Intégration technique



AXEPTA API

AXEPTA est une solution de paiement basée sur une API HTTPS POST / GET utilisant des paramètres en méthode NVP (Name-Value-Pairs) et des objets JSON.

La sécurisation des échanges est effectuée via l'usage d'authentification HMAC et l'utilisation de cryptage blowfish.

Cette section aborde les points suivants

- Ressources pour développeurs
- Sécurisation
 - Authentification HMAC
 - Blowfish ECB
- Construire une requête de paiement
 - Concepts
 - Paramètres
 - o Etapes de création d'une requête de paiement
 - Réponse
- Exemple pas à pas

Ressources pour développeurs

La section Ressources pour développeurs regroupent les éléments vous permettant de réaliser l'intégration technique d'Axepta Online.



Intégration technique - Ressources pour développeurs

Bases de l'intégration technique Créer un appel API Authentification HMAC (Requête) Authentification HMAC (Notify) Créer une requête de paiement pas à pas Demandes de statut (Inquire.aspx) Sources - Aide à l'implémentation Template XSLT (.zip) Third-party-cookies -> Browser cookies and session handling

La section Liste des données Axepta regroupe les données utilisées par la plateforme : codes retours, données 3DS, objets JSON



Données Axepta

Acronymes et abréviations <u>Tableau des devises Codes de pays Paramètres AVS Codes retour Données 3DS Liste des objets</u>
JSON Paramètres

Sécurisation

Authentification HMAC

Pour vous protéger contre toute manipulation non autorisée de vos transactions de paiement, la plate-forme Axepta vérifie à l'aide d'un code d'authentification de message haché (HMAC) si votre demande de paiement est authentique et n'a pas été modifiée. Pour cela, vous transférez une valeur HMAC à la Plateforme à chaque transaction dans le paramètre MAC.

La plateforme Axepta utilise un Hash Message Authentication Code (HMAC) pour vérifier l'authenticité de vos paiements. L'algorithme MAC SHA-256 est utilisé avec une longueur de clé à 32 chiffres (256 bits).

Pour plus de détails : HMAC Authentication (Request) et HMAC Authentication (Notify)

Blowfish ECB

Blowfish est un algorithme de chiffrement symétrique (c'est-à-dire « à clef secrète »).

Pour faciliter votre intégration, vous trouverez ci-dessous quelques exemples de librairies Blowfish ECB

Techno	Exemples
ASP	txmsCrypto.dll // txmsCrypto.BlowFish
ASP.NET	Computop.Core.Crypto.dll // Axepta.Core.Crypto.BlowFish
Java	Blowfish.java
PHP	function.inc.php ctHMAC ctEncrypt ctDecrypt

Construire une requête de paiement

Concepts

L'intégration de la solution de paiement Axepta se base principalement sur un concept de construction de requête de paiement dont les principes sont les suivants :

- Gestion des paramètres en méthode NVP (Name-Value-Pairs)
- Utilisation d'objets JSON
- Le calcul d'un HMAC
- Une chaîne de caractères correcte contient trois paramètres de base : MerchantID (Identifiant du commerçant), Len (Longueur) et Data (Données). Les paramètres MerchantID et Len ne sont pas chiffrés. Seul le paramètre Data est chiffré avec la méthode Blowfish

Paramètres

- Le paramètre Data (Données) comprend les détails de paiement essentiels comme le montant et la devise.
- Le paramètre Len (Longueur) est très important pour le chiffrement, car il contient la longueur de la chaîne de caractères non chiffrée dans le paramètre Data. La quantité de données à chiffrer étant multipliée par 8 dans le cas du chiffrement Blowfish, la longueur correcte de la chaîne de caractères doit être connue pour le déchiffrement, sans quoi d'autres caractères non prévus apparaissent à la fin de la chaîne de caractères.

Les paramètres sont transmis via HTTPS POST ou HTTPS GET.

La méthode de transmission recommandée est HTTPS POST, car la chaîne de caractères du paramètre dans le cas de GET, jointe à l'URL, est limitée à 2 048 octets selon le navigateur, contrairement à la méthode POST qui n'est pas limitée par la taille de l'URL.

Etapes de création d'une requête de paiement

Les étapes de création d'une requête sont :

- Calcul du HMAC pour sécuriser le montant et la devise cf. HMAC Authentication (Request)
- Construire les objets JSON et les encoder en Base64 avec padding cf. Fonctionnalités de paiement
- Assembler les paramètres (clé / valeur, objets JSON) de l'API
- Chiffrer tous les paramètres de l'API avec la clé Blowfish : cela permettra d'obtenir les paramètres Data et Len
- Si besoin, ajouter des paramètres simples pour personnaliser la page de paiement hébergée par (par exemple language="en" pour utiliser la langue anglaise, les customFields)
- Envoyer la demande d'API au endpoint choisi

Axepta Online utilise les méthodes POST et GET pour rediriger l'utilisateur vers le site e-commerce du marchand ou envoyer la notification du résultat du paiement

Moyen de paiement choisi par l'acheteur	Format des réponses
Paiements carte	Méthode POST pour l'URLFailure / URLSuccess / URLNotify
	Méthode GET pour l'URLFailure / URLSuccess / URLNotify en cas de fallback 3DSV1
Moyens de paiement alternatifs	Méthode GET pour l'URLFailure / URLSuccess / URLNotify

Exemple pas à pas



La page Créer une requête de paiement pas à pas permet de réaliser un premier paiement via la boutique de démo BNP_DEMO_AXEPTA. Cela peut être une première étape avant d'utiliser votre MID.