

Guide d'Intégration API WaMe Plus

Pour Développeurs

1. Introduction

L'API WaMe Plus vous permet d'intégrer l'envoi de messages WhatsApp dans vos applications. Cette documentation détaille les endpoints disponibles, l'authentification requise et les structures de données à utiliser.

2. Base URL

```
```  
https://api.wame.cm
```
```

3. Authentification

L'API utilise deux types d'authentification :

3.1. Basic Auth (Routes Admin)

```
```http  
Authorization: Basic base64(username:password)
```
```

3.2. Bearer Token (Routes Client)

```
```http  
Authorization: Bearer <votre_token_jwt>
```
```

Pour obtenir un token :

```
```http  
POST /auth/wa-login
{
 "code": "votre_code_whatsapp",
 "pin": "votre_pin"
}
```
```

Réponse :

```
```json  
{
 "success": true,
 "data": {
 "token": "jwt_token_here",
 "whatsapp": {
 "code": "12345678",
 "name": "Mon Compte",
 "mobile": "237600000000",
 "pricing": 25.00,
 "balance": 1000.00,
 "ceiling": 900
 }
 }
}
```
```

```
}  
...
```

4. Format des Réponses

Toutes les réponses suivent ce format :

```
#### Succès  
```json  
{
 "success": true,
 "data": {...} // ou
 "counter": 10, // pour les tableaux
 "data": [...]
}
...
```

```
Erreur
```json  
{  
  "success": false,  
  "error": "error_code_here"  
}  
...
```

5. Routes Principales

5.1. Envoi de Message Simple

```
```http  
POST /message/send
Authorization: Bearer <token>
Content-Type: application/json

{
 "country": "CM",
 "recipient": "237600000000",
 "content": "Votre message",
 "scheduled": "2025-01-15 14:30:00" // Optionnel
}
...
```

Réponse :

```
```json  
{  
  "success": true,  
  "data": {  
    "token": "WM202501150001",  
    "scheduled": "2025-01-15 14:30:00",  
    "recipient": "237600000000",  
    "content": "Votre message",  
    "state": "pending",  
    "cost": 25.00  
  }  
}  
}
```

```

#### #### 5.2. Envoi via Template

```http

POST /message/service/{token}

Authorization: Bearer <token>

Content-Type: application/json

```
{
  "country": "CM",
  "recipient": "237600000000",
  "variables": {
    "1": "Variable 1",
    "2": "Variable 2"
  },
  "scheduled": "2025-01-15 14:30:00" // Optionnel
}
```

```

#### #### 5.3. État d'un Message

```http

GET /message/{token}

Authorization: Bearer <token>

```

Réponse :

```json

```
{
  "success": true,
  "data": {
    "token": "WM202501150001",
    "recipient": "237600000000",
    "content": "Votre message",
    "variables": null,
    "scheduled": "2025-01-15 14:30:00",
    "sent": "2025-01-15 14:30:05",
    "delivered": "2025-01-15 14:30:10",
    "viewed": null,
    "state": "delivered",
    "cost": 25.00,
    "reference": "wamid.xxx",
    "service": null,
    "template": null,
    "whatsapp": {
      "code": "12345678",
      "name": "Mon Compte",
      "mobile": "237600000000",
      "pricing": 25.00,
      "balance": 1000.00,
      "ceiling": 900
    }
  }
}
```

}

```

### ### 6. États des Messages

Les messages peuvent avoir les états suivants :

- `pending` : En attente d'envoi
- `scheduled` : Programmé
- `sent` : Envoyé
- `delivered` : Livré
- `read` : Lu
- `failed` : Échec

### ### 7. Gestion des Campagnes

#### #### 7.1. Créer une Campagne

```http

POST /message/campaign/{token}

Authorization: Bearer <token>

```
{
  "messages": [
    {
      "country": "CM",
      "recipient": "237600000000",
      "variables": {
        "1": "John",
        "2": "25000"
      }
    },
    // Plus de messages...
  ]
}
```

7.2. État d'une Campagne

```http

GET /campaign/{token}/stats

Authorization: Bearer <token>

```

8. Gestion des Erreurs

Codes d'erreur communs :

- `invalid_credentials` : Identifiants incorrects
- `insufficient_balance` : Solde insuffisant
- `invalid_recipient` : Numéro invalide
- `template_not_found` : Template inexistant
- `campaign_not_found` : Campagne inexistante

9. Bonnes Pratiques

1. **Gestion du Token**

- Stockez le token de manière sécurisée
- Renouvelez-le avant expiration (24h)
- Gérez les erreurs 401/403

2. ****Envoi de Messages****
 - Validez les numéros avant envoi
 - Gérez les erreurs d'envoi
 - Suivez l'état des messages
3. ****Rate Limiting****
 - Maximum 100 requêtes/minute
 - Implémentez une file d'attente si nécessaire
4. ****Campagnes****
 - Maximum 10 messages par batch
 - Surveillez les stats de livraison
 - Gérez les retours d'erreur

10. Exemples d'Intégration

JavaScript/Node.js avec Axios

```
```javascript
const axios = require('axios');

class WaMeAPI {
 constructor(token) {
 this.api = axios.create({
 baseURL: 'https://api.wame.cm',
 headers: {
 'Authorization': `Bearer ${token}`,
 'Content-Type': 'application/json'
 }
 });
 }

 async sendMessage(recipient, content, country = 'CM') {
 try {
 const response = await this.api.post('/message/send', {
 recipient,
 content,
 country
 });
 return response.data;
 } catch (error) {
 this.handleError(error);
 }
 }

 handleError(error) {
 if (error.response) {
 throw new Error(error.response.data.error);
 }
 throw error;
 }
}
```
```

PHP avec Guzzle

```
```php
use GuzzleHttp\Client;

class WaMeAPI {
 private $client;

 public function __construct($token) {
 $this->client = new Client([
 'base_uri' => 'https://api.wame.cm',
 'headers' => [
 'Authorization' => "Bearer {$token}",
 'Content-Type' => 'application/json'
]
]);
 }

 public function sendMessage($recipient, $content, $country =
'CM') {
 try {
 $response = $this->client->post('/message/send', [
 'json' => [
 'recipient' => $recipient,
 'content' => $content,
 'country' => $country
]
]);
 return json_decode($response->getBody(), true);
 } catch (RequestException $e) {
 $this->handleError($e);
 }
 }
}
```
```

11. Support et Aide

Pour toute assistance :

- Email : support@wame.cm
- Documentation complète : <https://docs.wame.cm>
- Support technique : cyrille@imediatis.net