

# Configurando NPM

## Puntos Clave

- Parece probable que la configuración de NPM implique editar el archivo `.npmrc` para establecer configuraciones de proxy y registro, lo cual se puede hacer manualmente o mediante comandos.
- La investigación sugiere que las configuraciones de proxy requieren el formato `proxy = http://user:password@proxy` con caracteres especiales codificados en URL.
- La evidencia apunta a establecer el registro como `registry = https://registry.npmjs.org/` por defecto, con opciones para deshabilitar la verificación SSL si es necesario.

## Configurando NPM y Trabajando con `.npmrc`

NPM, o Node Package Manager, es una herramienta para gestionar dependencias de JavaScript, y su configuración a menudo implica el archivo `.npmrc` para configuraciones como el registro y el proxy.

**¿Qué es `.npmrc` y Dónde se Encuentra?** El archivo `.npmrc` es un archivo de configuración para NPM, permitiendo la personalización de su comportamiento. Puede ser: - **Global:** Encontrado en `~/.npmrc` en sistemas tipo Unix o `%APPDATA%\npm\npmrc` en Windows (generalmente `C:\Users\<username>\AppData\Roaming\npm\npmrc`). - **Local a un proyecto:** Ubicado en `./package/.npmrc` en el directorio raíz del proyecto. Las configuraciones locales anulan las globales.

**Configurando Proxy y HTTPS Proxy** Para trabajar detrás de un proxy, agregue estas líneas a `.npmrc`:  
`proxy = http://user:password@proxy_server:port - https_proxy = http://user:password@proxy_server:port`

Tenga en cuenta que incluso para `https_proxy`, use `http://` como protocolo. Si su nombre de usuario o contraseña tiene caracteres especiales (por ejemplo, `\`, `@`), códelos en URL (por ejemplo, `domain\user` se convierte en `domain%5Cuser`). También puede configurarlos mediante comandos: `- npm config set proxy http://user:password@proxy_server:port - npm config set https_proxy http://user:password@proxy_server:port`

**Configurando el Registro** El registro es la URL que NPM usa para obtener paquetes, generalmente configurado como: `- registry = https://registry.npmjs.org/`

Para registros personalizados, reemplace con su URL (por ejemplo, un registro privado de la empresa). Configúrelo mediante: `- npm config set registry https://your_custom_registry.com/`

**Manejo de SSL y Verificación** Si enfrenta problemas de SSL, considere: `- strict-ssl = false` en `.npmrc` para ignorar errores de SSL, aunque esto es menos seguro. Configúrelo mediante: `- npm config set strict-ssl false`

**Usando y Verificando Configuraciones** Edite `.npmrc` directamente usando un editor de texto, o use comandos `npm config set`. Para verificar: - Liste todas las configuraciones con `npm config list`. - Verifique una configuración específica con `npm config get proxy`. - Depure problemas con `npm install --verbose` para obtener una salida detallada.

Un archivo `.npmrc` de ejemplo podría verse así:

```
proxy = http://user:password@proxy_server:8080
https_proxy = http://user:password@proxy_server:8080
registry = https://registry.npmjs.org/
strict-ssl = false
```

## **Nota de Encuesta: Guía Completa para Configurar NPM con `.npmrc`**

Esta sección proporciona una exploración detallada de la configuración de NPM, centrándose en el archivo `.npmrc` para configuraciones de registro y proxy, ampliando la respuesta directa con contexto adicional y detalles técnicos.

**Introducción a `.npmrc` y Su Rol** El archivo `.npmrc` es un archivo de configuración crítico para NPM, habilitando la personalización de comportamientos de gestión de paquetes. Soporta un formato INI con pares clave-valor, donde los comentarios comienzan con un punto y coma (;). Puede estar ubicado globalmente o localmente: - **Ubicación Global:** En sistemas tipo Unix, está en `~/.npmrc`; en Windows, en `%APPDATA%\npm\npmrc`, típicamente `C:\Users\<username>\AppData\Roaming\npm\npmrc`. - **Ubicación Local:** Dentro del directorio raíz del proyecto como `./package/.npmrc`. Las configuraciones locales anulan las globales, proporcionando configuraciones específicas del proyecto.

Esta jerarquía asegura flexibilidad, permitiendo a los desarrolladores gestionar configuraciones tanto a nivel de usuario como de proyecto, lo cual es particularmente útil en entornos con múltiples proyectos o sistemas compartidos.

**Configuración Detallada del Proxy** Trabajar detrás de un proxy requiere configurar tanto `proxy` como `https_proxy` en `.npmrc`. El formato es: - `proxy = http://user:password@proxy_server:port` - `https_proxy = http://user:password@proxy_server:port`

Notablemente, `https_proxy` usa `http://` como protocolo, no `https://`, debido a que el servidor proxy maneja la conexión, no la cifra hacia el proxy mismo. Este es un punto común de confusión, ya que la carga útil hacia el destino permanece cifrada con SSL.

Para la autenticación, incluya el nombre de usuario y la contraseña, pero si contienen caracteres especiales, la codificación en URL es esencial: - Barra invertida (`\`) se convierte en `%5C`. - Símbolo arroba (`@`) se convierte en `%40`. - Dos puntos (`:`) se convierte en `%3A`.

Por ejemplo, si el nombre de usuario es `domain\user` y la contraseña es `pass@word`, sería `proxy = http://domain%5Cuser:pass%40word@proxy_server:8080`. Esta codificación asegura un análisis correcto por parte de NPM.

Los usuarios de Windows, especialmente en entornos corporativos con Active Directory, pueden enfrentar problemas con barras invertidas. Algunos informes indican que `domain\username` en `.npmrc` podría convertirse en `domain/username` al ser leído, potencialmente causando fallos de autenticación. En tales casos, herramientas como NTLM Authorization Proxy Server o Cntlm pueden ayudar, aunque están fuera de la configuración directa de `.npmrc`.

**Configuración del Registro y Consideraciones de SSL** La configuración del registro define de dónde NPM obtiene paquetes, estableciéndose por defecto en `https://registry.npmjs.org/`. Para registros personalizados o privados, configure: `- registry = https://your_custom_registry.com/`

Si la verificación SSL falla, a menudo debido a proxies corporativos con certificados autofirmados, puede deshabilitar SSL estricto: `- strict-ssl = false`

Sin embargo, esto reduce la seguridad, ya que omite la validación del certificado. Alternativamente, configure la autoridad de certificación: `- Use npm config -g set cafile [YOUR_CERTIFICATE_DIR]/[CERTIFICATE_NAME].crt` para especificar un archivo CA confiable.

Algunos desarrolladores optan por `registry = http://registry.npmjs.org/` para evitar SSL por completo, pero esto no se recomienda por razones de seguridad, especialmente en producción.

**Usando .npmrc: Edición Manual vs. Línea de Comandos** Puede editar `.npmrc` directamente con cualquier editor de texto, asegurándose de que cada configuración esté en una nueva línea con el formato `key = value`. Los comentarios comienzan con `;`, por ejemplo:

```
; Configuraciones de proxy para la red corporativa
proxy = http://user:password@proxy_server:8080
https_proxy = http://user:password@proxy_server:8080
registry = https://registry.npmjs.org/
strict-ssl = false
```

Alternativamente, use comandos NPM para conveniencia: `- Establecer proxy: npm config set proxy http://user:password@proxy_server:port` `- Establecer registro: npm config set registry https://registry.npmjs.org/` `- Establecer strict-ssl: npm config set strict-ssl false`

Estos comandos actualizan el archivo `.npmrc` apropiado según el contexto (global o local). Para cambiar entre global y local, use `-g` para global, por ejemplo, `npm config set -g proxy ....`

**Verificación y Solución de Problemas** Para verificar configuraciones, use: `- npm config list` para listar todas las configuraciones, mostrando tanto los valores efectivos como los predeterminados. `- npm config get proxy` para recuperar una configuración específica.

Para solucionar problemas, especialmente detrás de proxies, ejecute: `- npm install --verbose` para ver registros detallados, lo cual puede ayudar a identificar problemas de conexión, como `ECONNREFUSED` o `socket hang up`.

Si las configuraciones no surten efecto, verifique las variables de entorno (`HTTP_PROXY`, `HTTPS_PROXY`) que podrían anular `.npmrc`. Estas pueden configurarse a nivel del sistema y tener precedencia, por lo que asegúrese de la consistencia.

## Mejores Prácticas y Consideraciones Adicionales

- **Seguridad:** Evite `strict-ssl = false` en producción; en su lugar, configure certificados confiables.
- **Variables de Entorno:** Tenga en cuenta que `NODE_ENV` y `HTTPS_PROXY` pueden anular configuraciones de `.npmrc`, útiles para pipelines CI/CD pero potencialmente confusas localmente.
- **Niveles de Registro:** Ajuste `loglevel` en `.npmrc` (por ejemplo, `loglevel = http`) para obtener una salida más detallada durante la depuración, con opciones que van de `silent` a `silly` (7 niveles en total).
- **Proxies Corporativos:** Para proxies NTLM, considere herramientas como `Cntlm`, configurando un servidor proxy local y configurando `.npmrc` para apuntar a `localhost:3128`, simplificando la autenticación.

**Configuraciones de Ejemplo** Aquí hay una tabla que resume configuraciones comunes de `.npmrc` para proxy y registro, con ejemplos:

| Configuración | Descripción  | Ejemplo  |
|---------------|--|--|
| proxy         | URL del servidor proxy HTTP                            | <code>proxy = http://user:password@proxy:8080</code>       |
| https_proxy   | URL del servidor proxy HTTPS                           | <code>https_proxy = http://user:password@proxy:8080</code> |
| registry      | URL del registro de paquetes                           | <code>registry = https://registry.npmjs.org/</code>        |
| strict-ssl    | Habilitar/deshabilitar verificación SSL                | <code>strict-ssl = false</code>                            |
| loglevel      | Nivel de salida de consola (silent a silly, 7 niveles) | <code>loglevel = http</code>                               |

Esta tabla ayuda en la referencia rápida, especialmente para desarrolladores nuevos en `.npmrc`.

**Detalle Inesperado: Impacto en Configuraciones Globales vs. Locales** Un aspecto inesperado es cómo las configuraciones globales de `.npmrc` afectan a todos los proyectos, mientras que las configuraciones locales pueden anularlas para proyectos específicos. Esta capa puede llevar a confusión si, por ejemplo, se establece un proxy global pero un proyecto local usa un registro diferente, potencialmente causando problemas de autenticación. Comprender esta jerarquía es crucial para entornos con múltiples proyectos.

**Conclusión** Configurar NPM a través de `.npmrc` para configuraciones de proxy y registro es sencillo con el formato correcto y consideraciones para la codificación y SSL. Editando directamente o usando comandos, y verificando con `npm config list`, asegura una gestión de paquetes fluida, especialmente en redes corporativas con proxies. Para lectura adicional, consulte la documentación oficial en NPM CLI Config o explore mejores prácticas en Configuring Your `.npmrc`.

## Citaciones Clave

- Documentación de NPM CLI Config
- Configuring Your `.npmrc` for an Optimal Node.js Environment
- How to setup npm proxy settings behind proxy on Windows