

Documentación de API - NEO

Esta documentación describe las rutas de API disponibles en el sistema NEO.

Autenticación

Todas las rutas de API requieren autenticación excepto las rutas públicas. La autenticación se maneja mediante NextAuth.js.

Endpoints de Autenticación

```
POST /api/auth/signin
POST /api/auth/signout
GET  /api/auth/session
```

Rutas de API Principales

Dashboard

```
GET /api/dashboard/stats
GET /api/dashboard/recent-activities
GET /api/dashboard/charts
```

CRM

```
GET    /api/crm/contacts
POST   /api/crm/contacts
PUT    /api/crm/contacts/[id]
DELETE /api/crm/contacts/[id]

GET    /api/crm/leads
POST   /api/crm/leads
PUT    /api/crm/leads/[id]
DELETE /api/crm/leads/[id]

GET    /api/crm/opportunities
POST   /api/crm/opportunities
PUT    /api/crm/opportunities/[id]
DELETE /api/crm/opportunities/[id]
```

Proyectos

```
GET    /api/projects
POST   /api/projects
PUT    /api/projects/[id]
DELETE /api/projects/[id]

GET    /api/projects/[id]/tasks
POST   /api/projects/[id]/tasks
PUT    /api/projects/[id]/tasks/[taskId]
DELETE /api/projects/[id]/tasks/[taskId]
```

Recursos Humanos

```
GET    /api/hr/employees
POST   /api/hr/employees
PUT    /api/hr/employees/[id]
DELETE /api/hr/employees/[id]

GET    /api/hr/attendance
POST   /api/hr/attendance
GET    /api/hr/payroll
POST   /api/hr/payroll
```

Finanzas

```
GET    /api/finance/transactions
POST   /api/finance/transactions
PUT    /api/finance/transactions/[id]
DELETE /api/finance/transactions/[id]

GET    /api/finance/budgets
POST   /api/finance/budgets
GET    /api/finance/reports
```

Inventario

```
GET    /api/inventory/products
POST   /api/inventory/products
PUT    /api/inventory/products/[id]
DELETE /api/inventory/products/[id]

GET    /api/inventory/stock
POST   /api/inventory/stock/adjust
```

Ejemplos de Uso

Obtener Contactos CRM

```
const response = await fetch('/api/crm/contacts', {
  method: 'GET',
  headers: {
    'Content-Type': 'application/json',
  },
});

const contacts = await response.json();
```

Crear Nuevo Proyecto

```
const response = await fetch('/api/projects', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    name: 'Nuevo Proyecto',
    description: 'Descripción del proyecto',
    startDate: '2024-01-01',
    endDate: '2024-12-31',
    status: 'active'
  }),
});

const project = await response.json();
```

Actualizar Empleado

```
const response = await fetch(`/api/hr/employees/${employeeId}`, {
  method: 'PUT',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    name: 'Nombre Actualizado',
    position: 'Nueva Posición',
    salary: 50000
  }),
});

const updatedEmployee = await response.json();
```

Códigos de Respuesta

- 200 - Éxito
- 201 - Creado exitosamente
- 400 - Solicitud incorrecta
- 401 - No autorizado
- 403 - Prohibido

- 404 - No encontrado
- 500 - Error interno del servidor

Paginación

Las rutas que devuelven listas soportan paginación:

```
GET /api/crm/contacts?page=1&limit=10&search=john
```

Parámetros:

- page : Número de página (por defecto: 1)
- limit : Elementos por página (por defecto: 10, máximo: 100)
- search : Término de búsqueda (opcional)

Filtros y Ordenamiento

```
GET /api/projects?status=active&sortBy=createdAt&sortOrder=desc
```

Parámetros comunes:

- sortBy : Campo por el cual ordenar
- sortOrder : asc o desc
- Filtros específicos por entidad

Manejo de Errores

Todas las respuestas de error siguen este formato:

```
{
  "error": true,
  "message": "Descripción del error",
  "code": "ERROR_CODE",
  "details": {}
}
```

Rate Limiting

Las APIs tienen límites de velocidad:

- 100 solicitudes por minuto por IP
- 1000 solicitudes por hora por usuario autenticado

Webhooks

El sistema soporta webhooks para eventos importantes:

```
POST /api/webhooks/register
POST /api/webhooks/unregister
```

Eventos disponibles:

- `contact.created`
- `project.updated`
- `transaction.created`
- `employee.updated`