

Implementing OAuth for FileMaker using Node.js with Azure AD

Steps to Implement OAuth with Azure AD

1. Configure Azure AD:

- Register your application in Azure AD to obtain the Client ID, Client Secret, and Redirect URI.
- Configure the necessary permissions to access the user's profile.

2. Configure a Node.js Project:

- Create a new Node.js project.
- Install the necessary dependencies.

3. Implement the OAuth Flow:

- Set up the route to initiate the authentication process.
- Implement the OAuth callback to handle the response from Azure AD.
- Obtain the access token and use it to authenticate with the FileMaker API.

Node.js Project Setup

Create a new Node.js project and install the necessary dependencies:

```
```sh
```

```
mkdir filemaker-oauth-azure
```

```
cd filemaker-oauth-azure
```

```
npm init -y
```

```
npm install express axios querystring dotenv
```

## Implementing OAuth for FileMaker using Node.js with Azure AD

...

Create a `.env` file to store your sensitive credentials:

```
```.env
```

```
AZURE_CLIENT_ID=your-azure-client-id
```

```
AZURE_CLIENT_SECRET=your-azure-client-secret
```

```
AZURE_REDIRECT_URI=http://localhost:3000/oauth2callback
```

```
AZURE_TENANT_ID=your-azure-tenant-id
```

```
FILEMAKER_HOST=https://your-filemaker-server.com
```

```
FILEMAKER_DATABASE=your-database-name
```

...

Implementing the Node.js Server

Create a `server.js` file with the following content:

```
```js
```

```
const express = require('express');
```

```
const axios = require('axios');
```

```
const querystring = require('querystring');
```

```
require('dotenv').config();
```

```
const app = express();
```

```
const PORT = 3000;
```

## Implementing OAuth for FileMaker using Node.js with Azure AD

// Endpoint to initiate the OAuth authentication process

```
app.get('/auth', (req, res) => {

 const authUrl =
`https://login.microsoftonline.com/${process.env.AZURE_TENANT_ID}/oauth2/v2.0/authorize?${que
rystring.stringify({
 client_id: process.env.AZURE_CLIENT_ID,
 response_type: 'code',
 redirect_uri: process.env.AZURE_REDIRECT_URI,
 response_mode: 'query',
 scope: 'https://graph.microsoft.com/User.Read openid profile',
 state: '12345',
 })}`;
 res.redirect(authUrl);
});
```

// OAuth callback

```
app.get('/oauth2callback', async (req, res) => {

 const code = req.query.code;

 try {

 const tokenResponse = await
axios.post(`https://login.microsoftonline.com/${process.env.AZURE_TENANT_ID}/oauth2/v2.0/token
`, querystring.stringify({
 client_id: process.env.AZURE_CLIENT_ID,
 client_secret: process.env.AZURE_CLIENT_SECRET,
```

## Implementing OAuth for FileMaker using Node.js with Azure AD

```
code,

redirect_uri: process.env.AZURE_REDIRECT_URI,

grant_type: 'authorization_code',

}), {

 headers: {

 'Content-Type': 'application/x-www-form-urlencoded'

 }

});

const accessToken = tokenResponse.data.access_token;

res.send(`Your access token: ${accessToken}`);

} catch (error) {

 console.error('Error authenticating:', error);

 res.status(500).send('Error authenticating');

}

});

app.listen(PORT, () => {

 console.log(`Server running at http://localhost:${PORT}`);

});

...

```

## Configuring FileMaker Desktop

In FileMaker, you need to configure a script to call the Node.js server and obtain the access token. You can use the FileMaker script to make an HTTP request to your proxy server.

## Implementing OAuth for FileMaker using Node.js with Azure AD

Here is an example of how you can do this:

### 1. Create a Script in FileMaker:

- Go to "Scripts" and create a new script.
- Add a script step to `Insert from URL` that makes a GET request to `http://localhost:3000/auth`

(this will open the browser to authenticate the user).

### 2. Obtain the Access Token:

- After authentication, the user will be redirected to `http://localhost:3000/oauth2callback`.
- FileMaker should wait for the response with the access token.
- Use the access token in subsequent requests to the FileMaker API.

Example of a FileMaker Script:

```
```plaintext
```

```
# Step 1: Open URL to start authentication
```

```
Open URL [ No dialog ; "http://localhost:3000/auth" ]
```

```
# Step 2: Wait for response with access token (simplified example)
```

```
# Here you need a method to capture the access token
```

```
# and store it in a global variable or field
```

```
Insert from URL [ Select ; With dialog: Off ; Target: Table::Field ;  
"http://localhost:3000/oauth2callback" ]
```

Implementing OAuth for FileMaker using Node.js with Azure AD

Step 3: Use the access token to authenticate with the FileMaker API

This is a simplified example of how you could do this

Set Variable [\$token ; Value: Table::Field]

Insert from URL [Select ; With dialog: Off ; Target: Table::Field ;
"https://your-filemaker-server.com/fmi/data/vLatest/databases/your-database/sessions" ; HTTP
Headers: "Authorization: Bearer " & \$token]

...

Considerations

- a. Add more robust error handling in both the Node.js server and the FileMaker script.
- b. Implement a more user-friendly interface in FileMaker to manage the login and token capture process.