

Monetico Online

Païement sécurisé sur Internet

**Demande de jeton de saisie
temporaire de moyen de paiement**



SOMMAIRE

1.	<i>Introduction</i>	3
2.	<i>Prérequis</i>	4
3.	<i>Spécifications des messages vers la plateforme Monetico Online</i>	5
3.1.	Sécurité des échanges	5
3.2.	Format des échanges	5
3.3.	Calcul du sceau HMAC	5
1.1.1.	Clé de sécurité commerçant	5
1.1.2.	Principe du calcul du sceau	5
3.4.	Environnements mis à disposition	7
1.1.3.	Environnement de test dit « sandbox »	7
1.1.4.	Environnement d'exploitation dit « live »	7
3.5.	Appel au service « PaymentMeanTokenGenerator »	8
1.1.5.	Les données d'en tête	8
1.1.6.	Les données du corps.	8
1.1.7.	Exemples d'appel au service	12
3.6.	Retour du service « PaymentMeanTokenGenerator »	15
1.1.8.	Structure du retour du service	15
1.1.9.	Exemples de retour du service	19

1. Introduction

Le rôle du service « `PaymentMeanTokenGenerator` » de Monetico Online est de générer un identifiant, dit « jeton » ou « *token* », permettant d'effectuer la saisie temporaire d'un moyen de paiement dans le cadre d'un parcours de paiement ou d'enregistrement de carte de paiement sur un site e-commerce.

Un jeton de saisie temporaire de moyen de paiement est généré pour un numéro de contrat, une configuration de site et une langue données. Il est valable pour une durée d'une heure, temps accordé au porteur pour effectuer la saisie de son moyen de paiement.

Les éléments d'interface pour la saisie du moyen de paiement associé au jeton sont fournis par le SDK en JavaScript des *Hosted Fields* de Monetico Online. L'utilisation de cette bibliothèque n'est permise qu'en fournissant un jeton en cours de validité, ainsi qu'en précisant le numéro de contrat associé à celui-ci.

Une fois la saisie des informations de moyen de paiement complétée par le porteur, le jeton sera utilisable pour effectuer une opération de paiement ainsi que des opérations de demande de renseignement sur le moyen de paiement via l'API « `PaymentService` » de Monetico Online. Ces opérations devront être effectuées avec la même configuration de contrat utilisée lors de la génération du jeton, et sur le même environnement de paiement, à savoir « *sandbox* » ou « *live* ».

2. Prérequis

L'intégration du système *Hosted Fields* de Monetico Online fait partie des offres « Starter » et « Premium » de Monetico Online.

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 prévue pour les appels aux API de Monetico Online
- le « code langue » associé au « code site » décrit ci-dessus

3. Spécifications des messages vers la plateforme Monetico Online

3.1. Sécurité des échanges

Les informations de la demande de paiement sont envoyées au serveur Monetico Online par un message HTTPS, en utilisant le protocole de sécurisation des échanges **TLS V1.2** uniquement.

3.2. Format des échanges

L'application du commerçant doit émettre une requête HTTP de type **POST** à destination du service de demande de paiement par API sur les serveurs de la banque.

Cette requête doit contenir certaines informations en en-tête, et son corps consiste en un document au format **JSON/UTF-8**.

3.3. Calcul du sceau HMAC

1.1.1. Clé de sécurité commerçant

Une clé de sécurité, propre à chaque TPE, destinée à certifier les données échangées entre le serveur du commerçant et le serveur sécurisé Monetico Online, est indispensable pour utiliser le service de génération de jeton de moyen de paiement. Un lien permettant de télécharger cette clé de sécurité depuis le tableau de bord est envoyée par notre centre de support au commerçant.

Le commerçant peut demander la génération d'une nouvelle clé, périodiquement ou à l'occasion d'événements tels qu'une mise en production, un changement d'hébergeur, un changement de prestataire, etc.

Il est de la responsabilité du commerçant de conserver cette clé de façon sûre et confidentielle en exploitant les meilleurs outils disponibles dans son environnement.

La clé de sécurité est représentée de façon externe par 40 caractères hexadécimaux (par exemple : 0123456789ABCDEF0123456789ABCDEF01234567). Cette représentation externe doit être convertie en une chaîne de 20 octets (représentation opérationnelle) avant utilisation.

1.1.2. Principe du calcul du sceau

Le sceau, à mettre dans le header MAC, est calculé à l'aide d'une fonction de hachage cryptographique en combinaison avec une clé secrète respectant les spécifications de la RFC 2104.

Cette fonction générera le sceau à partir de données à certifier et de la clé de sécurité commerçant sous sa forme opérationnelle.

Les données à certifier sont l'intégralité du corps de la requête HTTP.

Remarque : Les espaces, tabulations, ou retours à la ligne figurant dans le corps de la requête doivent être pris en compte pour le calcul du sceau MAC.

Un exemple de calcul de sceau MAC :

```
POST /test/paymentservice.cgi HTTP/1.1
Content-Type:application/json
MAC:418F0B411E6E57105AA6FF592B8EE6DB52351E19
Host:payment-api.e-i.com
Accept-Encoding:gzip,
deflate
Connection:keep-alive
{
  "action" : "InitializePaymentMean",
  "merchant_configuration" : {
    "point_of_sale" : "9000001",
    "language" : "FR",
    "configuration" : "emulation3d"
  },
  "datetime": " 2020-09-24T15:50:54"
}
```

En **rouge**, le MAC calculé en utilisant la clé décrite en section 3.3.1

En **vert**, un exemple des données à utiliser pour le calcul du sceau mac.

3.4. Environnements mis à disposition

1.1.3. Environnement de test dit « sandbox »

Le rôle de notre serveur de test « sandbox » est de vous permettre de tester et de valider vos développements.

Sur ce serveur, le seul contrôle effectué sur les données de moyen de paiement est un contrôle de structure. Il n'y a pas d'autres contrôles effectués : date d'expiration, contrôle du fichier des cartes en opposition, etc., comme cela existe sur notre serveur de production.

Bien sûr, aucun paiement accepté par notre serveur de paiement de test ne donne lieu à une mise en recouvrement.

Le service de génération d'un jeton de moyen de paiement sur l'environnement de test est disponible à l'adresse suivante :

<https://payment-api.e-i.com/test/paymentmeantokengenerator.cgi>

1.1.4. Environnement d'exploitation dit « live »

Vous devez faire une demande auprès de l'assistance technique pour faire passer votre TPE en production.

Une fois le passage en production effectué, vous pourrez vous adresser au service de génération d'un jeton sur le serveur de production, disponible à l'adresse suivante :

<https://payment-api.e-i.com/paymentmeantokengenerator.cgi>

Nous attirons votre attention sur le fait que les requêtes de paiement adressées au serveur de production seront des paiements réels.

3.5. Appel au service « PaymentMeanTokenGenerator »

1.1.5. Les données d'en tête

Les données d'en-tête attendues sont les suivantes :

Champ	Content-Type
Présence	Obligatoire
Description	Précise le format des échanges.
Format	application/json; charset=utf-8
Valeur(s) possible(s)	

Champ	MAC
Présence	Obligatoire
Description	Sceau issu de la certification de données envoyées au service de génération de jeton.
Format	40 caractères hexadécimaux
Valeur(s) possible(s)	[0-9A-Fa-f]{40}
Exemple(s)	3BB2E1BFE87A72380B2455C2F1C80D784DC02346

1.1.6. Les données du corps.

Le corps de la requête est un document au format JSON/UTF-8. Il contient les informations suivantes :

Champ JSON	Type JSON	Présence	Détails
action	Chaîne	Obligatoire	lien
context	Objet	Optionnelle	lien
datetime	Chaîne	Obligatoire	lien
merchant_configuration	Objet	Obligatoire	lien
restrictions	Objet	Facultatif	lien

Champ	action
Présence	Obligatoire
Description	Permet de savoir quelle action concernant le jeton est à réaliser
Format	InitializePaymentMean
Valeur(s) possible(s)	
Exemple(s)	InitializePaymentMean

Attribut	datetime
Description	Date et heure de l'appel
Format	Chaîne
Restrictions	Format ISO 8601 Du type AAAA-MM-JJTHH:MM:SS Où AAAA = année sur 4 chiffres, MM = mois sur 2 chiffres, JJ = jour sur deux chiffres, HH = heure, MM = minutes, SS = secondes.
Exemple(s)	2020-09-22T17:15:06

1.1.6.1. Objet « context »

L'objet « context » contient des informations sur le parcours dans le cadre duquel le moyen de paiement associé au jeton généré sera saisi par le consommateur.

L'objet « context » doit obligatoirement être renseigné pour les deux types de parcours suivants :

- Parcours d'enregistrement de carte dans le système d'information de Monetico Online
- Parcours de paiement avec une carte enregistrée au préalable dans le système d'information de Monetico Online

Si l'objet « context » n'est pas renseigné, Monetico Online considère que la saisie du moyen de paiement s'effectue dans le cadre d'un parcours de paiement sans possibilité de réutilisation d'une carte déjà enregistrée et sans possibilité d'enregistrer la carte en fin de parcours.

Champ JSON	Type JSON	Présence	Détails
use_case	Chaîne	Obligatoire	lien
wallet_id	Chaîne	Optionnelle	lien

Champ	use_case
Présence	Obligatoire
Description	Cas d'utilisation de la saisie du moyen de paiement associé au jeton Correspond au type de parcours dans le cadre duquel la saisie sera effectuée par le porteur CardRegistration : parcours d'enregistrement de carte via demande de renseignement à 0 EUR Payment : parcours de paiement avec ou sans enregistrement de carte en fin de parcours, et avec ou sans réutilisation de carte déjà enregistrée
Format Valeur(s) possible(s)	Choix parmi : CardRegistration, Payment
Exemple(s)	CardRegistration

Champ	wallet_id
Présence	Optionnelle Doit être renseigné dans le cadre d'un parcours d'enregistrement de carte ou d'un parcours de paiement avec réutilisation d'une carte enregistrée au préalable Nécessite la mise en place de l'option « paiement express » ou « one-click » sur le contrat Monetico Online. Se rapprocher du support pour plus d'informations.
Description	Identifiant du wallet client.
Format Valeur(s) possible(s)	De 1 à 64 caractères alphanumériques [a-zA-Z0-9]{1,64}
Exemple(s)	monClientRef001

1.1.6.2. Objet « merchant_configuration »

L'objet « merchant_configuration » contient la configuration commerçant liée au contrat. Ces informations seront liées au jeton carte retourné.

Champ JSON	Type JSON	Présence	Détails
configuration	Chaîne	Obligatoire	lien
language	Chaîne	Obligatoire	lien
point_of_sale	Chaîne	Obligatoire	lien

Champ	configuration
Présence	Obligatoire
Description	Code alphanumérique permettant au commerçant d'utiliser le même TPE Virtuel pour des sites différents (paramétrages distincts) se rapportant à la même activité. Il s'agit de votre code société. Doit être identique à celui utilisé ultérieurement pour la demande de paiement ou de renseignement via l'API « PaymentService ».
Format	Chaîne de caractères générée à la création de votre contrat
Valeur(s) possible(s)	
Exemple(s)	maSociete

Champ	language
Présence	Obligatoire
Description	Code langue associé à la configuration du contrat Doit être identique à celui utilisé ultérieurement pour la demande de paiement ou de renseignement via l'API « PaymentService ».
Format	Choix parmi : DE EN ES FR IT JA NL PT SV
Valeur(s) possible(s)	
Exemple(s)	FR

Champ	point_of_sale
Présence	Obligatoire
Description	Numéro de votre TPE virtuel Doit être identique à celui utilisé ultérieurement pour la demande de paiement ou de renseignement via l'API « PaymentService ».
Format	7 caractères alphanumériques
Valeur(s) possible(s)	[A-Za-z0-9]{7}
Exemple(s)	1234567

1.1.6.3. Objet « restrictions »

L'objet « restrictions » permet de définir des critères conditionnant l'acceptation des cartes de paiement qui seront saisies dans les champs Hosted Fields une fois le jeton de moyen de paiement généré.

Attention : il est possible que des critères par défaut d'acceptation des cartes de paiement aient été définis lors de la création du TPE (si une demande spécifique dans ce sens a été effectuée auprès du

support technique). Dans ce cas, l'ensemble des critères par défaut du TPE seront surchargés par le contenu de l'objet « restrictions » fourni au service « PaymentMeanTokenGenerator ».

Champ JSON	Type JSON	Présence	Détails
expiration_delay	Nombre	Facultatif	lien
virtual_card	Booléen	Facultatif	lien
subscheme	Liste de chaînes	Facultatif	lien
account_type	Chaîne	Facultatif	lien
prepaid_card	Booléen	Facultatif	lien

Champ	expiration_delay
Présence	Facultatif
Description	Restriction sur la date d'expiration de carte (en mois). Lorsque ce critère est défini, le délais d'expiration des cartes doit être supérieur à la valeur fournie
Format	Nombre supérieur à 0
Valeur(s) possible(s)	
Exemple(s)	4 (la carte doit expirer dans plus de 4 mois à compter de la date du jour)

Champ	virtual_card
Présence	Facultatif
Description	Restriction sur le type de carte. Lorsque ce critère est défini, les cartes virtuelles (e-card) seront acceptées ou non.
Format	true, false
Valeur(s) possible(s)	

Champ	subscheme
Présence	Facultatif
Description	Restriction sur la marque de la carte. Lorsque ce critère est défini, les cartes dont la marque appartient à la liste ne seront plus acceptées.
Format	"MAESTRO", "ELECTRON", "CIRRUS"
Valeur(s) possible(s)	
Exemple(s)	["MAESTRO", "ELECTRON"]

Champ	account_type
Présence	Facultatif
Description	Restriction sur le type de compte de la carte. Lorsque cette restriction est définie, les cartes dont le type du compte bancaire associé correspond à la valeur définie ne seront pas acceptées.
Format	"individual", "business", "both", "other_type", "no_account_type"
Valeur(s) possible(s)	

Champ	prepaid_card
Présence	Facultatif
Description	Restriction sur les carte prépayées. Lorsque cette restrictions est définie, les cartes prépayée seront acceptées ou non.
Format Valeur(s) possible(s)	true, false

1.1.7. Exemples d'appel au service

1.1.7.1. Appel sans contexte de parcours précisé

```
1. {
2.   "action" : "InitializePaymentMean",
3.   "datetime": "2020-09-22T17:15:06",
4.   "merchant_configuration": {
5.     "configuration": "emulation3d",
6.     "language": "FR",
7.     "point_of_sale": "9000001"
8.   }
9. }
```

1.1.7.2. Appel dans un contexte d'enregistrement de carte via demande de renseignement

```
1. {
2.   "action" : "InitializePaymentMean",
3.   "context": {
4.     "use_case": "CardRegistration",
5.     "wallet_id": "monClientRef001"
6.   },
7.   "datetime": "2020-09-22T17:15:06",
8.   "merchant_configuration": {
9.     "configuration": "emulation3d",
10.    "language": "FR",
11.    "point_of_sale": "9000001"
12.  }
13. }
```

1.1.7.3. Appel dans un contexte de paiement sans enregistrement de carte et sans réutilisation d'une carte enregistrée

```
14. {
15.   "action" : "InitializePaymentMean",
16.   "context": {
17.     "use_case": "Payment"
18.   },
19.   "datetime": "2020-09-22T17:15:06",
20.   "merchant_configuration": {
21.     "configuration": "emulation3d",
22.     "language": "FR",
23.     "point_of_sale": "9000001"
24.   }
25. }
```

1.1.7.4. Appel dans un contexte de paiement avec enregistrement de carte ou avec réutilisation d'une carte enregistrée

```
26. {
27.   "action" : "InitializePaymentMean",
28.   "context": {
29.     "use_case": "Payment",
30.     "wallet_id": " monClientRef001"
31.   },
32.   "datetime": "2020-09-22T17:15:06",
33.   "merchant_configuration": {
34.     "configuration": "emulation3d",
35.     "language": "FR",
36.     "point_of_sale": "9000001"
37.   }
38. }
```

1.1.7.5. Appel dans un contexte de paiement avec enregistrement de carte ou avec réutilisation d'une carte enregistrée avec restrictions sur le délai d'expiration, les cartes virtuelles, sous-réseau, type de compte, carte prépayée

```
1. {
2.   "action" : "InitializePaymentMean",
3.   "context": {
4.     "use_case": "Payment",
5.     "wallet_id": " monClientRef001"
6.   },
7.   "datetime": "2020-09-22T17:15:06",
8.   "merchant_configuration": {
9.     "configuration": "emulation3d",
10.    "language": "FR",
11.    "point_of_sale": "9000001"
12.  },
13.  "restrictions": {
14.    "expiration_delay": 3,
15.    "virtual_card": true,
16.    "subscheme":
17.    [
18.      "MAESTRO"
19.    ],
20.    "account_type": "individual",
21.    "prepaid_card": false
22.  }
23. }
```

1.1.7.6. Appel dans un contexte de paiement avec enregistrement de carte ou avec réutilisation d'une carte enregistrée avec restrictions sur le délai d'expiration

```
1. {  
2.   "action" : "InitializePaymentMean",  
3.   "context": {  
4.     "use_case": "Payment",  
5.     "wallet_id": "monClientRef001"  
6.   },  
7.   "datetime": "2020-09-22T17:15:06",  
8.   "merchant_configuration": {  
9.     "configuration": "emulation3d",  
10.    "language": "FR",  
11.    "point_of_sale": "9000001"  
12.  },  
13.  "restrictions": {  
14.    "expiration_delay": 3  
15.  }  
16. }
```

3.6. Retour du service « PaymentMeanTokenGenerator »

1.1.8. Structure du retour du service

Le service de demande de paiement par API retourne des informations au format JSON/UTF-8. Ce document comporte les paramètres suivants :

Champ JSON	Type JSON	Présence	Détails
entry_fields	Tableau d'objets	Conditionnelle	lien
error_message	Chaîne	Conditionnelle	lien
return_code	Entier	Obligatoire	lien
token	Chaîne	Conditionnelle	lien

Champ	entry_fields
Présence	Conditionnelle Renseigné dans le cas où l'objet « context » est fourni en entrée du service
Description	Liste des informations de moyen de paiement qui peuvent ou doivent être saisies pour obtenir un moyen de paiement valide et utilisable pour une demande de paiement ou de renseignement via l'API « PaymentService ». Correspond aux champs de saisie de type <i>hosted fields</i> pouvant être instanciés à la suite de la génération du jeton via le SDK « hostedfields.js » de Monetico Online
Format	Liste d'objets de type « entry_field »
Valeur(s) possible(s)	
Exemple(s)	<pre>[{ "field_type": "CardVx", "presence": "Required" }, { "field_type": "CardExpDate", "presence": "Required" }, { "field_type": "CardHolderName", "presence": "Required" }, { "field_type": "CardCvx", "presence": "Required" }]</pre>

Champ	error_message
Présence	Conditionnelle Présent si une erreur s'est produite lors de l'exécution du service
Description	Description de l'erreur retournée.
Format	Cf. la liste des valeurs ci-dessous.
Valeur(s) possible(s)	
Exemple(s)	Invalid MAC header.

Champ	return_code
Présence	Obligatoire
Description	Résultat de la demande de génération de jeton.
Format	Cf. la liste des valeurs ci-dessous.
Valeur(s) possible(s)	
Exemple(s)	0

Champ	token
Présence	Conditionnelle Présent si le jeton a été généré avec succès par le service
Description	Identifiant permettant la saisie temporaire d'un moyen de paiement. Valable 1h et utilisable, une fois la saisie du moyen de paiement complétée, sur l'API « PaymentService » pour effectuer des demandes de paiement ou de renseignement.
Format	UUID (RFC 4122)
Valeur(s) possible(s)	
Exemple(s)	eb2294de-a864-4a1f-abda-1b568bf7ca9a

1.1.8.1. Objet de type « entry_field »

L'objet de type « entry_field » représente une information à saisir par le porteur du moyen de paiement associé au jeton généré.

A chaque objet de type « entry_field » spécifié correspond un type de champ *hosted field* qui peut ou doit être instancié sur le parcours de paiement ou d'enregistrement de carte à la suite de la génération du jeton. Le type du champ *hosted field* correspondant peut être déterminé grâce au champ « field_type » décrit ci-dessous.

Attention : dans le cas où le contexte du parcours est précisé en entrée du service et que la liste des champs à saisir est fournie en sortie de celui-ci, il ne sera pas possible d'instancier un champ *hosted field* si celui-ci correspond à une donnée de moyen de paiement qui ne fait pas partie de la liste renvoyée par le service. La génération d'un tel champ demandée via le *SDK Hosted Fields* sera bloquée par Monetico Online.

Champ JSON	Type JSON	Présence	Détails
field_type	Chaîne	Obligatoire	lien
presence	Chaîne	Obligatoire	lien

Champ	field_type
Présence	Obligatoire
Description	Nom de la donnée du moyen de paiement qui pourra être saisie à l'intérieur du champ <i>hosted field</i> correspondant.
Format Valeur(s) possible(s)	Choix parmi : CardNumber, CardExpDate, CardCvx, CardHolderName, CardPreview, CardSwitch, CardRegistrationConsent
Exemple(s)	CardNumber

Champ	presence
Présence	Obligatoire
Description	Indique si la donnée de moyen de paiement correspondante doit obligatoirement être saisie ou non pour obtenir un moyen de paiement complet et valide
Format Valeur(s) possible(s)	Choix parmi : Required, Optional
Exemple(s)	Required

1.1.8.2. Champs « return_code » et « error_message »

Dans le cas où le jeton a été généré avec succès, les données renvoyées par le service sont les suivantes :

return_code	error_message	Description
0	Non renseigné	Les paramètres envoyés sont correctes et un jeton carte a été créé et associé à ce commerçant.

Dans le cas où une erreur s'est produite et que le jeton n'a pas été généré avec succès, voici les données qui peuvent être remontées par le service :

return_code	error_message	Description
-1	Technical error.	Un problème technique est survenu : réitérer la demande.
-3	Unauthenticated call	La demande ne peut pas être authentifiée, le résultat du calcul du MAC dans l'en-tête ne correspond pas à celui du serveur d'applications ou le TPE est inexistant.
-15	Invalid input parameters	Les paramètres transmis sont erronés. Cela peut être lié à la nature des paramètres d'entrée (syntaxe incorrecte) ou à l'absence des paramètres obligatoires.
-28	Unable to parse JSon body.	La syntaxe du corps de la requête est fausse et ne ressemble pas à un format JSON.
-29	Unable to create a payment mean token. It is possible that the merchant configuration does not allow this type of action.	La jetonisation a échoué, cela est souvent lié à un problème de configuration du TPE : il faut que la configuration commerçant permette la création de jetons.

1.1.9. Exemples de retour du service

1.1.9.1. Retour en succès sans contexte de parcours précisé en entrée

```
1. {
2.   "return_code": 0,
3.   "token": "3ab5278b-b6fa-439f-a83b-3fe5227e5f37"
4. }
```

1.1.9.2. Retour en succès dans un contexte d'enregistrement de carte

```
1. {
2.   "entry_fields": [
3.     {
4.       "field_type": "CardNumber",
5.       "presence": "Required"
6.     },
7.     {
8.       "field_type": "CardExpDate",
9.       "presence": "Required"
10.    },
11.    {
12.      "field_type": "CardHolderName",
13.      "presence": "Required"
14.    },
15.    {
16.      "field_type": "CardCvx",
17.      "presence": "Required"
18.    },
19.    {
20.      "field_type": "CardRegistrationConsent",
21.      "presence": "Required"
22.    }
23.  ],
24.  "return_code": 0,
25.  "token": "3ab5278b-b6fa-439f-a83b-3fe5227e5f37"
26. }
```

1.1.9.3. Retour en succès dans un contexte de paiement sans réutilisation de carte

```
1. {
2.   "entry_fields": [
3.     {
4.       "field_type": "CardNumber",
5.       "presence": "Required"
6.     },
7.     {
8.       "field_type": "CardExpDate",
9.       "presence": "Required"
10.    },
11.    {
12.      "field_type": "CardHolderName",
13.      "presence": "Required"
14.    },
15.    {
16.      "field_type": "CardCvx",
17.      "presence": "Required"
18.    }
19.  ],
20.  "return_code": 0,
21.  "token": "3ab5278b-b6fa-439f-a83b-3fe5227e5f37"
22. }
```

1.1.9.4. Retour en succès dans un contexte de paiement avec réutilisation d'une carte enregistrée

```
1. {
2.   "entry_fields": [
3.     {
4.       "field_type": "CardCvx",
5.       "presence": "Required"
6.     },
7.     {
8.       "field_type": "CardPreview",
9.       "presence": "Optional"
10.    },
11.    {
12.      "field_type": "CardSwitch",
13.      "presence": "Optional"
14.    }
15.  ],
16.  "return_code": 0,
17.  "token": "3ab5278b-b6fa-439f-a83b-3fe5227e5f37"
18. }
```

1.1.9.5. Retour en erreur

```
1. {
2.   "error_message": "Invalid input parameters.",
3.   "return_code": -15
4. }
```