Configuración de Reverse Proxy para API .NET Core en IIS usando Node.js

# Introducción

Este documento describe cómo configurar un reverse proxy para una aplicación API .NET Core publicada en IIS, utilizando Node.js para redirigir las solicitudes. El objetivo es ocultar el puerto real de la API y proporcionar una URL base más amigable.

# Pre-requisitos

- Visual Studio instalado.  
- IIS instalado en el servidor Windows.  
- Aplicación API .NET Core lista para ser publicada.  
- Node.js y npm instalados.

# Paso 1: Publicar la API .NET Core en IIS

1. Compilar la aplicación en modo Release desde Visual Studio.  
2. Publicar la aplicación en IIS seleccionando 'IIS, FTP, etc.' como método de publicación y configurando los detalles necesarios.  
3. Asegurarse de que IIS esté configurado para usar el puerto 443 con SSL.  
4. Verificar que la aplicación funcione correctamente accediendo a través de la URL configurada en IIS.

# Paso 2: Instalar Node.js

1. Descargar la última versión LTS de Node.js desde la página oficial y completar la instalación.  
2. Verificar la instalación ejecutando `node -v` y `npm -v` en una terminal.

# Paso 3: Configurar el Reverse Proxy con Node.js

1. Crear un nuevo directorio para el proyecto de reverse proxy y ejecutar `npm init -y` para inicializar un nuevo proyecto de Node.js.  
2. Instalar las dependencias necesarias con `npm install express http-proxy-middleware pm2 --save`.  
3. Crear un archivo `index.js` con el siguiente código para configurar el reverse proxy:

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

**const { createProxyMiddleware } = require('http-proxy-middleware');**

**const app = express();**

**// Configuración del reverse proxy para redirigir solicitudes a la API .NET Core**

**const apiProxy = createProxyMiddleware('/MyProjectTempApi', {**

**target: 'http://localhost', // Apunta al host donde tu API está corriendo**

**changeOrigin: true,**

**pathRewrite: { '^/MyProjectTempApi': '/MyProjectTempApi' }, // No necesita reescribir en este caso**

**logLevel: 'debug' // Habilita la depuración para ver qué está pasando**

**});**

**app.use('/MyProjectTempApi', apiProxy);**

**const PORT = 30000;**

**app.listen(PORT, () => {**

**console.log(`Reverse proxy corriendo en http://localhost:${PORT}`);**

**});**

4. Usar PM2 para iniciar el reverse proxy con `pm2 start index.js --name my-reverse-proxy`.  
5. Verificar el funcionamiento accediendo a `http://localhost:30000/ProjectName/Api/Controlador/Metodo`.

6. Para consumir el ReverseProxy es importante validar como se esta configurando en IIS ya que puede ser como localhost o con el nombre de dominio que se asigno, en este caso se dejo el localhost: http://localhost:30000/MyProjectTempApi/Api/ArchDesign/GetAll

# Detener el Reverse Proxy con PM2

Para gestionar o detener el reverse proxy que está corriendo a través de PM2, puedes utilizar los siguientes comandos:

## Detener el Reverse Proxy

pm2 stop my-reverse-proxy

## Detener Todos los Procesos de PM2

pm2 stop all

## Eliminar el Proceso de la Lista de PM2

pm2 delete my-reverse-proxy

pm2 delete all

## Verificar el Estado de los Procesos

pm2 status

## Reiniciar un Proceso

pm2 restart my-reverse-proxy

**Limpieza del Caché de NPM**

En algunas situaciones, puede ser necesario limpiar el caché de los paquetes instalados por npm para resolver problemas de dependencias o actualizaciones. Aunque npm gestiona su propio caché, en ocasiones podrías querer eliminar manualmente los datos guardados en el caché o verificar las instalaciones de paquetes a nivel global.

## Ubicaciones del Caché de NPM

Los datos de caché y los módulos globales de npm se almacenan en diferentes ubicaciones según el sistema operativo y la versión. Aquí te proporcionamos las rutas típicas donde puedes encontrar estos archivos en sistemas Windows:

**En Windows XP:**

Módulos Globales: %USERPROFILE%\Application Data\npm\node\_modules

**En Versiones más Recientes de Windows (Vista y posteriores):**

Módulos Globales: %AppData%\npm\node\_modules

Otra Ubicación Común: %AppData%\Roaming\npm\node\_modules

## Cómo Limpiar el Caché de NPM

Para una gestión eficiente del caché, npm ofrece un comando específico que te permite limpiar el caché de manera segura y efectiva, eliminando cualquier dato almacenado que ya no sea necesario o que pueda estar causando conflictos. Puedes ejecutar el siguiente comando en tu terminal o línea de comandos para limpiar el caché de npm:

npm cache clean –force

Nota: Usar --force es necesario porque npm evita la limpieza del caché en circunstancias normales, ya que esto puede afectar al rendimiento de las instalaciones posteriores. Sin embargo, esta acción puede ser útil para resolver problemas específicos.

## Consideraciones

Antes de proceder con la limpieza del caché, asegúrate de que no haya procesos de npm en ejecución que puedan verse afectados. Limpiar el caché es una operación segura, pero se recomienda utilizarla cuando se encuentren problemas específicos que no se puedan resolver con comandos más convencionales como npm install o npm update.