

API Documentation Template

Emmanuel BRUNO
Université de Toulon, LIS UMR CNRS 7020
emmanuel.bruno@univ-tln.fr

2026-10-03

Table des matières

1	API Documentation Template	1
1.1	API Overview	1
1.2	Authentication	1
1.2.a	Authentication Flow	1
1.3	API Versioning	1
1.4	Common Response Format	1
1.5	Error Handling	2
1.5.a	Error Codes	2
1.5.b	Error Response Example	2
1.6	Resource Endpoints	2
1.6.a	User Endpoints	2
1.7	Template Validation	3

1 API Documentation Template

i Template Usage

1. Replace [API_NAME] with your API name
2. Define authentication method
3. Document endpoints and data structures
4. Add specific error codes and responses

1.1 API Overview

[API_NAME] exposes RESTful APIs for [PURPOSE].

Base URL: [BASE_URL]

1.2 Authentication

[Define authentication method]

1.2.a Authentication Flow

```
sequenceDiagram
    participant Client
    participant Auth
    participant API

    Client->>Auth: Authentication Request
    Auth-->>Client: Auth Token
    Client->>API: API Request + Token
    API-->>Client: Response
```

1.3 API Versioning

The API uses URL-based versioning (/v1, /v2) for major changes and header-based versioning for minor changes.

For minor version selection, include:

```
API-Version: [VERSION]
```

1.4 Common Response Format

All API responses follow this JSON structure:

```
{
  "status": "success|error",
  "data": {
    // Response data object(s)
  }
}
```

```

    },
    "message": "Human readable message
(mainly for errors)",
    "code": 200, // HTTP status code
    "timestamp": 1652345678910
  }

```

1.5 Error Handling

1.5.a Error Codes

Status Code	Description
400	Bad Request - Invalid input parameters
401	Unauthorized - Authentication required
403	Forbidden - Insufficient permissions
404	Not Found - Resource doesn't exist
409	Conflict - Resource state conflict
422	Unprocessable Entity - Validation error
429	Too Many Requests - Rate limit exceeded
500	Internal Server Error - Server-side issue
503	Service Unavailable - Temporary outage

1.5.b Error Response Example

```

{
  "status": "error",
  "message": "Validation failed",
  "code": 422,
  "timestamp": 1652345678910,
  "errors": [
    {

```

```

      "field": "weight",
      "message": "Weight must be a
positive number"
    },
    {
      "field": "reps",
      "message": "Reps must be between
1 and 100"
    }
  ]
}

```

1.6 Resource Endpoints

1.6.a User Endpoints

1.6.a.a Get User Profile

GET /users/profile

Response:

```

{
  "status": "success",
  "data": {
    "userId": "user123",
    "displayName": "John Doe",
    "email": "john@example.com",
    "profileImageUrl": "https://cdn.fitnesstracker.com/profiles/user123.jpg",
    "dateJoined": 1641034800000,
    "fitnessLevel": "INTERMEDIATE",
    "heightCm": 180,
    "weightKg": 75,
    "dateOfBirth": 599634000000,
    "gender": "MALE",
    "isPremium": true,
    "settings": {
      "useMetricSystem": true,
      "enableNotifications": true,
      "darkModePreference": "SYSTEM",
      "restTimerDuration": 60
    }
  }
}

```

```
},  
"code": 200,  
"timestamp": 1652345678910  
}
```

1.6.a.b Update User Profile

PUT /users/profile

Request Body:

```
{  
  "displayName": "John Doe",  
  "heightCm": 180,  
  "weightKg": 75,  
  "fitnessLevel": "INTERMEDIATE",  
  "gender": "MALE"  
}
```

Response:

```
{  
  "status": "success",  
  "data": {  
    "userId": "user123",  
    "displayName": "John Doe",  
    // ...updated user data  
  },  
  "message": "Profile updated  
successfully",  
  "code": 200,  
  "timestamp": 1652345678910  
}
```

1.7 Template Validation

- ☐ API overview completed
- ☐ Authentication documented
- ☐ Endpoints described
- ☐ Examples provided
- ☐ Error handling defined
- ☐ Team review completed