**"My Plugin" Documentation**

**Introduction**

"My Plugin" is a plugin developed for Microsoft Security Copilot that allows data retrieval from an external API. This document describes the functionality of the plugin, its main components, and the interactions between the various files in the GitHub repository.

**Plugin Structure**

**1. Descriptor (mio-plugin.json)**

The mio-plugin.json file defines the basic information of the plugin, such as the name, description, and icon. It also specifies the skills and settings necessary for the plugin to function.

**Descriptor Example:**

{

"Descriptor": {

"Name": "my-plugin",

"DisplayName": "My Plugin",

"Description": "A simple plugin for Microsoft Security Copilot",

"Icon": "https://ascanick.github.io/testRepo/icon.png"

},

"SkillGroups": [

{

"Format": "API",

"Settings": {

"OpenApiSpecUrl": "https://ascanick.github.io/testRepo/openapi.yaml"

},

"Skills": [

{

"Name": "FetchData",

"Description": "Fetches data from the API",

"Inputs": []

}

]

}

]

}

**2. OpenAPI Specification (openapi.yaml)**

The openapi.yaml file describes the API that the plugin uses to fetch data. It defines the /data endpoint and the expected response.

**OpenAPI Specification Example:**

openapi: 3.0.0

info:

title: Example API

description: This is an example API for your plugin.

version: 1.0.0

servers:

- url: https://ascanick.github.io/testRepo

paths:

/data:

get:

summary: Fetch data

description: Fetches data from the example API.

responses:

'200':

description: Success

content:

application/json:

schema:

type: object

properties:

message:

type: string

example: "Data fetched successfully"

**3. JavaScript Code (fetchData.js)**

The fetchData.js file contains the code that makes the request to the API and handles the response.

**JavaScript Code Example:**

const axios = require('axios');

async function fetchData() {

try {

const response = await axios.get('https://ascanick.github.io/testRepo/data.json');

console.log(response.data);

} catch (error) {

console.error('Error fetching data:', error);

}

}

fetchData();

**Plugin Functionality**

1. **Initialization**: When the plugin is loaded, the mio-plugin.json file provides the necessary information to configure the plugin.
2. **API Request**: The "FetchData" skill uses the endpoint defined in openapi.yaml to make a GET request to the API.
3. **Data Retrieval**: The JavaScript code in fetchData.js executes the request and logs the retrieved data.
4. **Error Handling**: If an error occurs during the request, it is caught and logged in the console.

**Conclusion**

This plugin demonstrates how to integrate an external API into Microsoft Security Copilot to retrieve and manage data. Ensure that all files are correctly configured and accessible to guarantee the proper functioning of the plugin.