

# API Documentation

## Overview

This document describes the API endpoints available in the Enobase application. The API is RESTful and uses JSON for requests and responses.

## Base URL

- Development: `http://localhost:3000/api`
- Production: `[Production URL]/api`

## Authentication

All API endpoints require authentication unless specified otherwise.

### Authentication Header

```
Authorization: Bearer <token>
```

## API Endpoints

### Formula Calculations

#### Calculate Formula

```
POST /api/calculate
```

#### Request Body

```
{
  "formula": "string",
  "variables": {
    "key": "value"
  }
}
```

#### Response

```
{
  "result": "number",
  "steps": [
    {
      "operation": "string",
      "value": "number"
    }
  ]
}
```

#### Status Codes

- 200: Success

- 400: Invalid formula or variables
- 401: Unauthorized
- 500: Server error

## Data Management

### Get Data Sources

```
GET /api/data-sources
```

#### Response

```
{
  "sources": [
    {
      "id": "string",
      "name": "string",
      "type": "string"
    }
  ]
}
```

### Create Data Source

```
POST /api/data-sources
```

#### Request Body

```
{
  "name": "string",
  "type": "string",
  "configuration": {
    "key": "value"
  }
}
```

## Formula Templates

### Get Templates

```
GET /api/templates
```

### Create Template

```
POST /api/templates
```

#### Request Body

```
{
  "name": "string",
  "description": "string",
  "formula": "string"
}
```

## Error Handling

### Error Response Format

```
{
  "error": {
    "code": "string",
    "message": "string",
    "details": {}
  }
}
```

### Common Error Codes

- `INVALID_FORMULA` : Formula syntax is invalid
- `MISSING_VARIABLES` : Required variables not provided
- `CALCULATION_ERROR` : Error during formula calculation
- `UNAUTHORIZED` : Authentication required
- `FORBIDDEN` : Insufficient permissions

## Rate Limiting

- Rate limit: 100 requests per minute
- Rate limit header: `X-RateLimit-Limit`
- Remaining requests: `X-RateLimit-Remaining`
- Reset time: `X-RateLimit-Reset`

## Versioning

API versioning is handled through the URL path:

```
/api/v1/resource
```

## Data Types

### Formula Object

```
interface Formula {
  id: string;
  name: string;
  expression: string;
  variables: Variable[];
  created_at: string;
  updated_at: string;
}
```

### Variable Object

```
interface Variable {
  name: string;
  type: "number" | "string" | "boolean";
  default?: any;
}
```

```
    required: boolean;
  }
```

## WebSocket API

### Connection

```
const ws = new WebSocket('ws://localhost:3000/api/ws');
```

### Events

#### Formula Update

```
{
  "type": "formula_update",
  "data": {
    "formula_id": "string",
    "result": "number"
  }
}
```

## Examples

### Calculate Simple Formula

```
// Request
fetch('/api/calculate', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    'Authorization': 'Bearer <token>'
  },
  body: JSON.stringify({
    function: 'multiply',
    num1: 2,
    num2: 5
  })
});
```

#### Expected Response:

```
{
  "result": 10
}
```

### String Operation Example

```
// Request
fetch('/api/calculate', {
  method: 'POST',
  headers: {
```

```
    'Content-Type': 'application/json',
    'Authorization': 'Bearer <token>'
  },
  body: JSON.stringify({
    function: 'upper',
    str: 'hello world'
  })
});
```

#### Expected Response:

```
{
  "result": "HELLO WORLD"
}
```

#### List Operation Example

```
// Request
fetch('/api/calculate', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    'Authorization': 'Bearer <token>'
  },
  body: JSON.stringify({
    function: 'max',
    numbers: [1, 10, 7]
  })
});
```

#### Expected Response:

```
{
  "result": 10
}
```

#### Date Operation Example

```
// Request
fetch('/api/calculate', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    'Authorization': 'Bearer <token>'
  },
  body: JSON.stringify({
    function: 'formatDate',
    date: new Date('2024-01-01'),
    format: 'yyyy-MM-dd'
  })
});
```

**Expected Response:**

```
{  
  "result": "2024-01-01"  
}
```

**Error Response Example**

```
{  
  "error": "Invalid request body"  
}
```