import { sp } from "@pnp/sp/presets/all";

import "@pnp/sp/webs";

import "@pnp/sp/lists";

import "@pnp/sp/items";

import "@pnp/sp/sites";

import "@pnp/sp/appcatalog";

// Define interface for solution package

interface SolutionPackage {

Title: string; // Solution Name

SolutionVersion: string; // Current Version

Status: string; // Active / Deprecated

SiteUrl: string; // Site where deployed

LastUpdated: string; // Last update timestamp

}

// Function to compare versions

function isVersionOutdated(current: string, latest: string): boolean {

const currentParts = current.split(".").map(Number);

const latestParts = latest.split(".").map(Number);

for (let i = 0; i < latestParts.length; i++) {

if ((currentParts[i] || 0) < latestParts[i]) {

return true;

} else if ((currentParts[i] || 0) > latestParts[i]) {

return false;

}

}

return false;

}

// Main function to check solution versions

export async function checkSolutionVersions() {

try {

// Connect to SharePoint

console.log("Checking solution package versions...");

// 1. Get all solutions from App Catalog

const appCatalogSolutions = await sp.site.getAppCatalog().availableApps();

console.log("App Catalog Solutions Found:", appCatalogSolutions.length);

for (let app of appCatalogSolutions) {

const solutionName = app.Title;

const latestVersion = app.InstalledVersion;

console.log(\nSolution: ${solutionName} | Latest Version: ${latestVersion});

// 2. Query all site collections for this solution

const sites = await sp.site.getAllSites(); // Requires Admin Permissions

for (let site of sites) {

const siteUrl = site.Url;

// Get installed apps in the site

const siteApps = await sp.site.getAppCatalog().installedApps();

const siteApp = siteApps.find((sa: any) => sa.Title === solutionName);

if (siteApp) {

const currentVersion = siteApp.InstalledVersion;

// Compare versions

if (isVersionOutdated(currentVersion, latestVersion)) {

console.warn(⚠ Outdated version found at ${siteUrl}: ${currentVersion});

// Optional: Trigger alert to admins

await sendAlertToAdmins(solutionName, currentVersion, latestVersion, siteUrl);

}

}

}

}

console.log("Version check completed.");

} catch (error) {

console.error("Error checking solution versions:", error);

}

}

// Function to send alert (placeholder)

async function sendAlertToAdmins(

solutionName: string,

currentVersion: string,

latestVersion: string,

siteUrl: string

) {

// Example: Add an item to a SharePoint Alerts List

await sp.web.lists.getByTitle("SolutionVersionAlerts").items.add({

Title: solutionName,

CurrentVersion: currentVersion,

LatestVersion: latestVersion,

SiteUrl: siteUrl,

AlertDate: new Date().toISOString(),

Status: "Outdated"

});

console.log(Alert added for ${solutionName} in site ${siteUrl});

}

// Execute script

checkSolutionVersions();