**SvelteKit Microsoft Authentication - Complete Troubleshooting Guide**

This document contains all errors encountered during SvelteKit Microsoft Authentication setup and their solutions, compiled for future reference and troubleshooting.

**Table of Contents**

1. [Project Setup Errors](#project-setup-errors)
2. [TypeScript Configuration Errors](#typescript-configuration-errors)
3. [Dependency Management Errors](#dependency-management-errors)
4. [MSAL Authentication Errors](#msal-authentication-errors)
5. [Azure Configuration Errors](#azure-configuration-errors)
6. [SvelteKit Route Errors](#sveltekit-route-errors)
7. [Environment Variable Errors](#environment-variable-errors)

**Project Setup Errors**

**Error 1: Missing "dev" Script**

npm error Missing script: "dev"

npm error To see a list of scripts, run: npm error npm run

**Cause**: The package.json file is missing the "dev" script definition.

**Solution**: Add the dev script to package.json:

{

"scripts": {

"dev": "vite dev",

"build": "vite build",

"preview": "vite preview"

}

}

**Note**: For SvelteKit projects, the dev script should always be "dev": "vite dev" since SvelteKit uses Vite as its build tool.

**Error 2: Missing svelte.config.js**

Missing /Users/.../svelte.config.js — skipping

**Cause**: SvelteKit requires a configuration file to function properly.

**Solution**: Create svelte.config.js in the project root:

import adapter from '@sveltejs/adapter-auto';

/\*\* @type {import('@sveltejs/kit').Config} \*/

const config = {

kit: {

adapter: adapter()

}

};

export default config;

**Critical**: This file must be in the project root (same level as package.json).

**TypeScript Configuration Errors**

**Error 3: Cannot find module './$types'**

TS2307: Cannot find module './$types' or its corresponding type declarations.

**Cause**: SvelteKit hasn't generated the type definitions yet.

**Solution**: Run SvelteKit sync to generate types:

# Clean and regenerate types

rm -rf .svelte-kit

npx svelte-kit sync

**Important**: The npx svelte-kit sync command must run successfully to generate the .svelte-kit/types/ directory.

**Error 4: File not included in any tsconfig.json**

File is not included in any tsconfig.json

**Cause**: TypeScript configuration is incomplete or SvelteKit types aren't generated.

**Solution**:

1. Ensure proper tsconfig.json:

{

"extends": "./.svelte-kit/tsconfig.json",

"compilerOptions": {

"allowJs": true,

"checkJs": true,

"esModuleInterop": true,

"forceConsistentCasingInFileNames": true,

"resolveJsonModule": true,

"skipLibCheck": true,

"sourceMap": true,

"strict": true,

"moduleResolution": "bundler"

}

}

1. Create src/app.d.ts:

declare global {

namespace App {

interface Locals {

user?: {

id: string;

name: string;

email: string;

};

}

}

}

export {};

1. Run sync command:

npx svelte-kit sync

**Error 5: jsonwebtoken Import Error**

TS1192: Module 'jsonwebtoken' has no default export.

[vite] Named export 'sign' not found. The requested module 'jsonwebtoken' is a CommonJS module

**Cause**: CommonJS module imported incorrectly in ES module environment.

**Solution**: Use correct import syntax:

// Wrong

import { sign, verify } from 'jsonwebtoken';

// Correct

import pkg from 'jsonwebtoken';

const { sign, verify } = pkg;

**Alternative**: Use jose library for better ES module support:

npm uninstall jsonwebtoken @types/jsonwebtoken

npm install jose

**Error 6: Invalid exports in route files**

TS71001: Invalid export 'POST' (valid exports are prerender, ssr, csr, trailingSlash, config, actions, load, entries, or anything with a '\_' prefix)

**Cause**: Using server route exports (POST, GET) in page server files (+page.server.ts).

**Solution**: Use the correct SvelteKit file structure:

* **For API endpoints**: Use +server.ts with POST, GET exports
* **For page data**: Use +page.server.ts with load and actions exports

// src/routes/api/auth/+server.ts (correct for JSON APIs)

export const POST: RequestHandler = async ({ request }) => {

// Handle JSON

};

// src/routes/auth/+page.server.ts (correct for form actions)

export const actions: Actions = {

default: async ({ request }) => {

// Handle form data

}

};

**Dependency Management Errors**

**Error 7: ERESOLVE Dependency Conflicts**

npm error code ERESOLVE

npm error ERESOLVE could not resolve

npm error Conflicting peer dependency: @sveltejs/kit@2.21.1

**Cause**: Incompatible package versions in the SvelteKit ecosystem.

**Solutions** (in order of preference):

1. **Use legacy peer deps**:

npm install --legacy-peer-deps

1. **Use compatible versions**:

{

"devDependencies": {

"@sveltejs/adapter-auto": "^2.1.1",

"@sveltejs/kit": "^1.30.4",

"svelte": "^4.2.7",

"vite": "^4.4.2"

}

}

1. **Start fresh**:

npx sv create fresh-project

**Error 8: SvelteKit CLI Update**

'npm create svelte' has been replaced with 'npx sv create'

**Cause**: SvelteKit CLI has been updated.

**Solution**: Use the new CLI command:

# Old (deprecated)

npm create svelte@latest project-name

# New (current)

npx sv create project-name

**MSAL Authentication Errors**

**Error 9: Uninitialized MSAL Instance**

uninitialized\_public\_client\_application: You must call and await the initialize function before attempting to call any other MSAL API

**Cause**: MSAL instance used before initialization.

**Solution**: Initialize MSAL before use:

// src/lib/msal.js

import { PublicClientApplication } from '@azure/msal-browser';

import { browser } from '$app/environment';

const msalConfig = {

auth: {

clientId: import.meta.env.VITE\_AZURE\_CLIENT\_ID,

authority: `https://login.microsoftonline.com/${import.meta.env.VITE\_AZURE\_TENANT\_ID}`,

redirectUri: import.meta.env.VITE\_AZURE\_REDIRECT\_URI

},

cache: {

cacheLocation: 'sessionStorage',

storeAuthStateInCookie: false

}

};

let msalInstance = null;

let isInitialized = false;

if (browser) {

msalInstance = new PublicClientApplication(msalConfig);

}

export async function initializeMsal() {

if (browser && msalInstance && !isInitialized) {

await msalInstance.initialize();

isInitialized = true;

}

return msalInstance;

}

**Usage in components**:

<script lang="ts">

import { onMount } from 'svelte';

import { initializeMsal } from '$lib/msal';

let msalReady = false;

let localMsalInstance = null;

onMount(async () => {

localMsalInstance = await initializeMsal();

msalReady = !!localMsalInstance;

});

async function login() {

if (!msalReady) return;

// Use localMsalInstance for login

}

</script>

**Error 10: Cannot Assign to Import**

[plugin:vite-plugin-svelte] Cannot assign to import

**Cause**: Attempting to reassign an imported variable.

**Solution**: Use local variables instead of reassigning imports:

<script lang="ts">

import { msalInstance } from '$lib/msal'; // Don't reassign this

let localMsalInstance = null; // Use this instead

onMount(async () => {

localMsalInstance = await initializeMsal(); // Assign to local variable

});

</script>

**Azure Configuration Errors**

**Error 11: Application Not Found**

AADSTS700016: Application with identifier 'HBK8Q~taYtr4dCw2rl2Cpk3v8I.SA6EgvqMIvck5' was not found in the directory

**Cause**: Using client secret as client ID.

**Problem**: The error shows a client secret (HBK8Q~...) being used as client ID.

**Solution**: Use the correct values from Azure Portal:

* **Client ID**: GUID format (e.g., 12345678-1234-1234-1234-123456789012)
* **Tenant ID**: GUID format (e.g., 87654321-4321-4321-4321-210987654321)
* **Client Secret**: Random string (e.g., HBK8Q~taYtr4dCw2rl2Cpk3v8I.SA6EgvqMIvck5)

**Correct .env format**:

VITE\_AZURE\_CLIENT\_ID=12345678-1234-1234-1234-123456789012

VITE\_AZURE\_TENANT\_ID=87654321-4321-4321-4321-210987654321

VITE\_AZURE\_REDIRECT\_URI=http://localhost:5173/auth/callback

AZURE\_CLIENT\_SECRET=HBK8Q~taYtr4dCw2rl2Cpk3v8I.SA6EgvqMIvck5

**Error 12: Cross-Origin Token Redemption**

AADSTS9002326: Cross-origin token redemption is permitted only for the 'Single-Page Application' client-type

**Cause**: Azure app registration configured as "Web" application instead of "Single-Page Application".

**Solution**: Configure Azure app as SPA:

1. Go to Azure Portal → App registrations → Your app
2. Click **Authentication**
3. Remove any "Web" platform configurations
4. Add **Single-page application** platform
5. Add redirect URI: http://localhost:5173/auth/callback
6. Ensure **Implicit grant** options are unchecked
7. Save changes

**Correct Azure configuration**:

Platform configurations

├── Single-page application

└── Redirect URIs

└── http://localhost:5173/auth/callback

Advanced settings

├── Allow public client flows: No

├── Implicit grant and hybrid flows

├── Access tokens: Unchecked

└── ID tokens: Unchecked

**Error 13: Redirect URI Mismatch**

Failed to update Flash8-test application. Error detail: Redirect URIs must have distinct values

**Cause**: Duplicate redirect URIs across multiple platform configurations.

**Solution**: Remove duplicates:

1. Delete all existing platform configurations
2. Add only one **Single-page application** platform
3. Add redirect URI only to this platform: http://localhost:5173/auth/callback

**Don't do this** (multiple platforms with same URI):

❌ Platform configurations

├── Web: http://localhost:5173/auth/callback

├── Single-page application: http://localhost:5173/auth/callback ← Duplicate!

**Do this instead**:

✅ Platform configurations

├── Single-page application

└── Redirect URIs: http://localhost:5173/auth/callback

**SvelteKit Route Errors**

**Error 14: Form Actions vs JSON API Mismatch**

SvelteKitError: Form actions expect form-encoded data — received application/json

**Cause**: Sending JSON data to a route with form actions.

**Problem**: JavaScript fetch() sending JSON to +page.server.ts with actions.

**Solution**: Use correct route types:

**For JSON APIs** - Use +server.ts:

// src/routes/auth/callback/+server.ts

import type { RequestHandler } from './$types';

export const POST: RequestHandler = async ({ request }) => {

const { idToken, account } = await request.json(); // Handle JSON

// Process authentication

return json({ success: true });

};

**For Form Actions** - Use +page.server.ts:

// src/routes/auth/callback/+page.server.ts

import type { Actions } from './$types';

export const actions: Actions = {

default: async ({ request }) => {

const formData = await request.formData(); // Handle form data

// Process form submission

}

};

**Client-side usage**:

// For JSON API (+server.ts)

fetch('/auth/callback', {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify(data)

});

// For Form Action (+page.server.ts)

const formData = new FormData();

formData.append('field', value);

fetch('/auth/callback', {

method: 'POST',

body: formData

});

**Environment Variable Errors**

**Error 15: Missing Environment Variables**

Debug Info:

Client ID: Missing

Tenant ID: Missing

Redirect URI: undefined

**Cause**: Environment variables not loading properly.

**Common Issues**:

1. **Wrong file location**:

# Wrong

src/.env

# Correct

.env # In project root

1. **Wrong variable names**:

# Wrong - missing VITE\_ prefix

AZURE\_CLIENT\_ID=123

# Correct - with VITE\_ prefix for browser access

VITE\_AZURE\_CLIENT\_ID=123

1. **Incorrect format**:

# Wrong - spaces around =

VITE\_AZURE\_CLIENT\_ID = 123

# Correct - no spaces

VITE\_AZURE\_CLIENT\_ID=123

**Solution**: Create correct .env file in project root:

VITE\_AZURE\_CLIENT\_ID=your-client-id-guid

VITE\_AZURE\_TENANT\_ID=your-tenant-id-guid

VITE\_AZURE\_REDIRECT\_URI=http://localhost:5173/auth/callback

COOKIE\_SECRET=your-random-secret-string

**Verification**: Test environment variables:

<!-- src/routes/test-env/+page.svelte -->

<script>

let envVars = {

clientId: import.meta.env.VITE\_AZURE\_CLIENT\_ID,

tenantId: import.meta.env.VITE\_AZURE\_TENANT\_ID,

redirectUri: import.meta.env.VITE\_AZURE\_REDIRECT\_URI

};

</script>

<pre>{JSON.stringify(envVars, null, 2)}</pre>

**File Structure Errors**

**Error 16: Missing CSS File**

Failed to load url ../app.css (resolved id: ../app.css). Does the file exist?

**Cause**: Layout component importing non-existent CSS file.

**Solutions**:

1. **Remove the import** (quickest):

<script lang="ts">

// Remove this line:

// import '../app.css';

</script>

1. **Create the CSS file**:

/\* src/app.css \*/

html, body {

margin: 0;

padding: 0;

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, sans-serif;

}

.button {

padding: 0.75rem 1.5rem;

background-color: #007bff;

color: white;

text-decoration: none;

border-radius: 4px;

}

**Prevention and Best Practices**

**1. Project Setup Checklist**

* [ ] Use npx sv create for new projects
* [ ] Install dependencies with --legacy-peer-deps if needed
* [ ] Run npx svelte-kit sync after setup
* [ ] Verify svelte.config.js exists in project root

**2. Azure Configuration Checklist**

* [ ] Use **Single-page application** platform only
* [ ] Use GUIDs for Client ID and Tenant ID
* [ ] Add correct redirect URI: http://localhost:5173/auth/callback
* [ ] Verify no duplicate redirect URIs
* [ ] Disable implicit grant flows

**3. Environment Variables Checklist**

* [ ] .env file in project root (same level as package.json)
* [ ] Use VITE\_ prefix for browser-accessible variables
* [ ] No spaces around = in .env file
* [ ] Restart dev server after .env changes

**4. TypeScript Configuration Checklist**

* [ ] Run npx svelte-kit sync to generate types
* [ ] Create proper src/app.d.ts with Locals interface
* [ ] Use correct import syntax for CommonJS modules
* [ ] Ensure .svelte-kit/tsconfig.json exists

**5. Route Structure Best Practices**

* [ ] Use +server.ts for JSON APIs
* [ ] Use +page.server.ts for form actions and page data
* [ ] Match client request type with server route type
* [ ] Initialize MSAL before use in components

**Quick Resolution Commands**

# Complete reset and setup

rm -rf node\_modules package-lock.json .svelte-kit

npm install --legacy-peer-deps

npx svelte-kit sync

npm run dev

# Fix TypeScript issues

npx svelte-kit sync

npm run check

# Test environment variables

echo $VITE\_AZURE\_CLIENT\_ID # Should show your client ID

# Debug MSAL in browser console

console.log(import.meta.env.VITE\_AZURE\_CLIENT\_ID);

This document covers all major errors encountered during SvelteKit Microsoft Authentication setup. Keep this as a reference for future implementations and troubleshooting.