Guide d'intégration

Envoie de code OTP

Envoyer des codes OTP aux utilisateurs

Informations sur l'automatisation

GUID:	`1758796201`
Template:	toke_opt_deliver
Catégorie:	Authentification
Variables:	1 variable(s)
Créé le:	24/09/2025 19:24

Ce document vous guide dans l'intégration de votre automatisation WhatsApp

Généré le 29/09/2025 à 12:40

Configuration Webhook

Pour déclencher votre automatisation, utilisez l'endpoint API ci-dessous. Cette API nécessite une authentification HMAC pour sécuriser vos requêtes.

Endpoint de l'API

```
https://p.wap.cm/automation/
```

Informations sur l'endpoint

Méthode:	**POST**
Content-Type:	`application/json`
Authentification:	**HMAC (obligatoire)**
Limite de taux:	100 requests per minute

Authentification HMAC

Cette API utilise l'authentification HMAC pour sécuriser les requêtes. Vous devez inclure les en-têtes d'authentification suivants:

```
API Key: {{code_fourni_dans_le_contrat}}
API Secret: {{cle_secrete_fournie_dans_le_contrat}}

NOTE: Ces informations sont spécifiques à votre contrat.
Gardez votre API Secret confidentiel et ne l'exposez jamais côté client.
```

Structure du payload

Votre requête POST doit contenir les champs suivants:

```
{
    "reference": "string (GUID de l'automation - obligatoire)",
    "country": "string (Code pays ISO 2 lettres - obligatoire, ex: \"CM\", \"FR\")",
    "recipient": "string (Numéro WhatsApp au format local sans indicatif - obligatoire)",
    "scheduled": "string (Planification de l'envoi dans un timestamp ISO 8601 avec millisecondes -
obligatoire)",
    "variables": "object (vos variables si le template en utilise)"
}
```

Exemples de format "recipient":

- `Cameroun: "690000000" (pour +237690000000)`
- `France: "612345678" (pour +33612345678)`

En-têtes d'authentification HMAC requis:

```
X-Api-Key: {{votre_api_key}}
X-Api-Timestamp: {{timestamp_unix}}
X-Api-Signature: {{hmac_sha256_signature}}
```

La signature HMAC-SHA256 est calculée sur: api_key + timestamp

Structure de la réponse

Réponse en cas de succès (HTTP 200):


```
{
    "success": true,
    "status": 200,
    "message": "message_queued_successfully",
    "data": {
        "reference": "WAP250929112004.X84803",
        "meta_id": null,
        "recipient": "2376XXXXXXXX",
        "country": "CM - Cameroun ??",
        "status": "scheduled",
        "cost": 10,
        "currency": "XAF",
        "scheduled": "2025-09-29T11:20:03.324Z",
        "last_update": "2025-09-29T11:20:04.365Z"
    }
}
```


Vérification du statut

Pour vérifier les changements de statut d'un message:

```
GET https://p.wap.cm/automation/{{automation-reference}}

Exemple:
GET https://p.wap.cm/automation/WAP250929112004.C84803

En-têtes requis:
X-Api-Key: {{votre_api_key}}
X-Api-Timestamp: {{timestamp_unix}}
X-Api-Signature: {{hmac_signature}}
```

Variables et exemples

Votre template utilise des variables dynamiques. Voici le mapping et des exemples d'utilisation:

Placeholder	Nom de variable	Exemple de valeur
{{1}}	**code_otp**	123456

Exemple de payload complet

```
{
    "reference": 1758796201,
    "country": "CM",
    "recipient": "690000000",
    "scheduled": "2025-09-29T12:40:14.590Z",
    "variables": {
        "code_otp": "123456"
    }
}
```

Exemples de code

Exemple cURL avec authentification HMAC

```
# Génération du timestamp
TIMESTAMP=1759149614
API_KEY="{{votre_api_key}}"
API_SECRET="{{votre_api_secret_base64}}"

# Génération de la signature HMAC-SHA256
MESSAGE="$API_KEY$TIMESTAMP"
SIGNATURE=$(echo -n "$MESSAGE" | openssl dgst -sha256 -hmac $(echo $API_SECRET | base64 -d) -binary | base64)

curl -X POST 'https://p.wap.cm/automation/' \
  -H 'Content-Type: application/json' \
  -H 'X-Api-Key: $API_KEY' \
  -H 'X-Api-Timestamp: $TIMESTAMP' \
  -H 'X-Api-Signature: $SIGNATURE' \
  -d
'{"reference":1758796201,"country":"CM","recipient":"690000000","scheduled":"2025-09-29T12:40:14.591Z","variables":{"code_otp":"123456"}}'
```

Exemple PHP avec authentification HMAC

```
<?php

// Configuration API
$apiKey = '{{votre_api_key}}';
$apiSecret = '{{votre_api_secret_base64}}';
$endpoint = 'https://p.wap.cm/automation/';

// Génération timestamp
$timestamp = time();

// Message à signer (apiKey + timestamp)
$message = $apiKey . $timestamp;

// Décodage du secret base64
$secretBytes = base64_decode($apiSecret);

// Génération signature HMAC-SHA256
$signature = base64_encode(hash_hmac('sha256', $message, $secretBytes, true));

// Payload
$payload = array (
  'reference' => 1758796201,
  'country' => 'CM',
  'recipient' => '690000000',
  'scheduled' => '2025-09-29T12:40:14.591Z',
  'variables' =>
  array (
    'code_otp' => '123456',
  ),
);

$jsonPayload = json_encode($payload);

// En-têtes avec authentification
$headers = [
    'Content-Type: application/json',
    'X-Api-Key: ' . $apiKey,
    'X-Api-Timestamp: ' . $timestamp,
    'X-Api-Signature: ' . $signature
];

// Envoi de la requête
$ch = curl_init($endpoint);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $jsonPayload);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

$response = curl_exec($ch);
$httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

echo "Response: " . $response;
```

Tests et bonnes pratiques

Environnement de test

Utilisez le même endpoint avec le paramètre test=1 pour tester sans envoyer de vrais messages:

```
https://p.wap.cm/automation/?test=1
```

Codes de réponse

Code	Description
200	Success - Message queued for delivery
400	Bad Request - Invalid payload or missing required fields
404	Not Found - Automation not found or inactive
429	Too Many Requests - Rate limit exceeded
500	Internal Server Error - System error occurred

Bonnes pratiques

- Testez toujours avec des numéros de téléphone valides
- Vérifiez que toutes les variables sont renseignées
- Implémentez une gestion d'erreur robuste
- Surveillez les temps de réponse de l'API
- Respectez les limites de taux pour éviter les blocages
- Loggez les réponses pour le debugging

Support technique

Pour toute question sur l'intégration ou problème technique:

Email: `tech@wap.cm`

Notre équipe technique vous répondra dans les 24 heures ouvrables.