



API JavaScript

Afin d'interagir avec le formulaire de paiement, Payline propose une API en JavaScript pour que vous gardiez la main sur le formulaire. Les fonctions disponibles depuis votre SI sont proposées avec la librairie **widget-min.js** :

Fonction	Description	Exemple
endToken()	Met fin à la vie du jeton de session web (token obtenu via DoWebPayment). Les paramètres : (additionalData : any, callback : Function, spinner : any, handledByMerchant : boolean)	<pre>Payline.Api.endToken(null, function() { window.location.href = Payline.Api. getCancelAndReturnUrls().cancelUrl; }, null, true);</pre> <pre>// en valorisant handledByMerchant à true, // la session de paiement sera associée au code retour 02014 // renvoyé par getWebPaymentDetails. // Dans le cas contraire c'est le code 02319 qui est renvoyé. // Il est conseillé de contrôler la présence du token session avant l'appel // de la fonction.</pre>
finalizeShortcut()	Déclenche la finalisation du paiement Shortcut (à appeler en context. state PAYMENT_TRANSITIONAL_SHORTCUT)	
getBuyerShortcut()	Renvoie un objet complexe contenant les données de l'acheteur provenant du partenaire shortcut. Cette fonction est disponible dans un contexte de state = PAYMENT_TRANSITIONAL_SHORTCUT.	<pre>{ "firstName": "DUPONT", "lastName": "Jean", "email": "jean.dupont@yopmail.com", "cityName": "Aix", "country": "FR", "street1": "5 rue de Marseille", "street2": "Batiment 2", "zipCode": "13390" }</pre>
getCancelAndReturnUrls()	Renvoie un objet complexe contenant la returnUrl et la cancelUrl provenant du DoWebPayment	On peut par exemple créer un bouton annuler qui déclenche les actions suivantes : <pre>function cancel() { Payline.Api.endToken(); // met fin à la vie du token window.location.href = Payline.Api.getCancelAndReturnUrls(). cancelUrl; // redirige l'utilisateur sur la cancelUrl</pre>

getContextInfo(key)	Renvoie une information du contexte grâce à sa clé parmi les valeurs suivantes :	<pre>{ "PaylineAmountSmallestUnit": 100, "PaylineCurrencyDigits": 2, "PaylineCurrencyCode": "EUR", "PaylineBuyerFirstName": "Jean", "PaylineBuyerLastName": "DUPONT", "PaylineBuyerShippingAddress.street2": "Batiment 2", "PaylineBuyerShippingAddress.country": "FR", "PaylineBuyerShippingAddress.name": "Perso", "PaylineBuyerShippingAddress.street1": "5 rue de Marseille", "PaylineBuyerShippingAddress.cityName": "Aix", "PaylineBuyerShippingAddress.zipCode": "13390", "PaylineBuyerMobilePhone": "0605040302", "PaylineBuyerShippingAddress.phone": "0605040302", "PaylineBuyerIp": "127.126.125.100", "PaylineFormattedAmount": "EUR1.00", "PaylineOrderDate": "05/26/2008 10:00", "PaylineOrderRef": "170630111901", "PaylineOrderDeliveryMode": "4", "PaylineOrderDeliveryTime": "1", "PaylineOrderDetails": [{ "ref": "1", "price": 998, "quantity": 1, "comment": "commentaire1", "category": "1", "brand": "66999", "subcategory1": "", "subcategory2": "", "additionalData": "" }], }</pre>
getCssIframeWhiteList()	Renvoie la liste des primitives CSS acceptées pour styliser les champs protégés (Carte bancaire, CVV, ...)	getCssIframeWhiteList(): any
getFragmentedPaymentInfo()	Dans un contexte de paiement fragmenté (Carte Cadeau Carrefour, prochainement ANCV, ...), permet de récupérer les différents paiements enregistrés jusqu'à maintenant. Vous pouvez récupérer les données lorsque vous êtes en mode tab ou col.	<pre>{ "amount": "100,00 EUR", // Montant global de la commande "amountLeft": "50,00 EUR", // Montant restant à payer "recaps": [503200XXXXXXXXXX1166: "50,00 EUR"] }</pre>
getLanguage()	Renvoie la clé du langage du widget (passé dans la trame DoWebPayment)	"en", "fr", ...

getOrderInfos()	Permet de récupérer un objet complexe contenant: - le montant avec la devise le tout formaté - la référence commande - le nom du point de vente - un objet complexe représentant l'adresse avec : - adresse 1 - adresse 2 - ville - code postal	<pre>Object { Payline.Api.getOrderInfos() on obtient : amountCurrency: "15.00 EUR" orderRef: "order-123456789" pointOfSale: "POS_Demo" pointOfSaleAddress: Object { address1: "260 rue Claude Nicolas Ledoux" address2: "ZI Les Milles" city: "Aix en Provence" zipCode: "13290" }}</pre>
getRecurringDetails()	Permet de récupérer l'échéancier du paiement. Voir le détail d'intégration : paiement N fois et paiement récurrent.	Pour afficher l'échéancier N fois : <pre>Payline.Underscore.forEach(Payline.Api. getRecurringDetails().schedules, function (key, value) { console.log(value + " " + key) });</pre> Pour récupérer les détails du paiement récurrent : <pre>Payline.Api.getRecurringDetails(). schedules;</pre>
getToken()	Retourne le token courant avec lequel le Widget a été initialisé.	getToken()
getTokenStatus()	Permet de connaître l'état de la session de paiement (Token de session) afin de renvoyer une demande de paiement sans recréer une nouvelle commande et éviter les problèmes de multiples sessions pour une commande. La fonction getTokenStatus avec l'identifiant de session de paiement en paramètre et retourne les états : ALIVE, EXPIRED, UNKNOWN.	<pre>Payline.Api.getTokenStatus ('1Iy8gHkAi6AtplkmM1381551350523598', show) function show(tokenStatus){ console.log('Le statut du token:'); console.log(tokenStatus); }</pre>
hide()	Demande de masquage du formulaire de paiement.	Payline.Api.hide(): void
init()	Demande d'initialisation du formulaire de paiement. Lors de l'initialisation, le widget Payline construit le formulaire en fonction du contexte de la demande de paiement: montant, moyen de paiement proposé, etc.	
isSandbox()	Permet de connaître l'environnement : production ou homologation. La fonction retourne true ou false.	Payline.Api.isSandbox();

reset()	Réinitialise totalement le widget avec le token et le template passé en paramètre	<pre>Payline.Api.reset(token: string = null, template: string = null): void</pre>
show()	Demande d'affichage du formulaire de paiement.	<pre>Payline.Api.show(): void</pre>
toggle()	Si le widget est affiché, cette méthode le cache, sinon elle l'affiche	
updateWebpaymentData()	Mise à jour des informations de la session de paiement (adresses, montant,...) après l'initialisation du widget et avant la finalisation du paiement. Cette fonction est validée seulement si l'option de mise à jour d'une commande au niveau du point de vente est activée (voir le lien du centre admin).	<pre>Payline.Api.updateWebpaymentData(Payline.Api.getToken(), { "payment": { "amount": 100, "currency": "978" } }, function (response) { if(response.status !== 200) { alert("Error occured at Payline.Api.updateWebpaymentData (code "+response.status+""); } });</pre> Ensuite il faut regarder dans network de chrome, quand il y a une erreur on peut obtenir la réponse. <pre>{ "payment": { "amount": "252", "currency": "978" }, "order": { "ref": "toto", "amount": "255", "currency": "978", "deliveryTime ": "1", "deliveryMode ": "1", "deliveryExpectedDate": "30/09/2016", "deliveryExpectedDelay": "1", "taxes": "10", "deliveryCharge": "10" "details": { "details": [{ "ref": "00001 (ref1)", "price": "50", "quantity": "1", "comment": "comment1 (rem1)", "category": "1", "brand": "My HERMES (brd1)", "subcategory1": "BOSS", "subcategory2": "MONTRE", "additionalData": " LegalAge=18;Discount=0105", "taxRate ": "1234" }, {</pre>

```
(ref2)",
"comment2 (rem2)",
HERMES (brd2)",
"HERMES",
"FOULARD",
LegalAge=18;Discount=0105",
"ref": "00002",
"price": "50",
"quantity": "1",
"comment":
"category": "1",
"brand": "My
"subcategory1":
"subcategory2":
"additionalData":
"taxRate ": "1234"
}, {
"ref": "00003",
"price": "50",
"quantity": "1",
"comment":
"category": "1",
"brand": "My
"subcategory1":
"subcategory2":
"additionalData":
"taxRate ": "1234"
}, {
"ref": "00004",
"price": "50",
"quantity": "1",
"comment":
"category": "1",
"brand": "My
"subcategory1":
"subcategory2":
"additionalData":
"taxRate ": "4321"
}, {
"ref": "00005",
"price": "51",
"quantity": "1",
"comment":
"category": "1",
"brand": "My
"subcategory1":
"subcategory2":
"additionalData":
"taxRate ": "1234"
}
]
}
```

```

    },
    "buyer": {
        "firstName": "Prénom",
        "lastName": "Nom",
        "email": "myemail@mail.com",
        "mobilePhone": "+033600000000",
        "shippingAddress": {
            "title": "2",
            "name": "BUYER (shipping)",
            "lastName": "PAYLINE
(shipping)",
            "firstName": "Validation
Team (shipping)",
            "street1": "Colombus Circle
(shipping)",
            "street2": "Central Park
(shipping)",
            "cityName": "New York
(shipping)",
            "zipCode": "NY 10023",
            "country": "FR",
            "phone": "0123458789",
            "state": "NYC (state)",
            "county": "Upper East Side
(shipping)",
            "phoneType": "1"
        },
        "billingAddress": {
            "title": "3",
            "name": "BUYER (billing)",
            "lastName": "NAME (billing)",
            "firstName": "FirstName
(billing)",
            "street1": "15 rue LEDOUX
(billing)",
            "street2": "Z.I. les Milles
(billing)",
            "cityName": "Aix-en-
Provence (billing)",
            "zipCode": "13902",
            "country": "FR",
            "phone": "0223458789",
            "state": "FRANCE (state)",
            "county": "Bouches-du-
Rhône",
            "phoneType": "4"
        }
    }
}
}

```

Cas d'utilisation

Exemple pour masquer, puis afficher le formulaire de paiement :

- `Payline.Api.hide(): void`
- `Payline.Api.show(): void`

Exemple pour changer le libellé du bouton Payer :

- Vous pouvez le modifier en utilisant l'instruction JavaScript `Payline.jQuery (". PI-pay-btn"). Texte (' étiquette du bouton ');`

Exemple pour changer le libellé du bouton Payer :

- Vous devez récupérer le montant dans la langue de l'acheteur via un appel à la fonction `Payline.Api.getContextInfo ('PaylineFormattedAmount') ;`

- Vous pouvez modifier le texte du bouton avec la fonction : `Payline.jQuery("<classe de l'élément à modifier>").Texte("<texte de remplacement>");`

⚠ Attention, utilisez toujours les classes pour ces instructions, jamais l'id de l'élément qui est généré à la volée

- Pour afficher le montant dans le bouton. L'instruction est donc `Payline.jQuery(".paylineWidget .pl-pay-btn, .PaylineWidget .pl -btn").Texte("Payer" + Payline.Api.getContextInfo("PaylineFormattedAmount"));`
- Cette instruction doit être appelée lorsque le widget déclenche l'événement « `didshowstate` » avec l'état `PAYMENT_METHODS_LIST`.

Pour obtenir plus d'information de personnalisation JQuery ou CSS, [veuillez trouver les informations ici](#).

Pages associées

- [Mode in-Shop](#)
- [Mode Lightbox](#)
- [Mode Shortcut](#)
- [Paiement Page Web](#)