# **Chat Application API Specification**

## **Overview**

This document provides a comprehensive API specification for the Chat Application built with Next.js, Azure MSAL authentication, and AI360 integration. The application allows authenticated users to interact with an AI assistant through a chat interface.

## **General Information**

**Base URL:** [Your Application Domain]

**API Version:** v1

**Protocol:** HTTPS

**Content-Type:** application/json

**Authentication:** Azure AD OAuth 2.0 Bearer Token

### **Technology Stack**

* **Framework:** Next.js 15.1.0
* **Authentication:** Azure MSAL (Microsoft Authentication Library)
* **Frontend:** React 18, TypeScript, Tailwind CSS
* **External AI API:** AI360 Lab45 API (api.lab45.ai)

## **Authentication**

### **Azure AD Configuration**

The application uses Azure Active Directory for authentication with the following configuration:

// MSAL Configuration  
{  
 auth: {  
 clientId: "a919164d-8b7c-43fb-8119-f1997d45ca4f",  
 authority: "<https://login.microsoftonline.com/258ac4e4-146a-411e-9dc8-79a9e12fd6da>",  
 redirectUri: "[Application Origin]"  
 },  
 cache: {  
 cacheLocation: "sessionStorage",  
 storeAuthStateInCookie: false  
 }  
}

### **Token Requirements**

All API requests must include a valid Azure AD access token in the Authorization header:

Authorization: Bearer <access\_token>

**Scope Required:** a919164d-8b7c-43fb-8119-f1997d45ca4f/.default

## **API Endpoints**

### **POST /api/completions**

**Purpose:** Generate AI-powered chat completions using the AI360 service.

**Authentication:** Required (Bearer Token)

#### **Request**

**Method:** POST

**URL:** /api/completions

**Headers:**

Authorization: Bearer <access\_token>  
Content-Type: application/json

**Body:**

{  
 "prompt": "string"  
}

**Parameters:**

* prompt (string, required): The user's message/question to send to the AI assistant

#### **Response**

**Success Response (200 OK):**

{  
 "data": {  
 "content": "string"  
 }  
}

**Response Fields:**

* data.content (string): The AI-generated response text

#### **Example**

**Request:**

curl -X POST <https://your-domain.com/api/completions> \  
 -H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIs..." \  
 -H "Content-Type: application/json" \  
 -d '{  
 "prompt": "What is the capital of France?"  
 }'

**Response:**

{  
 "data": {  
 "content": "The capital of France is Paris. It is located in the north-central part of France and serves as the country's political, economic, and cultural center."  
 }  
}

## **Request/Response Formats**

### **Request Format**

All requests must be sent with Content-Type: application/json header and include a valid JSON body where applicable.

### **Response Format**

All responses are returned in JSON format with the following structure:

**Success Response:**

{  
 "data": {  
 // Response data object  
 }  
}

**Error Response:**

{  
 "error": "string"  
}

## **Error Handling**

### **HTTP Status Codes**

|  |  |  |
| --- | --- | --- |
| **Status Code** | **Description** | **Response Format** |
| 200 | Success | {"data": {...}} |
| 401 | Unauthorized - Missing or invalid token | {"error": "No authorization token provided"} |
| 403 | Forbidden - Token expired or insufficient permissions | {"error": "Access denied"} |
| 500 | Internal Server Error | {"error": "Internal server error"} |
| 502 | Bad Gateway - External API error | {"error": "API request failed"} |

### **Error Response Examples**

**Missing Authorization Token (401):**

{  
 "error": "No authorization token provided"  
}

**External API Failure (502):**

{  
 "error": "API request failed"  
}

**Internal Server Error (500):**

{  
 "error": "Internal server error"  
}

## **External Dependencies**

### **AI360 Lab45 API**

**Base URL:** <https://api.lab45.ai/v1.1>

**Endpoint:** /skills/completion/query

**Request Format to AI360:**

{  
 "messages": [  
 {  
 "role": "user",  
 "content": "string"  
 }  
 ],  
 "skill\_parameters": {  
 "model\_name": "gpt-35-turbo-16k",  
 "max\_output\_tokens": 1000,  
 "temperature": 0.7  
 }  
}

**Response Format from AI360:**

* Streaming JSON responses
* Each chunk contains: {"data": {"content": "string"}}

## **Rate Limiting**

Currently, no specific rate limiting is implemented at the application level. Rate limiting is dependent on:

1. **Azure AD Token Limits:** Standard Azure AD token refresh limits apply
2. **AI360 API Limits:** External API provider limits (refer to AI360 documentation)

## **Security Considerations**

### **Authentication Security**

* Uses Azure AD OAuth 2.0 flow
* Tokens are stored in session storage (client-side)
* Automatic token refresh with silent authentication
* Fallback to popup authentication if silent fails

### **API Security**

* All requests require valid Azure AD bearer tokens
* HTTPS only communication
* No sensitive data stored in client-side code
* Authorization header validation on each request

### **Data Protection**

* No persistent storage of chat messages on the server
* Messages are processed in real-time and not logged
* Access tokens are scoped to specific Azure AD application

## **Frontend Integration**

### **React Hook for Token Management**

The application provides a custom React hook useAccessToken() for token management:

import { useAccessToken } from "@/app/hooks/useAccessToken";  
  
// Usage in component  
const accessToken = useAccessToken();  
  
// Use token in API calls  
const response = await fetch('/api/completions', {  
 method: 'POST',  
 headers: {  
 'Authorization': `Bearer ${accessToken}`,  
 'Content-Type': 'application/json',  
 },  
 body: JSON.stringify({ prompt: userMessage })  
});

### **Chat Interface Integration**

The ChatInterface component handles:

* Message state management
* API communication
* Error handling and display
* Loading states
* Real-time UI updates

## **Configuration**

### **Environment Variables**

The following environment variables may be required for deployment:

# Azure AD Configuration (if not hardcoded)  
AZURE\_CLIENT\_ID=a919164d-8b7c-43fb-8119-f1997d45ca4f  
AZURE\_TENANT\_ID=258ac4e4-146a-411e-9dc8-79a9e12fd6da  
  
# AI360 API Configuration  
AI360\_API\_URL=https://api.lab45.ai/v1.1

## **Changelog**

### **Version 1.0.0**

* Initial API implementation
* Azure AD authentication integration
* AI360 Lab45 API integration
* Basic chat interface support