merchant_configuration":{ "point_of_sale":"9000001", "version":"3.0", "language":"FR", "configuration":"emulation3d" }, "order":{ "date":"2022-03-06T17:23:44", "customer":{ "mail":"customer@mail.com" }, "context":{ ... } }, }</pre>

```

"payment":{
  "transaction_initiator":"cardholder",
  "reference":"89e8b64c-513f-4d9d-8ff9-4520d10ef780",
  "payment_mean":{
    "payment_mean_token":"eb2294de-a864-4a1f-abda-1b568bf7ca9a"
  },
  "amount":{
    "value":10001,
    "currency":"EUR",
    "exponent":2
  },
  "payment_options":{
    "merchant_identification":{
      "name":"MyWebSite"
    }
  }
},
"authentication":{
  "merchant_preference":"challenge_requested",
  "merchant_redirection_url":"https://www.mywebsite.com/authentication\_result.cgi",
  "challenge_window_size":"500x600"
}
}

```

Corps d'une réponse d'initialisation de paiement (phase 1)

```

{
  "authentication":{
    "details":{
      "transactionID":"9d7a8a34-4d15-4e10-810f-bc91694991b3"
    },
    "protocol":"3DSecure",
    "status":"threeDSmethod_requested",
    "version":"2.1.0"
  },
  "merchant_configuration":{
    "configuration":"emulation3d",
    "language":"FR",
    "point_of_sale":"9000001",
    "version":"3.0"
  },
  "next_step": {
    "data": {
      "threeDSMethodData": "eyJ0aHJlZURTTWV0aG9kTm90aWZpY2F0aW9uVVJMIjoiaHR0cHM6Ly9ycXQtcGF5bWVudC1hcGkuZS1pLmNvbS90ZXN0L3Rkc21ldGhvZGNhbGxiYWNRlmNnaSIsInRo cmVlRFNTZXJ2ZXJUcmFuc01EIjoiaWQ3YTlhMzQtNGQxNS00ZTEwLTgxMGYtYmM5MTY5NDk5MWIzIn0K"
    },
    "recommended_implementation": [
      "invisible_iframe"
    ],
    "step": "technical_information_collecting",
  }
}

```



```

        "CRes": "Y",
        "merchantPreference": "no_preference",
        "transactionID": "9d7a8a34-4d15-4e10-810f-bc91694991b3"
    },
    "protocol": "3DSecure",
    "status": "authenticated",
    "version": "2.1.0"
},
"merchant_configuration": {
    "configuration": "emulation3d",
    "language": "FR",
    "point_of_sale": "9000001",
    "version": "3.0"
},
"payment": {
    "amount": {
        "currency": "EUR",
        "exponent": 2,
        "value": 10001
    },
    "authorisation": {
        "date": "2019-09-06",
        "number": "000000"
    },
    "payment_mean": {
        "ecard": false,
        "expiry_date": "2035-12",
        "hpan": "C5F5A5ACAACB96129882D1A9DB9E1FBBED4FBE93",
        "masked_account_number": "000003*****0007",
        "payment_mean_token": "eb2294de-a864-4a1f-abda-1b568bf7ca9a",
        "origin": "FR",
        "scheme": "MASTERCARD"
    },
    "payment_options": {
        "merchant_identification": {
            "name": "MyWebSite"
        }
    },
    "reference": "89e8b64c-513f-4d9d-8ff9-4520d10ef780",
    "status": "captured"
},
"payment_token": "cc31528d-a881-4e29-bbba-f016fe2df1dc",
"request_token": "768ada05-3ca7-44c3-a53d-f110e1a3e4b7",
"return_code": 1
}

```

6.3.3.2 Demande d'enregistrement d'une carte de paiement

Corps d'une requête d'initialisation de paiement (phase 1)

```

{
    "merchant_configuration": {

```

```

    "point_of_sale": "9000001",
    "version": "3.0",
    "language": "FR",
    "configuration": "emulation3d"
  },
  "order": {
    "date": "2022-03-06T17:23:44",
    "customer": {
      "mail": "customer@mail.com"
    },
    "context": {
      ...
    }
  },
  "payment": {
    "transaction_initiator": "cardholder",
    "reference": "89e8b64c-513f-4d9d-8ff9-4520d10ef780",
    "payment_mean": {
      "payment_mean_token": "eb2294de-a864-4a1f-abda-1b568bf7ca9a"
    },
    "amount": {
      "value": 0,
      "currency": "EUR",
      "exponent": 2
    },
    "payment_options": {
      "merchant_identification": {
        "name": "MyWebSite"
      }
    }
  },
  "authentication": {
    "merchant_preference": "challenge_requested",
    "merchant_redirection_url": "https://www.mywebsite.com/authentication_result.cgi",
    "challenge_window_size": "500x600"
  }
}

```

Corps d'une réponse d'initialisation de paiement (phase 1)

```

{
  "authentication": {
    "details": {
      "transactionID": "9d7a8a34-4d15-4e10-810f-bc91694991b3"
    },
    "protocol": "3DSecure",
    "status": "threeDSmethod_requested",
    "version": "2.1.0"
  },
  "merchant_configuration": {
    "configuration": "emulation3d",
    "language": "FR",
    "point_of_sale": "9000001",
    "version": "3.0"
  }
}

```

```

    },
    "next_step": {
      "data": {
        "threeDSMethodData": "eyJ0aHJlZURTTWV0aG9kTm90aWZpY2F0aW9uVVJMIjoiHR0cHM6Ly9ycXQtcGF5bWVudC1hcGkuZS1pLmNvbS90ZXN0L3Rkc21ldGhvZGNhbGxiYWNRmNnaSIsInRo cmVlRFNTZXJ2ZXJUcmFuc0lEIjoiOWQ3YThhMzQtNGQxNS00ZTEwLTgxMGYtYmM5MTY5NDk5MWIzIn0K
"
      },
      "recommended_implementation": [
        "invisible_iframe"
      ],
      "step": "technical_information_collecting",
      "url": "https://url-acsc-client/acs.cgi"
    },
    "payment": {
      "amount": {
        "currency": "EUR",
        "exponent": 2,
        "value": 0
      },
      "payment_mean": {
        "ecard": false,
        "expiry_date": "2035-12",
        "hpan": "C5F5A5ACAACB96129882D1A9DB9E1FBBED4FBE93",
        "masked_account_number": "000003*****0007",
        "payment_mean_token": "eb2294de-a864-4a1f-abda-1b568bf7ca9a",
        "origin": "FR",
        "scheme": "MASTERCARD",
        "wallet_id": "monClientRef001"
      },
      "payment_options": {
        "merchant_identification": {
          "name": "MyWebSite"
        }
      },
      "reference": "89e8b64c-513f-4d9d-8ff9-4520d10ef780",
      "status": "cardholder_authentication_pending"
    },
    "payment_token": "cc31528d-a881-4e29-bbba-f016fe2df1dc",
    "request_token": "083bcfe3-62a3-4400-a0b0-1e1a8ac95505",
    "return_code": 2
  }
}

```

Corps d'une requête de finalisation de paiement (phase 4)

```

{
  "payment_token": "cc31528d-a881-4e29-bbba-f016fe2df1dc",
  "authentication": {
    "details": {
      "cres": "ewogICAiYWZzVHJhbnNJRCIgOiAiOGY4N2I3ZjQtZTUwMC00MDkwLThkZTIitZmNlMjIwNDQ0MGNIiIiwKICAgIm1lc3NhZ2VUeXB1IiA6ICJDUmVzIiwKICAgIm1lc3NhZ2VWZXJzaW9uIiA

```

```

6ICIyLjEuMCIsCiAgICJ0aHJlZURTU2VydmVyVHJhbnNJRClgOiAiODI1ZTlmZmQtYjA0NS00YTgxLWI
zYzItZDBmMzQ0MWMYNGZhIiwKICAgInRyYW5zU3RhZHVzIiA6ICJZIgp9Cg==",
    "threeDSSessionData": "982e7b2c-941e-454e-a6ad-400bf503d32b"
  }
}
}

```

Corps d'une réponse d'initialisation de paiement (phase 4)

```

{
  "authentication": {
    "details": {
      "ARes": "C",
      "CRes": "Y",
      "merchantPreference": "no_preference",
      "transactionID": "9d7a8a34-4d15-4e10-810f-bc91694991b3"
    },
    "protocol": "3DSecure",
    "status": "authenticated",
    "version": "2.1.0"
  },
  "merchant_configuration": {
    "configuration": "emulation3d",
    "language": "FR",
    "point_of_sale": "9000001",
    "version": "3.0"
  },
  "payment": {
    "amount": {
      "currency": "EUR",
      "exponent": 2,
      "value": 0
    },
    "authorisation": {
      "date": "2019-09-06",
      "number": "000000"
    },
    "payment_mean": {
      "card_acquired": true,
      "ecard": false,
      "expiry_date": "2035-12",
      "hpan": "C5F5A5ACAACB96129882D1A9DB9E1FBBED4FBE93",
      "masked_account_number": "000003*****0007",
      "payment_mean_token": "eb2294de-a864-4a1f-abda-1b568bf7ca9a",
      "origin": "FR",
      "scheme": "MASTERCARD",
      "wallet_id": "monClientRef001"
    },
    "payment_options": {
      "merchant_identification": {
        "name": "MyWebSite"
      }
    }
  }
}

```

```

    },
    "reference": "89e8b64c-513f-4d9d-8ff9-4520d10ef780",
    "status": "authorised"
  },
  "payment_token": "cc31528d-a881-4e29-bbba-f016fe2df1dc",
  "request_token": "768ada05-3ca7-44c3-a53d-f110e1a3e4b7",
  "return_code": 1
}

```

6.3.3.3 Demande de paiement avec réutilisation d'une carte de paiement

Corps d'une requête d'initialisation de paiement (phase 1)

```

{
  "merchant_configuration": {
    "point_of_sale": "9000001",
    "version": "3.0",
    "language": "FR",
    "configuration": "emulation3d"
  },
  "order": {
    "date": "2022-03-06T17:23:44",
    "customer": {
      "mail": "customer@mail.com"
    },
    "context": {
      ...
    }
  },
  "payment": {
    "transaction_initiator": "cardholder",
    "reference": "89e8b64c-513f-4d9d-8ff9-4520d10ef780",
    "payment_mean": {
      "payment_mean_token": "eb2294de-a864-4a1f-abda-1b568bf7ca9a"
    },
    "amount": {
      "value": 10001,
      "currency": "EUR",
      "exponent": 2
    },
    "payment_options": {
      "merchant_identification": {
        "name": "MyWebSite"
      }
    }
  },
  "authentication": {
    "merchant_preference": "challenge_requested",
    "merchant_redirection_url": "https://www.mywebsite.com/authentication\_result.cgi",
    "challenge_window_size": "500x600"
  }
}

```

Corps d'une réponse d'initialisation de paiement (phase 1)

```

{
  "authentication":{
    "details":{
      "transactionID":"9d7a8a34-4d15-4e10-810f-bc91694991b3
    },
    "protocol":"3DSecure",
    "status":"threedsmethod_requested",
    "version":"2.1.0"
  },
  "merchant_configuration":{
    "configuration":"emulation3d",
    "language":"FR",
    "point_of_sale":"9000001",
    "version":"3.0"
  },
  "next_step": {
    "data": {
      "threeDSMethodData": "eyJ0aHJlZURTTWV0aG9kTm90aWZpY2F0aW9uVVJMIjoiaHR0
cHM6Ly9ycXQtGF5bWVudC1hcGkuZS1pLmNvbS90ZXN0L3Rkc21ldGhvZGNhbGxiYWNRmNnaSIzInRo
cmVlRFNTZXJ2ZXJUcmFuc0lEIjoioWQ3YTlhMzQtNGQxNS00ZTEwLTgxMGYtYmM5MTY5NDk5MWIzIn0K
"
    },
    "recommended_implementation": [
      "invisible_iframe"
    ],
    "step": "technical_information_collecting",
    "url": "https://url-acsc-client/acs.cgi"
  },
  "payment":{
    "amount":{
      "currency":"EUR",
      "exponent":2,
      "value":10001
    },
    "payment_mean":{
      "ecard":false,
      "expiry_date":"2035-12",
      "hpan":"C5F5A5ACAACB96129882D1A9DB9E1FBBED4FBE93",
      "masked_account_number":"000003*****0007",
      "payment_mean_token":"eb2294de-a864-4a1f-abda-1b568bf7ca9a",
      "origin":"FR",
      "scheme":"MASTERCARD",
      "wallet_id":"monClientRef001"
    },
    "payment_options":{
      "merchant_identification":{
        "name":"MyWebSite"
      }
    },
    "reference":"89e8b64c-513f-4d9d-8ff9-4520d10ef780",
    "status":"cardholder_authentication_pending"
  }
}

```

```

    },
    "payment_token": "cc31528d-a881-4e29-bbba-f016fe2df1dc",
    "request_token": "083bcfe3-62a3-4400-a0b0-1e1a8ac95505",
    "return_code": 2
}

```

Corps d'une requête de finalisation de paiement (phase 4)

```

{
  "payment_token": "cc31528d-a881-4e29-bbba-f016fe2df1dc",
  "authentication": {
    "details": {
      "cres": "ewogICAiYWNzVHJhbnNJRCIgOiAiODI1ZTlmZmQtYjA0NS00YTgxLWIzYzItZDBmMzQ0MWMYNGZhIiwKICAgInRyYW5zU3RhZHVzIiA6ICJZIGp9Cg==",
      "threeDSSessionData": "982e7b2c-941e-454e-a6ad-400bf503d32b"
    }
  }
}

```

Corps d'une réponse d'initialisation de paiement (phase 4)

```

{
  "authentication": {
    "details": {
      "ARes": "C",
      "CRes": "Y",
      "merchantPreference": "no_preference",
      "transactionID": "9d7a8a34-4d15-4e10-810f-bc91694991b3"
    }
  },
  "protocol": "3DSecure",
  "status": "authenticated",
  "version": "2.1.0"
},
  "merchant_configuration": {
    "configuration": "emulation3d",
    "language": "FR",
    "point_of_sale": "9000001",
    "version": "3.0"
  },
  "payment": {
    "amount": {
      "currency": "EUR",
      "exponent": 2,
      "value": 10001
    },
    "authorisation": {
      "date": "2019-09-06",
      "number": "000000"
    }
  },
  "payment_mean": {

```

```

    "ecard":false,
    "expiry_date":"2035-12",
    "hpan":"C5F5A5ACAACB96129882D1A9DB9E1FBBED4FBE93",
    "masked_account_number":"000003*****0007",
    "payment_mean_token":"eb2294de-a864-4a1f-abda-1b568bf7ca9a",
    "origin":"FR",
    "scheme":"MASTERCARD",
    "wallet_id":"monClientRef001"
  },
  "payment_options":{
    "merchant_identification":{
      "name":"MyWebSite"
    }
  },
  "reference":"89e8b64c-513f-4d9d-8ff9-4520d10ef780",
  "status":"captured"
},
"payment_token":"cc31528d-a881-4e29-bbba-f016fe2df1dc",
"request_token":"768ada05-3ca7-44c3-a53d-f110e1a3e4b7",
"return_code":1
}

```

7 Environnements

Les services de Monetico Paiement sont disponibles sur différents environnements :

- un environnement de test dit « sandbox »
- un environnement d'exploitation dit « live »

L'environnement devra rester le même tout au long du parcours de paiement. Par exemple, un jeton de carte généré sur l'environnement live ne pourra pas être utilisé avec le SDK disponible sur l'environnement de sandbox (et vice-versa). De même, un jeton correspondant à une carte saisie sur l'environnement de sandbox ne pourra pas être utilisé pour effectuer une opération de paiement sur l'environnement live.

7.1 Environnement de test dit « sandbox »

Le rôle de notre serveur de test est de vous permettre de valider vos développements en vérifiant votre intégration à notre solution.

Aucun paiement accepté par notre serveur de paiement de test ne donne lieu à une mise en recouvrement.

Les différents services Monetico sont disponibles sur cet environnement aux URL suivantes :

API « PaymentMeanTokenGenerator »	https://payment-api.e-i.com/test/paymentmeantokengenerator.cgi
API « PaymentService »	https://payment-api.e-i.com/test/paymentservice.cgi
SDK « hostedfields.js »	https://p.monetico-services.com/test/hostedfields/hostedfields.js

Pour tester l'intégration des *hosted fields* dans cet environnement, nous mettons à votre disposition les cartes de paiement suivantes :

Scénario	Réseau carte	Paiement accepté	Paiement refusé
La carte n'est pas enrôlée au 3DSecure	Visa	0000010000000021	0000010000000022
	Mastercard	0000030000000021	0000030000000022
L'authentification est effectuée avec succès sans challenge	Visa	0000010000000023	0000010000000024
	Mastercard	0000030000000023	0000030000000024
L'authentification est effectuée avec succès avec challenge	Visa	0000010000000025	0000010000000026
	Mastercard	0000030000000025	0000030000000026
L'authentification n'a pas pu être complétée (problème technique ou autre)	Visa		0000010000000027
	Mastercard		0000030000000027
Une tentative d'authentification a bien été effectuée. L'authentification n'a pas pu se faire mais une preuve a été générée (CAVV)	Visa	0000010000000028	
	Mastercard	0000030000000028	
L'authentification a échoué sans challenge	Visa		0000010000000029
	Mastercard		0000030000000029
L'authentification a échoué avec challenge	Visa		0000010000000030
	Mastercard		0000030000000030
L'authentification a été refusée par l'émetteur	Visa		0000010000000031
	Mastercard		0000030000000031

7.2 Environnement d'exploitation dit « live »

Après avoir validé vos développements, vous pourrez utiliser les différents services Monetico en environnement live, disponibles aux URL suivantes :

API « PaymentMeanTokenGenerator »	https://payment-api.e-i.com/paymentmeantokengenerator.cgi
API « PaymentService »	https://payment-api.e-i.com/paymentservice.cgi
SDK « hostedfields.js »	https://p.monetico-services.com/hostedfields/hostedfields.js

Attention : les cartes utilisées pour la saisie devront respecter l'algorithme de Luhn et les opérations de paiement effectuées donneront lieu à de réels mouvements bancaires.