



# **Intégrer Apple Pay en mode inApp via un bouton natif**

## **Guide d'implémentation**

Version du document 1.0

# Sommaire

1. PRÉREQUIS.....	3
2. PRINCIPE DE FONCTIONNEMENT.....	4
3. INITIER UNE DEMANDE DE PAIEMENT DEPUIS UN BOUTON NATIF.....	5
4. ANALYSER LA RÉPONSE À LA FIN DU PAIEMENT.....	7

## 1. PRÉREQUIS

---

Pour demander et traiter les paiements Apple Pay dans votre application, vous devez ajouter un bouton natif dans votre application.

Cet ajout se déroule en deux étapes :

1. [Configurez votre application pour remplir les conditions requises](#)
2. [Proposez le bouton natif Apple Pay dans votre application](#)

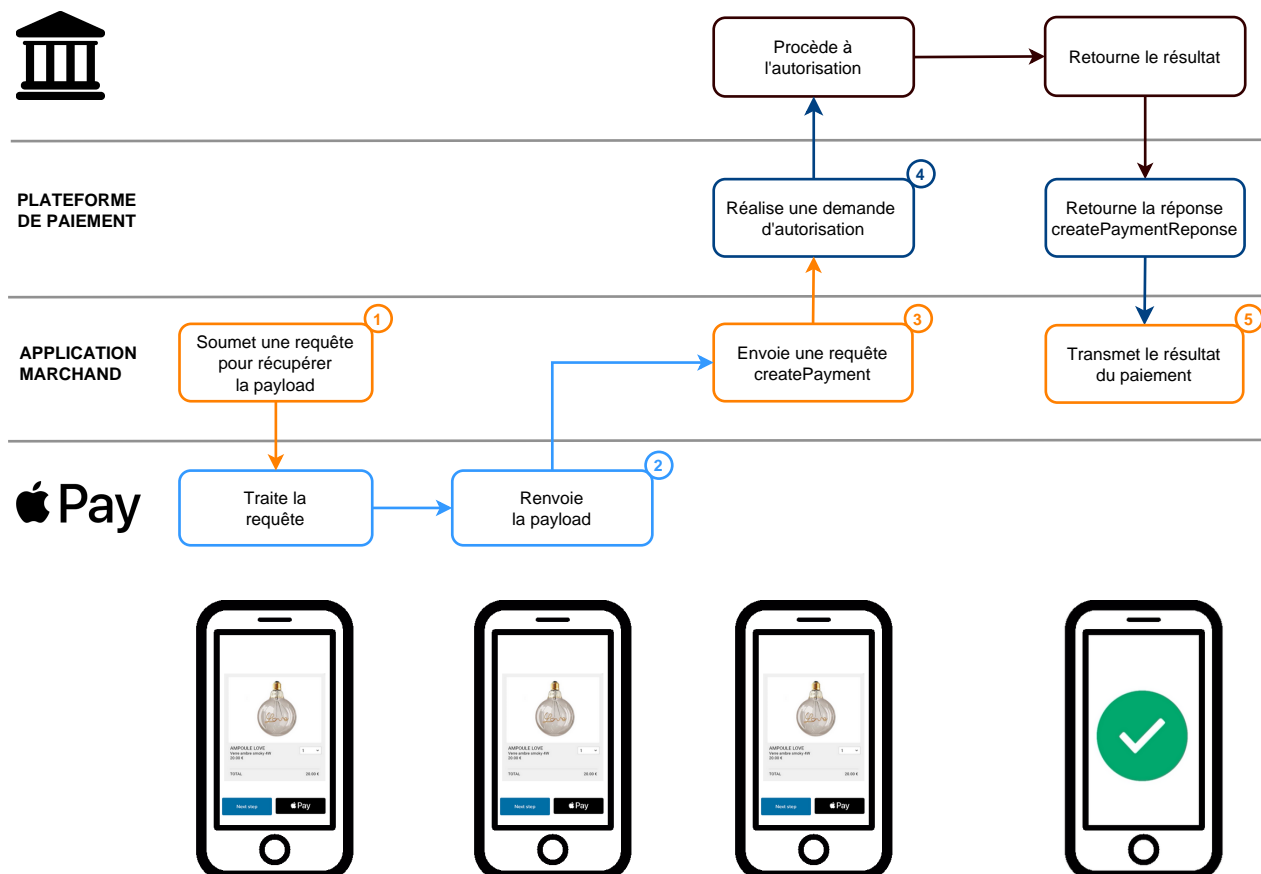
Référez-vous à la documentation complète dans [Apple Developer](#).

## 2. PRINCIPE DE FONCTIONNEMENT

Vous disposez de votre application et vous souhaitez proposer Apple Pay en bouton natif depuis cette application.

Cette documentation vous explique comment proposer un paiement via nos Web Services une fois ce bouton Apple Pay ajouté dans votre application.

Vous n'avez pas besoin d'être certifié PCI DSS.



1. Votre application soumet une requête à l'API Apple Pay pour collecter les données de la carte de l'acheteur.
2. L'API Apple Pay renvoie un objet token contenant les données chiffrées (appelé aussi *payload*).
3. L'application construit et soumet à la plateforme de paiement une requête `createPayment` en utilisant la *payload*.
4. La plateforme de paiement analyse la requête, déchiffre les données de carte et procède à la demande d'autorisation. Elle transmet ensuite le résultat du paiement à l'application.
5. L'application reçoit une réponse `createPaymentResponse` et analyse le résultat du paiement.

### 3. INITIER UNE DEMANDE DE PAIEMENT DEPUIS UN BOUTON NATIF

L'application du marchand reçoit un objet token d'Apple Pay. Cet objet contient la *payload* de paiement envoyée par Apple.

L'application initie une demande de paiement avec cette *payload* via un appel au Web Service `Charge/CreatePayment`.

**POST** `https://api.scelliuspaiement.labanquepostale.fr/api-payment/V4/Charge/CreatePayment`

En réponse, la plateforme de paiement renvoie un `formToken`, un objet encrypté avec les informations de la transaction et celles correspondant à la configuration de boutique.

Utilisez les champs ci-dessous pour construire la requête :

Nom du champ	Description	Requis
<code>amount</code>	Montant à payer, exprimé dans sa plus petite unité monétaire (le centime pour l'euro).	Oui
<code>currency</code>	Devise du paiement. Code ISO 4217 alpha-3. Ex: "EUR" pour l'euro	Oui
<code>orderId</code>	Référence de la commande.	Non
<code>payload.type</code>	Permet d'identifier le wallet Valorisé à <b>APPLE_PAY</b>	Oui
<code>payload.data.token.paymentData</code>	Contient les données de l'objet token envoyées par Apple. Ex : <code>(appleToken.paymentData)</code>	Oui
<code>payload.data.token.paymentMethod.displayName</code>	Contient les données de la carte envoyées par Apple. Ex : <code>(appleToken.paymentMethod.displayName)</code>	Oui
<code>payload.data.token.paymentMethod.network</code>	Contient les données du réseau de la carte envoyées par Apple. Ex : <code>(appleToken.paymentMethod.network)</code>	Oui
<code>payload.data.token.paymentMethod.type</code>	Contient les données du type de carte envoyées par Apple. Ex : <code>(appleToken.paymentMethod.type)</code>	Oui
<code>payload.data.token.transactionIdentifier</code>	Contient l'ID de la transaction envoyée par Apple. Ex : <code>(appleToken.transactionIdentifier)</code>	Oui
<code>ipnTargetUrl</code>	Permet de surcharger l'URL de notification instantanée (également appelée IPN) dans le formulaire.	Non

Nom du champ	Description	Requis
customer.email	Adresse e-mail qui sera utilisée pour l'envoi du ticket de paiement à l'acheteur.	Non
transactionOptions.cardOptions.captureDelay	Délai de remise.	Non
customer.reference	Identifiant de l'acheteur sur le site marchand.	Non
customer.billingDetails.firstName	Prénom de l'acheteur.	Non
customer.billingDetails.lastName	Nom de l'acheteur.	Non
customer.billingDetails.phoneNumber	Numéro de téléphone de l'acheteur.	Non
customer.billingDetails.address	Adresse de l'acheteur.	Non
customer.billingDetails.zipCode	Code postal.	Non
customer.billingDetails.city	Ville.	Non
customer.shippingDetails	Objet contenant le détail de l'adresse de livraison. <b>Cet objet et ses attributs sont recommandés.</b>	Non
customer.shoppingCart	Objet contenant le détail du panier. <b>Cet objet et ses attributs sont recommandés.</b>	Non
transactionOptions.cardOptions.manualValidat	Mode de validation du paiement.	Non



Pour obtenir une description plus complète des champs à utiliser, testez le Web Service [Charge/CreatePayment](#) depuis notre [playground](#).

### Exemple de requête

```
https://api.scelliuspaiement.labanquepostale.fr/api-payment/V4/Charge/CreatePayment
```

```
{
  "amount": 999,
  "currency": "EUR",
  "orderId": "myOrderId-930275",
  "payload": {
    "type": "APPLE_PAY",
    "data": {
      "token": {
        "paymentData": "(appleToken.paymentData)",
        "paymentMethod": {
          "displayName": "(appleToken.paymentMethod.displayName)",
          "network": "(appleToken.paymentMethod.network)",
          "type": "(appleToken.paymentMethod.type)"
        }
      }
    }
  }
}
```

## 4. ANALYSER LA RÉPONSE À LA FIN DU PAIEMENT

Le tableau suivant donne les informations principales permettant d'analyser la réponse à la fin d'un paiement via le wallet Apple Pay.

Pour obtenir plus de détails sur l'analyse de façon générale et des exemples de codes, consultez la page [Analyser le résultat du paiement](#).

<b>shopId</b> Description : Identifiant de la boutique. Ex : 12345678
<b>orderCycle</b> Description : Définit si la commande est ouverte ( <b>OPEN</b> , le paiement est autorisé), ou fermée ( <b>CLOSED</b> , elle ne peut plus être modifiée). Valeurs possibles : <ul style="list-style-type: none"><li>• <b>OPEN</b> : La commande est en cours. Vous devez conserver le panier car d'autres tentatives de paiement peuvent intervenir.</li><li>• <b>CLOSED</b> La commande est soit totalement payée, soit la dernière tentative a été refusée (dans ce cas, vous devez générer un nouveau formToken).</li></ul> Ex : CLOSED
<b>orderStatus</b> Description : Statut simplifié de la transaction. Valeurs possibles : <ul style="list-style-type: none"><li>• <b>PAID</b> : La commande est payée.</li><li>• <b>UNPAID</b> : La commande n'a pas été payée.</li><li>• <b>RUNNING</b> : La commande est en cours.</li></ul> Ex : RUNNING
<b>orderDetails.mode</b> Description : Mode de création de la transaction. Valeurs possibles : <ul style="list-style-type: none"><li>• <b>TEST</b> : pour une commande de test</li><li>• <b>PRODUCTION</b> : pour une commande réelle</li></ul> Ex : PRODUCTION
<b>orderDetails.orderId</b> Description : Référence de la commande. Ex : myOrderId-1234
<b>customer.reference</b> Description : Identifiant de l'acheteur sur le site marchand. Ex : MyReference-123456
<b>customer.email</b> Description : Adresse e-mail de l'acheteur. Ex : sample@example.com
<b>customer.billingDetails</b>

Description : Objet contenant le détail de l'adresse de facturation.

**customer.shippingDetails**

Description : Objet contenant le détail de l'adresse de livraison.

**customer.shoppingCart**

Description : Objet contenant le détail du panier.

**transactions.amount**

Description : Montant du paiement dans sa plus petite unité monétaire ( ( le centime pour l'euro) ).

Ex : 4525 pour 45,25 EUR

**transactions.uuid**

Description : Identifiant unique de la transaction généré par la plateforme de paiement.

Cet identifiant est ensuite utilisé pour la gestion des transactions par Web Service REST ou depuis le Back Office Marchand.

Ex : aab72593d0ce428ca6dd73ac7293d92d

**transactions.currency**

Description : devise du paiement selon la norme ISO 4217 alpha-3.

**transactions.detailedStatus**

Description : donne des informations détaillées sur le statut du paiement. Chacun des statuts est propre au moyen de paiement utilisé lors du paiement via Apple Pay.

Ex : AUTHORISED\_TO\_VALIDATE

Valeur possibles :

Valeur	Description
<b>ABANDONED</b>	<b>Abandonné</b> Paiement abandonné par l'acheteur. La transaction n'est pas créée et <b>n'est donc pas visible dans le Back Office Marchand.</b>
<b>AUTHORISED</b>	<b>En attente de remise</b> La transaction est acceptée et sera remise en banque automatiquement à la date prévue.
<b>AUTHORISED_TO_VALIDATE</b>	<b>À valider</b> La transaction, créée en validation manuelle, est autorisée. Le marchand doit valider manuellement la transaction afin qu'elle soit remise en banque. La transaction peut être validée tant que la date d'expiration de la demande d'autorisation n'est pas dépassée. Si cette date est dépassée alors le paiement prend le statut <b>EXPIRED</b> . Le statut <b>Expiré</b> est définitif.
<b>CANCELLED</b>	<b>Annulé</b> La transaction est annulée par le marchand.
<b>CAPTURED</b>	<b>Présenté</b> La transaction est remise en banque.
<b>CAPTURE_FAILED</b>	<b>La remise de la transaction a échoué.</b> Contactez le Support.
<b>EXPIRED</b>	<b>Expiré</b> Ce statut intervient dans le cycle de vie d'un paiement avec capture différée.



Valeur	Description
	La date d'expiration de la demande d'autorisation est atteinte et le marchand n'a pas validé la transaction. Le porteur ne sera donc pas débité.
<b>REFUSED</b>	<b>Refusé</b> La transaction est refusée.
<b>WAITING_AUTHORISATION</b>	<b>En attente d'autorisation</b> Le délai de remise en banque est supérieur à la durée de validité de l'autorisation. Une demande de renseignement (ou une autorisation à 1 EUR (ou demande de renseignement sur le réseau CB si l'acquéreur le supporte) si l'acquéreur ne supporte pas les demandes de renseignements) a été acceptée. La demande d'autorisation pour le montant total sera réalisée à J-1 avant la date de remise demandée. La remise en banque de la transaction est automatique.
<b>WAITING_AUTHORISATION_TO_VALIDA</b>	<b>A valider et autoriser</b> Le délai de remise en banque est supérieur à la durée de validité de l'autorisation. Une autorisation 1 EUR (ou demande de renseignement sur le réseau CB si l'acquéreur le supporte) a été acceptée. Le marchand doit valider manuellement la transaction afin que la demande d'autorisation et la remise aient lieu.

**transactions.paymentMethodType**

Description : moyen de paiement utilisé.

Ex : CARD

**transactions.transactionDetails.liabilityShift**

Description : Transfert de responsabilité.

Valeur possibles :

- **NO** : pas de Transfert de responsabilité sur la transaction.
- **YES** : Transfert de responsabilité applicable sur la transaction.

Ex : YES

**transactions.transactionDetails.wallet**

Description : wallet utilisé.

Valeur : APPLE\_PAY

**transactions.transactionDetails.cardDetails.effectiveBrand**

Description : Marque de la carte.

Ex : VISA

**transactions.transactionDetails.cardDetails.legacyTransId**

Description : Identifiant unique de la transaction généré par la plateforme de paiement.

Cet identifiant est ensuite utilisé pour la gestion des transactions depuis le Back Office Marchand.

Ex : 992364

**transactions.transactionDetails.cardDetails.manualValidation**

Description : Mode de validation de la transaction.

Valeur possibles :

- **NO** : Validation automatique par la plateforme de paiement.
- **YES** : Validation manuelle par le marchand.

Ex : NO

**transactions.transactionDetails.cardDetails.expectedCaptureDate**

Description : Date de remise en banque exprimée au format ISO 8601.

Ex : 2022-09-09T07:46:06+00:00

**transactions.transactionDetails.cardDetails.authorizationResponse.authorizationResult**

Description : Code retour de la demande d'autorisation.

Ex : 00

Consultez la [liste des codes d'autorisation](#) pour plus d'informations.