ApplePay - Intégration

Contenu

- Comment configurer votre compte ?
- Comment proposer ApplePay à vos clients ?
  - Les étapes sont les suivantes
- Les Fonctions de l'API DirectPayment avec paymentData
- Les Fonctions de l'API DirectPayment sans paymentData
- Comment réaliser des tests ?
- Logo du moyen de paiement
- Les codes de retour
- Les liens web
- Pages associées

Plus d'information

- Présentation du moyen de paiement

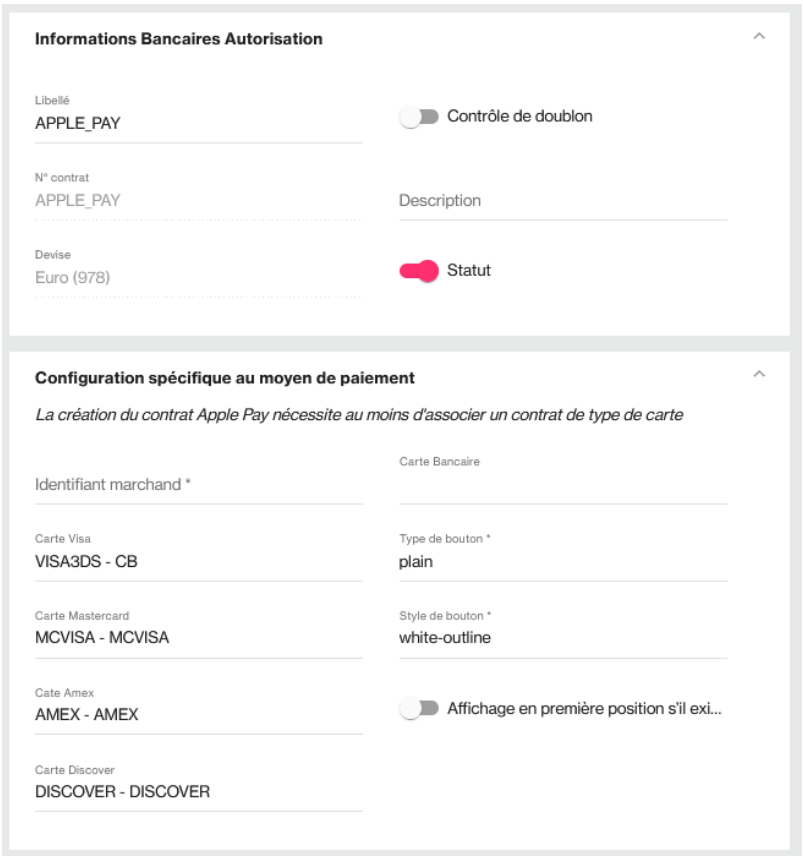
Comment configurer votre compte ?

Vous devez vous rapprocher de votre responsable de compte bancaire pour obtenir les informations nécessaires permettant de configurer votre moyen de paiement sur Monext.
Ensuite veuillez adresser un email à notre équipe support avec les informations (MarchandID du compte Apple Pay et la liste des contrats VAD associés) , afin qu'ils puissent procéder au paramétrage dans votre compte marchand Monext en indiquant l'ID Marchand et le nom de l'alias du contrat partenaire à créer.
Vous devez réaliser des transactions 'pilote' pour valider le bon fonctionnement en production.

Contactez le support Monext Online.

Dans le Centre administration de Monext, lors de la création du moyen de paiement, les champs à configurer sont les numéros de contrats Visa, Mastercard, Discover.

Le code (card_code) du moyen de paiement est : APPLE_PAY



Gestion des certificats

Lors de la création veuillez consulter la gestion des certificats ici.

Les options

- Vous pourrez sélectionner les options d'affichages du type du bouton et du style du bouton Apple Pay ;
- Configurez l'affichage d'Apple Pay afin qu'il soit systématiquement en première position si l'acheteur a déjà utilisé Apple Pay.

Réseau CB

Le moyen de paiement vous permet de traiter les transactions de cartes bancaire CB sur le réseau CB simplement

- en configurant votre moyen de paiement : paramètre Carte Bancaire ci-dessus.
- en alimentant la valeur du service de paiement : card.PaymentData.Network = **cartesBancaires**

Pour les autres réseaux

- VISA = **visa**
- MASTERCARD = **masterCard**
- CB = **cartesBancaires**

Voir documentation Apple :https://developer.apple.com/

Configuer l'EDI XCode d'Apple

Lorsque vous ouvrez un projet, il faut paramétrer le marchandID. Cet identifiant sera fourni par Monext.

La validation des domaines n'est pas systématiquement demandée par Apple. Cette étape est à effectuer côté Monext et sur votre compte Apple developer.
Pour configurer le contrat Apple pay en homologation, il faut juste nous transmettre merchant name et nous préciser le point de vente sur lequel nous devons générer les certificats, puis nous vous les transmettrons.

Comment proposer ApplePay à vos clients ?

Ce moyen de paiement est disponible avec l'API WebPayment (mode Widget uniquement) et l'API DirectPayment.

Deux intégrations possibles

- En App native : le commerçant récupère les info d'ApplePay et il réalise un doAuthorization depuis son SI avec les informations d'ApplePay. Le commerçant peut également décrypter le PaymentData d'ApplePay et réaliser la demande de paiement à Monext avec la carte bancaire.
- En Web/App avec webview : le commerçant réalise un doWebPayment sur un contrat ApplePay de Monext. Le consommateur utilise son Iphone, Ipad, ou un Mac avec un TouchID intégré (ou un Mac avec son Iphone à côté du Mac: Safari se charge de faire le lien).

Les étapes sont les suivantes

1. Le commerçant doit activer le moyen de paiement sur centre administration Monext.

Sur AppMobile (mode API DirectPayment) avec le certificat géré par Monext

Lorsque vous sélectionnez l'option 2 dans la création du moyen de paiement : Apple Pay - Création des éléments sur Monext Online et Apple Pay

2. Création d'une application mobile enregistrée dans l'AppleStore pour permettre aux consommateurs de sélectionner les produits et/ou services ;

3. L'application mobile doit proposer le bouton ApplePay et la page de paiement avec le TouchID en intégrant le PassKit du SDK d'ApplePay avec les serivces canMakePayments et ApplePaySession ;

4. Le commerçant utilise le service prepareSession pour créer la session de paiement côté ApplePay à partir du certificat idendity Monext.

   - Monext appelle le service MerchantValidation d'Apple Pay : pour validez l'identité marchand et recevoir un objet de session pour chaque demande de paiement.
   - Le commerçant transmet les paymentTokenData en retour de Monext vers le javascript d'ApplePay.
     - https://developer.apple.com/documentation/apple_pay_on_the_web/apple_pay_js_api/providing_merchant_validation
   - Ensuite, lors du paiement de l'acheteur, le marchand récupère les données de paiement chiffrées en provenance d'Apple Pay.
   - Le commerçant peut transmettre ces données à Monext via le service doAuthorization.

5. Monext décrypte le PaymentData et réalise la transaction puis retourne le résultat au serveur commerçant en temps réel ;
6. Le serveur commerçant informe l'application mobile qui affiche le résultat de l'opération.

Validation URL APPLE PAY à appeler

L'URL du service Apple Pay à utiliser diffère en fonction du contexte d'utilisation d'Apple Pay. Elle est transmise par l'IOS/Safari Apple Pay au commerçant qui le transmet à Monext dans le champ miscData.

Exemple miscData envoyé par un commerçant à Monext dans la request

```
{
   "displayName:"MyStore",
   domainName:"mystore.example.com",
   validationURL:"https://apple-pay-gateway.apple.com/paymentservices/paymentSession""
}
```

Exemple de champ data envoyé par Monext dans la response

```
{
    "epochTimestamp": 1665645679253,
    "expiresAt": 166564564569253,
    "merchantSessionIdentifier": "SSH690EA65CBF4568E40D09BA7CD.............",
    "nonce": "7dc980d2",
    "merchantIdentifier": "378E33CF6B48CA390B0D4572820597456ED66F4844C1307CAFCCF7F7B9",
    "domainName": "webpayment.dev.payline.com",
    "displayName": "Demo_REST",
    "signature": "308006092a6587416f70d010.................",
    "operationalAnalyticsIdentifier": "DEMO_REST:378E336B48CA39...........",
    "retries": 0,
    "pspId": "378E38CA390.........."
}
```

Exemple du service prepareSessionRequest

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:impl="http://impl.ws.
payline.experian.com">
   <soapenv:Header/>
   <soapenv:Body>
      <impl:prepareSessionRequest>
         <impl:version>22</impl:version>
         <impl:contractNumber>ApplePayTest08S</impl:contractNumber>
         <impl:orderRef>ODERrEF2222</impl:orderRef>
         <impl:miscData>{ "ApplePayTest08S" : "{\"displayName\":\"nameToDisply\",\"domainName\":\"webpayment.
dev.payline.com\",
\"validationURL\":\"https:\/\/apple-pay-gateway-cert.apple.com\/paymentservices\/startSession\"}"}</impl:
miscData>
      </impl:prepareSessionRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Exemple du service prepareSessionResponse

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Header/>
   <soapenv:Body>
      <impl:prepareSessionResponse xmlns:impl="http://impl.ws.payline.experian.com" xmlns:obj="http://obj.ws.
payline.experian.com">
         <impl:result>
            <obj:code>00000</obj:code>
            <obj:shortMessage>ACCEPTED</obj:shortMessage>
            <obj:longMessage>Transaction approved</obj:longMessage>
         </impl:result>
         <impl:data>{"epochTimestamp":1675935416164,"expiresAt":1675939016164,"merchantSessionIdentifier":"
SSHDE262B37C6824EA9A07290DE5A943F0F94927C24",
"nonce":"746ed113","merchantIdentifier":"73630A6EC0599D8245A8DD657A0B","domainName":"webpayment.dev.payline.
com",
"displayName":"nameToDisply","signature":"3080060928648ce..d079cfaa000000000000",
"operationalAnalyticsIdentifier":"nameToDisply:73630A6EC059E2915DD82D8245A8DD657A0B","retries":0,"pspId":"
B480C6D24121D1C2C5FFFF37B6"}
</impl:data>
      </impl:prepareSessionResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

Sur AppMobile (mode API DirectPayment) avec le certificat géré par le commerçant

2. Création d'une application mobile enregistrée dans l'AppleStore pour permettre aux consommateurs de sélectionner les produits et/ou services ;

3. L'application mobile doit proposer le bouton ApplePay et la page de paiement avec le TouchID en intégrant le PassKit du SDK d'ApplePay ;

4. L'application mobile récupère une chaine de caractère PaymentData en retour du SDK Apple.
   Ensuite l'application le transmet au serveur commerçant, qui réalise la demande de paiement à Monext avec le service doAuthorization ;

5. Monext décrypté le PaymentData et réalise la transaction puis retourne le résultat au serveur commerçant en temps réel ;

6. Le serveur commerçant informe l'application mobile qui affiche le résultat de l'opération.

Sur une page web (mode Widget uniquement)

2. Intégration du bouton ApplePay avec la librairie ApplePay.JS du SDK ApplePay sur la page web checkout en respectant les directives d'ApplePay ;

3. La page web propose doit proposer la page de paiement avec le TouchID ;

4. Le navigateur web récupère le PaymentData, qui est transmis au serveur commerçant. Ce dernier réalise une demande de paiement doAuthorization à Monext ;

5. Monext décrypté le PaymentData et réalise la transaction puis retourne le résultat au serveur commerçant en temps réel ;

6. Le serveur commerçant affiche le résultat au consommateur.

Les fonctionnalités de paiement disponibles :

- paiement à la commande : en utilisant le service doWebPaymentRequest / doAuthorization avec payment.mode à CPT et payment.action à 101 ;
- paiement à l'expédition : en utilisant les services doWebPaymentRequest / doAuthorization et doCapture avec payment.mode à CPT et payment.action à 100 ;
- paiement en différé : en utilisant les services doWebPaymentRequest / doAuthorization et doCapture avec payment.mode à DIF et payment.action à 100 ;
- paiement en N fois ou récurrent : en utilisant le services createWallet et en fonction du besoin un des services doImmediateWalletPayment, doScheduledWalletPayment, doRecurrentWalletPayment ;
- demande d'annulation d'une autorisation non encaissée : utilisez la fonction doReset ;
- demande de remboursement d'une transaction remisée : utilisez la fonction doRefund.

Les Fonctions de l'API DirectPayment avec paymentData

L'objet Card contient l'élément PaymentData. Cet élément est disponible en entrée des services de l'API :

- doAuthorizationRequest.card.paymentData
- createWalletRequest.wallet.card.paymentData
- updateWalletRequest.wallet.card.paymentData

Un commerçant qui souhaite utilisé le mode direct, doit remplir les champs du doAuthorization suivant avec les informations récupérées dans le PAYLOAD ApplePay en plus des champs standards Monext.

Payload Apple Pay	doAuthorizationRequest	Commentaires
paymentNetwork	card.PaymentData.Network	Pas de transformation : Réseaux supportés: VISA / MASTERCARD.
applicationPrimaryAccountNumber	card.number	Pas de transformation : Token PAN contenu dans le PAYLOAD déchiffré par le commerçant.
applicationExpirationDate	card.expirationDate	A convertir au format MMYY.
currencyCode	payment.currency	Code devise numérique ISO 4217.
transactionAmount	payment.amount	Plus petite unité de la devise (100 = 1€).
cardholderName	card.cardholder	Pas de transformation.
paymentData. onlinePaymentCryptogram	authentication3DSecure. cavv	Pas de transformation : Apple Pay fournit systématique le CAVV. Dans le cas où le CAVV ne serait pas transmis par le commerçant, Monext rejettera la demande de paiement.
paymentData.eciIndicator	authentication3DSecure.eci	Pas de transformation : Apple Pay ne fournit pas systématiquement l'ECI. Le commerçant doit renvoyer la valeur tel quel.

Les Fonctions de l'API DirectPayment **sans** paymentData

La solution proposée consiste à faire passer les informations déchiffrées par le commerçant à travers les doAuthorizationRequest dans les objets suivants:

- Object - card
- Object - authentication3DSecure

Le commerçant devra renseigner dans l'Object payment le **contract number** du type ApplePay correspondant.

Le moyen de paiement prend en compte les champs ci-dessous **si le champ card.paymentData.TokenData est vide** dans le doAuthorizationRequest .

Mapping du Payload d'ApplePay

Payload XPay	doAuthorizationRequest	Obligatoire	Commentaires
Réseau de la carte	card.PaymentData.Network	OUI	Donnée déchiffrée dans le PAYLOAD tel quel sans transformation.
Token ou PAN de la carte	card.number	OUI	
Date d'expiration	card.expirationDate	OUI	
-	card.panType	NON	Positionner la valeur "DEVICE_TOKEN" Valeur recommandée, permet d'identifier les transactions effectuées avec un token XPay
Devise	payment.currency	OUI	Ce champ n'est pas forcément présent dans le PAYLOAD. Dans ce cas, le commerçant doit le renseigner lui même.

Montant	payment.amount	OUI	Ce champ n'est pas forcément présent dans le PAYLOAD. Dans ce cas, le commerçant doit le renseigner lui même.
Titulaire de la carte	card.cardholder	NON	Ce champ n'est pas forcément présent dans le PAYLOAD. Dans ce cas, le commerçant doit le laisser vide.
Cryptogram	authentication3DSecure. cavv	NON	Ce champ n'est pas forcément présent dans le PAYLOAD. Dans ce cas, le commerçant doit le laisser vide.
ECI	authentication3DSecure.eci	NON	Ce champ n'est pas forcément présent dans le PAYLOAD. Dans ce cas, le commerçant doit le laisser vide.

Comment réaliser des tests ?

Les cartes Visa et Mastercard d'une banque partenaire d'Apple seront acceptées par ce moyen de paiement.
Apple met à disposition une plateforme de test « Sandbox » avec des jeux de tests : https://developer.apple.com/support/apple-pay-sandbox/

Note : Il est indispensable de réaliser des tests transactionnels sur l'environnement de production avec des cartes de crédit et de débit après la validation de la mise en œuvre sur l'environnement de test.

Logo du moyen de paiement

Apple propose un lien contenant toutes les normes à respecter : https://developer.apple.com/apple-pay/Apple-Pay-Identity-Guidelines.pdf

Les codes de retour

Monext Online vous informe du résultat d'un paiement via le ShortMessage, selon le mode d'intégration proposé par le moyen de paiement.

- pour l'API WebPayment avec le service getWebPaymentDetails,
- pour l'API DirectPayment en réponse du service doAuthorization de manière synchrone ou getTransactionDetails en réponse asynchrone.

Les états retournés :

- Le paiement est accepté avec l'état ACCEPTED et le code retour 00000.
- Le paiement est refusé avec l'état REFUSED. Le code varie en fonction du motif de refus (Par exemple : 01xxx pour une raison bancaire ou 04xxx pour une suspicion de fraude).

La gestion des états et des codes retour sont listés ici.

Les codes sont identiques à ceux utilisés par les moyens de paiement VISA, Mastercard, Discover.

Les liens web

Voici les liens très utiles de la présentation d'ApplePay :

- Apple Pay developer page where all the main documentation and guidelines for Apple Pay in-app are freely available.
- Video of our tech team doing a deep dive on Apple Pay (in-app) @ WWDC. There are some good tips there.
- Link to our Apple Pay identity guidelines for in-app.

Plus d'information commerciale : https://www.apple.com/apple-pay/

Developer Website : https://developer.apple.com

Pages associées

- Intégration 3D Secure 2 en mode API WebPayment
- Intégration API WebPayment
- PW - Intégration Widget
- PW - L'API JavaScript
- PW - Personnalisation du widget : Balises CSS
- PW - Personnalisation du widget : Feuille de style
- PW - Personnalisation du widget : Fonction CallBack