# Documentación de Endpoints de la API

IS-K Pop Backend

30 de noviembre de 2025

## Información General

**Base URL:** `http://localhost:{PORT}/api`

## 1. Autenticación (`/auth`)

| Método | URL | Autenticación | Rol Requerido | Descripción |
|---|---|---|---|---|
| POST | `/api/auth/login` | No | - | Iniciar sesión |

### POST /api/auth/login - Estructura

**Request Body:**

```
{
  "email": "usuario@gmail.com",
  "password": "password123"
}
```

**Response Exitosa (200):**

```
{
  "success": true,
  "data": {
    "user": {
      "id": 1,
      "email": "usuario@gmail.com",
```

```
      "name": "Nombre Usuario",
      "role": "admin",
      "profileData": {
        "agenciaId": 1  // Para manager/director
        // o "aprendizId": 1  // Para apprentice
        // o "artistaIdAp": 1, "artistaIdGr": 1  // Para artist
      }
    },
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "expiresIn": 3600
  }
}
```

**Response Error (401):**

```
{
  "success": false,
  "error": "Invalid credentials"
}
```

## 2. Usuarios (/user)

| Método | URL | Autenticación | Rol Requerido | Descripción |
|---|---|---|---|---|
| POST | `/api/user` | Sí | Admin | Crear usuario |
| GET | `/api/user` | Sí | - | Listar usuarios |
| GET | `/api/user/:id` | Sí | - | Obtener usuario por ID |
| PUT | `/api/user/:id` | Sí | Admin | Actualizar usuario |
| DELETE | `/api/user/:id` | Sí | Admin | Eliminar usuario |

### POST /api/user/ - Crear Usuario

**Headers:** `Authorization: Bearer {token}`
**Request Body - Admin:**

```
{
  "email": "admin@test.com",
  "name": "Admin Usuario",
```

```
  "password": "password123",
  "role": "admin"
}
```

**Request Body - Manager/Director:**

```
{
  "email": "manager@test.com",
  "name": "Manager Usuario",
  "password": "password123",
  "role": "manager",
  "agencyId": 1
}
```

**Request Body - Apprentice:**

```
{
  "email": "apprentice@test.com",
  "name": "Aprendiz Usuario",
  "password": "password123",
  "role": "apprentice",
  "IdAp": 1
}
```

**Request Body - Artist:**

```
{
  "email": "artist@test.com",
  "name": "Artista Usuario",
  "password": "password123",
  "role": "artist",
  "IdAp": 1,
  "IdGr": 1
}
```

**Response Exitosa (201):**

```
{
  "success": true,
  "data": {
    "id": 5,
```

```
    "email": "usuario@test.com",
    "name": "Usuario",
    "role": "manager",
    "profileData": { "agenciaId": 1 }
  }
}
```

**Response Error (400):**

```
{
  "success": false,
  "error": "Missing required fields"
}
```

### GET /api/user/ - Listar Usuarios

**Response Exitosa (200):**

```
{
  "success": true,
  "data": [
    {
      "id": 1,
      "email": "admin@gmail.com",
      "name": "Admin Usuario",
      "role": "admin",
      "profileData": {}
    }
  ]
}
```

### GET /api/user/:id - Obtener Usuario

**Response Exitosa (200):**

```
{
  "success": true,
  "data": {
    "id": 1,
    "email": "usuario@test.com",
```

```
    "name": "Usuario",
    "role": "manager",
    "profileData": { "agenciaId": 1 }
  }
}
```

### PUT /api/user/:id - Actualizar Usuario

**Request Body:** Campos opcionales a actualizar
**Response Exitosa (200):**

```
{
  "success": true,
  "data": { /* Usuario actualizado */ }
}
```

### DELETE /api/user/:id - Eliminar Usuario

**Response Exitosa (200):**

```
{
  "success": true,
  "message": "User deleted successfully"
}
```

## 3. Agencias (/agency)

| Método | URL | Autenticación | Rol Requerido | Descripción |
|---|---|---|---|---|
| POST | /api/agency | Sí | Staff | Crear agencia |
| GET | /api/agency | Sí | - | Listar todas las agencias |
| GET | /api/agency/:id | Sí | - | Obtener agencia por ID |
| PUT | /api/agency/:id | Sí | Staff | Actualizar agencia |
| DELETE | /api/agency/:id | Sí | Staff | Eliminar agencia |
| GET | /api/agency/search/agency_name | Sí | - | Buscar agencias por nombre |
| GET | /api/agency/search/agency_address | Sí | - | Buscar agencias por dirección |
| GET | /api/agency/search/agency_foundation | Sí | - | Buscar por fecha fundación |

### POST /api/agency/ - Crear Agencia

**Request Body:**

```
{
  "name": "SM Entertainment",
  "address": "Seúl, Corea del Sur",
  "foundation": "1995-02-14"
}
```

**Response Exitosa (201):**

```
{
  "success": true,
  "data": {
    "id": 1,
    "nombre": "SM Entertainment",
    "ubicacion": "Seúl, Corea del Sur",
    "fechaFundacion": "1995-02-14T00:00:00.000Z"
  }
}
```

### GET /api/agency/ - Listar Agencias

**Response Exitosa (200):**

```
{
  "success": true,
  "data": [
    {
      "id": 1,
      "nombre": "SM Entertainment",
      "ubicacion": "Seúl",
      "fechaFundacion": "1995-02-14T00:00:00.000Z"
    }
  ]
}
```

### GET /api/agency/:id - Obtener Agencia

**Response Exitosa (200):** Mismo formato que GET individual

### GET /api/agency/search/agency_name?name=X

**Query Params:** `name`
**Response:** Array de agencias que coincidan

## 4. Aprendices (`/apprentice`)

| Método | URL | Autenticación | Rol Requerido | Descripción |
|---|---|---|---|---|
| POST | `/api/apprentice/:id` | Sí | Staff | Crear aprendiz |
| GET | `/api/apprentice` | Sí | - | Listar todos los aprendices |
| GET | `/api/apprentice/:id` | Sí | - | Obtener aprendiz por ID |
| GET | `/api/apprentice/:name` | Sí | - | Obtener aprendiz por nombre |
| GET | `/api/apprentice/agency/:id` | Sí | - | Listar aprendices por agencia |
| PUT | `/api/apprentice/:id` | Sí | Staff | Actualizar aprendiz |
| DELETE | `/api/apprentice/:id` | Sí | Staff | Eliminar aprendiz |

### POST /api/apprentice/:id - Crear Aprendiz

**URL Param:** `id` - ID de la agencia
**Request Body:**

```
{
  "name": "Kim Minju",
  "dateOfBirth": "2001-02-05",
  "age": 22,
  "trainingLv": 3,
  "status": "En entrenamiento"
}
```

**Validaciones:**

- `age` $\geq 15$
- `trainingLv` $\geq 0$

**Response Exitosa (201):**

```
{
  "success": true,
  "data": {
    "id": 1,
    "nombreCompleto": "Kim Minju",
    "fechaNacimiento": "2001-02-05T00:00:00.000Z",
    "edad": 22,
    "nivelEntrenamiento": 3,
    "estadoAprendiz": "En entrenamiento"
  }
}
```

### GET /api/apprentice/ - Listar Aprendices

**Response Exitosa (200):**

```
{
  "success": true,
  "data": [ /* Array de aprendices */ ]
}
```

### GET /api/apprentice?name=X - Buscar por Nombre

**Query Params:** `name`
**Response:** Aprendiz(es) que coincidan con el nombre

## 5. Artistas (`/artist`)

| Método | URL | Autenticación | Rol Requerido | Descripción |
|---|---|---|---|---|
| POST | `/api/artist` | Sí | Staff | Crear artista |
| GET | `/api/artist` | Sí | Agency Access | Listar todos los artistas |
| GET | `/api/artist/:apprenticeId&:groupId` | Sí | - | Obtener artista por IDs |
| GET | `/api/artist/:id` | Sí | - | Obtener artistas por agencia |
| PUT | `/api/artist/:apprenticeId&:groupId` | Sí | Staff | Actualizar artista |
| DELETE | `/api/artist/:apprenticeId&:groupId` | Sí | Staff | Eliminar artista |

### POST /api/artist/ - Crear Artista

**Request Body:**

```
{
  "apprenticeId": 1,
  "groupId": 1,
  "stageName": "Lisa",
  "debutDate": "2016-08-08",
  "status": "Activo"
}
```

**Response Exitosa (201):**

```
{
  "success": true,
  "data": {
    "idAp": 1,
    "idGr": 1,
    "nombreArtistico": "Lisa",
    "fsechaDebut": "2016-08-08T00:00:00.000Z",
    "estadoArtista": "Activo"
  }
}
```

### GET /api/artist/ - Listar Artistas

**Response Exitosa (200):**

```
{
  "success": true,
  "data": [ /* Array de artistas */ ]
}
```

### GET /api/artist/:apprenticeId&:groupId - Obtener Artista

**Nota:** Usa llave compuesta (apprenticeId y groupId)

## 6. Conceptos (/concept)

| Método | URL | Autenticación | Rol Requerido | Descripción |
|---|---|---|---|---|
| POST | /api/concept | No | - | Crear concepto |
| GET | /api/concept | No | - | Listar todos los conceptos |
| GET | /api/concept/:id | No | - | Obtener concepto por ID |
| PUT | /api/concept/:id | No | - | Actualizar concepto |
| DELETE | /api/concept/:id | No | - | Eliminar concepto |

### POST /api/concept/ - Crear Concepto

**Request Body:**

```
{
  "description": "Concepto futurista y tecnológico"
}
```

**Response Exitosa (201):**

```
{
  "success": true,
  "data": {
    "id": 1,
    "descripcion": "Concepto futurista y tecnológico"
  }
}
```

### GET /api/concept/ - Listar Conceptos

**Response Exitosa (200):**

```
{
  "success": true,
  "data": [ /* Array de conceptos */ ]
}
```

# 7. Notas Importantes

## Estructura de Respuestas

**Todas las respuestas exitosas:**

```
{
  "success": true,
  "data": { ... }  // o [ ... ] para arrays
}
```

**Todas las respuestas con error:**

```
{
  "success": false,
  "error": "Mensaje de error descriptivo"
}
```

## Códigos HTTP

- **200 OK:** Operación exitosa (GET, PUT, DELETE)
- **201 Created:** Recurso creado (POST)
- **400 Bad Request:** Datos inválidos
- **401 Unauthorized:** Sin autenticación o credenciales inválidas
- **403 Forbidden:** Sin permisos
- **404 Not Found:** Recurso no encontrado
- **500 Internal Server Error:** Error del servidor

## Autenticación

Los endpoints que requieren autenticación necesitan un token JWT en el header:
`Authorization: Bearer {token}`

### Roles Disponibles

- **Admin**: Acceso completo al sistema
- **Staff**: Director/Manager - puede gestionar agencias, aprendices y artistas
- **Agency Access**: Acceso relacionado con agencias específicas

### Advertencia

**Conflicto detectado:** En el archivo `index.ts` línea 23, las rutas de `/concept` están usando `artistRoutes.getRouter()` en lugar de `conceptRoutes.getRouter()`. Se recomienda corregir esto.

### Configuración

El puerto debe ser configurado en el archivo `secrets.ts`.