🔌 Referencia de APIs - SMS CloudMX

📋 Tabla de Contenidos

🔐 Autenticación

Todas las APIs requieren autenticación mediante NextAuth.js session cookies.

Headers Requeridos

```
Content-Type: application/json
Cookie: next-auth.session-token=...
```

Verificar Sesión

```
import { getServerSession } from 'next-auth'
import { authOptions } from '@/app/api/auth/[...nextauth]/route'

const session = await getServerSession(authOptions)
if (!session) {
  return NextResponse.json({ error: 'No autorizado' }, { status: 401 })
}
```

📋 Listas de Contactos

GET `/api/dashboard/lists`

Obtener todas las listas del usuario autenticado

Query Parameters:

```
interface ListsQuery {
  page?: string        // Página (default: 1)
  limit?: string       // Límite por página (default: 10)
  search?: string      // Búsqueda por nombre
  type?: 'STATIC' | 'DYNAMIC'  // Filtrar por tipo
}
```

Response:

```
interface ListsResponse {
  success: boolean
  data: {
    lists: ContactList[]
    total: number
    page: number
    limit: number
  }
}

interface ContactList {
  id: string
  name: string
  description: string | null
  type: 'STATIC' | 'DYNAMIC'
  segmentationCriteria: any | null
  _count: {
    memberships: number
  }
  createdAt: string
  updatedAt: string
}
```

Ejemplo:

```
curl -X GET "http://localhost:3000/api/dashboard/lists?page=1&limit=5" \
  -H "Cookie: next-auth.session-token=..."
```

POST `/api/dashboard/lists`

Crear una nueva lista de contactos

Request Body:

```
interface CreateListRequest {
  name: string                    // Requerido, max 100 caracteres
  description?: string           // Opcional, max 500 caracteres
  type: 'STATIC' | 'DYNAMIC'    // Requerido
  segmentationCriteria?: {       // Requerido si type === 'DYNAMIC'
    location?: string
    dateRange?: {
      start: string
      end: string
    }
    // Más criterios...
  }
}
```

Response:

```
interface CreateListResponse {
  success: boolean
  data: ContactList
}
```

Ejemplo:

```bash
curl -X POST "http://localhost:3000/api/dashboard/lists" \
  -H "Content-Type: application/json" \
  -H "Cookie: next-auth.session-token=..." \
  -d '{
    "name": "Clientes VIP",
    "description": "Lista de clientes premium",
    "type": "STATIC"
  }'
```

GET `/api/dashboard/lists/[id]`

Obtener detalles de una lista específica

Path Parameters:
- `id`: ID de la lista (string)

Response:

```typescript
interface ListDetailResponse {
  success: boolean
  data: ContactList & {
    memberships: Array<{
      contact: Contact
      subscriptionStatus: 'SUBSCRIBED' | 'UNSUBSCRIBED'
      subscribedAt: string | null
      unsubscribedAt: string | null
    }>
    _count: {
      memberships: number
    }
  }
}
```

PUT `/api/dashboard/lists/[id]`

Actualizar una lista existente

Request Body: Igual que `POST /api/dashboard/lists`

Response:

```typescript
interface UpdateListResponse {
  success: boolean
  data: ContactList
}
```

DELETE `/api/dashboard/lists/[id]`

Eliminar una lista

Response:

```
interface DeleteListResponse {
  success: boolean
  message: string
}
```

POST `/api/dashboard/lists/[id]/contacts`

Gestionar contactos de una lista

Request Body:

```
interface ManageContactsRequest {
  action: 'ADD' | 'REMOVE' | 'BULK_ADD'
  contactIds?: string[]          // Para ADD/REMOVE
  contacts?: Array<{            // Para BULK_ADD
    phone: string               // Formato: +1234567890
    firstName?: string
    lastName?: string
    email?: string
    metadata?: Record<string, any>
  }>
}
```

Response:

```
interface ManageContactsResponse {
  success: boolean
  data: {
    added: number
    removed: number
    existing: number
    invalid: Array<{
      contact: any
      reason: string
    }>
  }
}
```

Ejemplo - Agregar contactos masivamente:

```
curl -X POST "http://localhost:3000/api/dashboard/lists/clist123/contacts" \
  -H "Content-Type: application/json" \
  -H "Cookie: next-auth.session-token=..." \
  -d '{
    "action": "BULK_ADD",
    "contacts": [
      {
        "phone": "+1234567890",
        "firstName": "Juan",
        "lastName": "Pérez",
        "email": "juan@ejemplo.com"
      }
    ]
  }'
```

POST `/api/dashboard/lists/[id]/subscriptions`

Gestionar suscripciones de contactos

Request Body:

```typescript
interface ManageSubscriptionsRequest {
  action: 'SUBSCRIBE' | 'UNSUBSCRIBE'
  contactIds: string[]
}
```

POST `/api/dashboard/lists/segment`

Aplicar segmentación para crear/actualizar listas dinámicas

Request Body:

```typescript
interface SegmentationRequest {
  listId?: string               // Para actualizar lista existente
  criteria: {
    location?: string
    dateRange?: {
      start: string            // ISO date string
      end: string
    }
    metadata?: Record<string, any>
    phone?: {
      countryCode?: string
      areaCode?: string
    }
  }
  preview?: boolean            // Solo mostrar resultados sin crear
}
```

Response:

```typescript
interface SegmentationResponse {
  success: boolean
  data: {
    contacts: Contact[]
    total: number
    criteria: any
  }
}
```

GET `/api/dashboard/lists/stats`

Obtener estadísticas globales de listas

Response:

```
interface ListsStatsResponse {
  success: boolean
  data: {
    totalLists: number
    totalContacts: number
    totalSubscribed: number
    totalUnsubscribed: number
    listsGrowth: number        // Porcentaje de crecimiento
    contactsGrowth: number
    topLists: Array<{
      id: string
      name: string
      contactCount: number
    }>
  }
}
```

📤 Campañas SMS

GET `/api/dashboard/campaigns`

Obtener todas las campañas del usuario

Query Parameters:

```
interface CampaignsQuery {
  page?: string
  limit?: string
  status?: 'DRAFT' | 'SCHEDULED' | 'SENDING' | 'SENT' | 'FAILED'
  search?: string
}
```

Response:

```
interface CampaignsResponse {
  success: boolean
  data: {
    campaigns: Campaign[]
    total: number
    page: number
    limit: number
  }
}

interface Campaign {
  id: string
  name: string
  message: string
  status: 'DRAFT' | 'SCHEDULED' | 'SENDING' | 'SENT' | 'FAILED'
  scheduledAt: string | null
  sentAt: string | null
  totalRecipients: number
  successfulSends: number
  failedSends: number
  creditsUsed: number
  lists: Array<{
    id: string
    name: string
  }>
  createdAt: string
  updatedAt: string
}
```

POST `/api/dashboard/campaigns`

Crear nueva campaña SMS

Request Body:

```
interface CreateCampaignRequest {
  name: string                  // Requerido, max 100 caracteres
  message: string              // Requerido, max 1600 caracteres (SMS limit)
  listIds: string[]           // Array de IDs de listas
  scheduledAt?: string        // ISO date string para programar
  sendImmediately?: boolean   // true para enviar inmediatamente
}
```

Response:

```
interface CreateCampaignResponse {
  success: boolean
  data: Campaign
}
```

Ejemplo:

```
curl -X POST "http://localhost:3000/api/dashboard/campaigns" \
  -H "Content-Type: application/json" \
  -H "Cookie: next-auth.session-token=..." \
  -d '{
    "name": "Promoción Verano",
    "message": "¡50% de descuento en todos nuestros productos! Válido hasta el 31/08.
Usa código VERANO50",
    "listIds": ["clist123", "clist456"],
    "sendImmediately": false,
    "scheduledAt": "2024-07-15T10:00:00Z"
  }'
```

GET `/api/dashboard/campaigns/[id]`

Obtener detalles de una campaña específica

Response:

```
interface CampaignDetailResponse {
  success: boolean
  data: Campaign & {
    messages: Array<{
      id: string
      phone: string
      status: 'PENDING' | 'SENT' | 'DELIVERED' | 'FAILED'
      sentAt: string | null
      deliveredAt: string | null
      errorMessage: string | null
      contact: {
        firstName: string
        lastName: string
      }
    }>
    stats: {
      totalSent: number
      totalDelivered: number
      totalFailed: number
      deliveryRate: number
      avgDeliveryTime: number
    }
  }
}
```

PUT `/api/dashboard/campaigns/[id]`

Actualizar una campaña (solo si status === ‘DRAFT’)

Request Body: Igual que `POST /api/dashboard/campaigns`

DELETE `/api/dashboard/campaigns/[id]`

Eliminar una campaña (solo si status === ‘DRAFT’)

Response:

```
interface DeleteCampaignResponse {
  success: boolean
  message: string
}
```

POST `/api/dashboard/campaigns/[id]/send`

Enviar una campaña inmediatamente

Request Body:

```
interface SendCampaignRequest {
  confirm: boolean             // Debe ser true para confirmar envío
}
```

Response:

```
interface SendCampaignResponse {
  success: boolean
  data: {
    campaignId: string
    totalRecipients: number
    estimatedCredits: number
    startedAt: string
  }
}
```

GET `/api/dashboard/campaigns/[id]/stats`

Obtener estadísticas detalladas de una campaña

Response:

```
interface CampaignStatsResponse {
  success: boolean
  data: {
    overview: {
      totalSent: number
      totalDelivered: number
      totalFailed: number
      deliveryRate: number
      creditsUsed: number
    }
    timeline: Array<{
      hour: string
      sent: number
      delivered: number
      failed: number
    }>
    errors: Array<{
      errorCode: string
      errorMessage: string
      count: number
    }>
    topPerformingLists: Array<{
      listId: string
      listName: string
      deliveryRate: number
      totalContacts: number
    }>
  }
}
```

📊 Dashboard

GET `/api/dashboard/stats`

Obtener estadísticas generales del dashboard

Response:

```
interface DashboardStatsResponse {
  success: boolean
  data: {
    overview: {
      totalLists: number
      totalContacts: number
      totalCampaigns: number
      creditsRemaining: number
    }
    recentActivity: Array<{
      id: string
      type: 'CAMPAIGN_SENT' | 'LIST_CREATED' | 'CONTACTS_IMPORTED'
      title: string
      description: string
      timestamp: string
    }>
    campaignStats: {
      thisMonth: {
        sent: number
        delivered: number
        deliveryRate: number
      }
      lastMonth: {
        sent: number
        delivered: number
        deliveryRate: number
      }
      growth: {
        sent: number           // Porcentaje de crecimiento
        deliveryRate: number
      }
    }
    topLists: Array<{
      id: string
      name: string
      contactCount: number
      recentActivity: number
    }>
  }
}
```

Ejemplo:

```
curl -X GET "http://localhost:3000/api/dashboard/stats" \
  -H "Cookie: next-auth.session-token=..."
```

📋 Modelos de Datos

Contact

```
interface Contact {
  id: string
  phone: string               // Formato: +1234567890
  firstName: string | null
  lastName: string | null
  email: string | null
  metadata: any | null        // JSON con campos personalizados
  createdAt: Date
  updatedAt: Date
  userId: string
}
```

ContactList

```
interface ContactList {
  id: string
  name: string
  description: string | null
  type: 'STATIC' | 'DYNAMIC'
  segmentationCriteria: any | null
  createdAt: Date
  updatedAt: Date
  userId: string
}
```

ListMembership

```
interface ListMembership {
  id: string
  contactId: string
  listId: string
  subscriptionStatus: 'SUBSCRIBED' | 'UNSUBSCRIBED'
  subscribedAt: Date | null
  unsubscribedAt: Date | null
  addedAt: Date
}
```

Campaign

```typescript
interface Campaign {
  id: string
  name: string
  message: string
  status: 'DRAFT' | 'SCHEDULED' | 'SENDING' | 'SENT' | 'FAILED'
  scheduledAt: Date | null
  sentAt: Date | null
  totalRecipients: number
  successfulSends: number
  failedSends: number
  creditsUsed: number
  createdAt: Date
  updatedAt: Date
  userId: string
}
```

Message

```typescript
interface Message {
  id: string
  campaignId: string
  contactId: string
  phone: string
  message: string
  status: 'PENDING' | 'SENT' | 'DELIVERED' | 'FAILED'
  sentAt: Date | null
  deliveredAt: Date | null
  errorMessage: string | null
  creditsUsed: number
  createdAt: Date
}
```

❌ Códigos de Error

Errores de Autenticación (401)

```json
{
  "success": false,
  "error": "No autorizado",
  "code": "UNAUTHORIZED"
}
```

Errores de Validación (400)

```json
{
  "success": false,
  "error": "Datos inválidos",
  "code": "VALIDATION_ERROR",
  "details": {
    "name": ["El nombre es requerido"],
    "phone": ["Formato de teléfono inválido"]
  }
}
```

Errores de Recursos No Encontrados (404)

```json
{
  "success": false,
  "error": "Lista no encontrada",
  "code": "NOT_FOUND"
}
```

Errores de Límites (429)

```json
{
  "success": false,
  "error": "Límite de rate excedido",
  "code": "RATE_LIMIT_EXCEEDED",
  "retryAfter": 60
}
```

Errores de Servidor (500)

```json
{
  "success": false,
  "error": "Error interno del servidor",
  "code": "INTERNAL_ERROR",
  "requestId": "req_123456789"
}
```

Errores de Créditos (402)

```json
{
  "success": false,
  "error": "Créditos insuficientes",
  "code": "INSUFFICIENT_CREDITS",
  "required": 100,
  "available": 50
}
```

🔧 Utilidades para Testing

Postman Collection

Importa esta colección en Postman para probar todas las APIs:

```json
{
  "info": {
    "name": "SMS CloudMX APIs",
    "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"
  },
  "variable": [
    {
      "key": "baseUrl",
      "value": "http://localhost:3000"
    }
  ],
  "auth": {
    "type": "bearer",
    "bearer": [
      {
        "key": "token",
        "value": "{{sessionToken}}",
        "type": "string"
      }
    ]
  }
}
```

cURL Examples Script

```bash
#!/bin/bash
# Script para probar APIs rápidamente

BASE_URL="http://localhost:3000"
SESSION_COOKIE="next-auth.session-token=your-session-token"

# Obtener listas
curl -X GET "$BASE_URL/api/dashboard/lists" \
  -H "Cookie: $SESSION_COOKIE"

# Crear nueva lista
curl -X POST "$BASE_URL/api/dashboard/lists" \
  -H "Content-Type: application/json" \
  -H "Cookie: $SESSION_COOKIE" \
  -d '{"name":"Test List","type":"STATIC"}'

# Obtener campañas
curl -X GET "$BASE_URL/api/dashboard/campaigns" \
  -H "Cookie: $SESSION_COOKIE"
```

Documentación generada automáticamente
Versión: v1.0.0
Última actualización: $(date)