

Configurer NPM

Points Clés

- Il semble probable que la configuration de NPM implique la modification du fichier `.npmrc` pour définir les paramètres de proxy et de registre, ce qui peut être fait manuellement ou via des commandes.
- La recherche suggère que les paramètres de proxy nécessitent le format `proxy = http://user:password@proxy_server:port` avec les caractères spéciaux codés en URL.
- Les preuves penchent en faveur de la définition du registre comme `registry = https://registry.npmjs.org/` par défaut, avec des options pour désactiver la vérification SSL si nécessaire.

Configuration de NPM et Utilisation de `.npmrc`

NPM, ou Node Package Manager, est un outil pour gérer les dépendances JavaScript, et sa configuration implique souvent le fichier `.npmrc` pour les paramètres comme le registre et le proxy.

Qu'est-ce que `.npmrc` et Où se Trouve-t-il ? Le fichier `.npmrc` est un fichier de configuration pour NPM, permettant de personnaliser son comportement. Il peut être : - **Global** : Trouvé à `~/.npmrc` sur les systèmes Unix-like ou `%APPDATA%\npm\npmrc` sur Windows (généralement `C:\Users\<username>\AppData\Roaming\npm\npmrc`). - **Local à un projet** : Situé à `./package/.npmrc` dans le répertoire racine du projet. Les paramètres locaux remplacent les paramètres globaux.

Définir le Proxy et le Proxy HTTPS Pour travailler derrière un proxy, ajoutez ces lignes à `.npmrc` : `- proxy = http://user:password@proxy_server:port - https_proxy = http://user:password@proxy_server:port`

Notez que même pour `https_proxy`, utilisez `http://` comme protocole. Si votre nom d'utilisateur ou mot de passe contient des caractères spéciaux (par exemple, `\`, `@`), encodez-les en URL (par exemple, `domain\user` devient `domain%5Cuser`). Vous pouvez également les définir via des commandes : `- npm config set proxy http://user:password@proxy_server:port - npm config set https_proxy http://user:password@proxy_server:port`

Définir le Registre Le registre est l'URL utilisée par NPM pour récupérer les packages, généralement défini comme : `- registry = https://registry.npmjs.org/`

Pour des registres personnalisés, remplacez par votre URL (par exemple, un registre privé d'entreprise). Définissez-le via : `- npm config set registry https://your_custom_registry.com/`

Gestion SSL et Vérification Si vous rencontrez des problèmes SSL, envisagez : `- strict-ssl = false` dans `.npmrc` pour ignorer les erreurs SSL, bien que cela soit moins sécurisé. Définissez-le via : `- npm config set strict-ssl false`

Utilisation et Vérification des Configurations Éditez `.npmrc` directement à l'aide d'un éditeur de texte, ou utilisez les commandes `npm config set`. Pour vérifier : - Listez tous les paramètres avec `npm config list`. - Vérifiez un paramètre spécifique avec `npm config get proxy`. - Débuguez les problèmes avec `npm install --verbose` pour un output détaillé.

Un exemple de fichier `.npmrc` pourrait ressembler à ceci :

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

Note de Sondage : Guide Complet pour Configurer NPM avec `.npmrc`

Cette section offre une exploration détaillée de la configuration de NPM, se concentrant sur le fichier `.npmrc` pour les paramètres de registre et de proxy, en approfondissant la réponse directe avec des contextes et détails techniques supplémentaires.

Introduction à `.npmrc` et Son Rôle Le fichier `.npmrc` est un fichier de configuration critique pour NPM, permettant de personnaliser les comportements de gestion des packages. Il prend en charge un format INI avec des paires clé-valeur, où les commentaires commencent par un point-virgule (;). Il peut être situé globalement ou localement : - **Emplacement Global** : Sur les systèmes Unix-like, il se trouve à `~/.npmrc` ; sur Windows, à `%APPDATA%\npm\npmrc`, généralement `C:\Users\<username>\AppData\Roaming\npm\npmrc`. - **Emplacement Local** : Dans le répertoire racine d'un projet sous la forme `./package/.npmrc`. Les paramètres locaux remplacent les paramètres globaux, fournissant des configurations spécifiques au projet.

Cette hiérarchie assure la flexibilité, permettant aux développeurs de gérer les paramètres à la fois au niveau utilisateur et projet, ce qui est particulièrement utile dans les environnements avec plusieurs projets ou systèmes partagés.

Configuration Proxy Détaillée Travailler derrière un proxy nécessite de définir à la fois `proxy` et `https_proxy` dans `.npmrc`. Le format est : - `proxy = http://user:password@proxy_server:port` - `https_proxy = http://user:password@proxy_server:port`

Notamment, `https_proxy` utilise `http://` comme protocole, et non `https://`, car le serveur proxy gère la connexion, sans chiffrer la connexion vers le proxy lui-même. C'est un point de confusion courant, car la charge utile vers la destination reste chiffrée SSL.

Pour l'authentification, incluez le nom d'utilisateur et le mot de passe, mais si ceux-ci contiennent des caractères spéciaux, le codage URL est essentiel : - Le backslash (\) devient `%5C`. - Le symbole @ devient `%40`. - Le deux-points (:) devient `%3A`.

Par exemple, si le nom d'utilisateur est `domain\user` et le mot de passe est `pass@word`, il serait `proxy = http://domain%5Cuser:pass%40word@proxy_server:8080`. Ce codage assure un bon parsing par NPM.

Les utilisateurs Windows, surtout dans des environnements d'entreprise avec Active Directory, peuvent rencontrer des problèmes avec les backslashes. Certains rapports indiquent que `domain\username` dans `.npmrc` pourrait se convertir en `domain/username` lors de la lecture, potentiellement causant des échecs d'authentification. Dans de tels cas, des outils comme NTLM Authorization Proxy Server ou Cntlrm peuvent aider, bien qu'ils soient en dehors de la configuration directe de `.npmrc`.

Configuration du Registre et Considérations SSL Le paramètre de registre définit d'où NPM récupère les packages, par défaut `https://registry.npmjs.org/`. Pour des registres personnalisés ou privés, définissez : `- registry = https://your_custom_registry.com/`

Si la vérification SSL échoue, souvent en raison de proxies d'entreprise avec des certificats auto-signés, vous pouvez désactiver le SSL strict : `- strict-ssl = false`

Cependant, cela réduit la sécurité, car cela contourne la validation des certificats. Alternativement, configurez l'autorité de certification : - Utilisez `npm config -g set cafile [YOUR_CERTIFICATE_DIR]/[CERTIFICATE_NAME].crt` pour spécifier un fichier CA de confiance.

Certains développeurs optent pour `registry = http://registry.npmjs.org/` pour éviter SSL entièrement, mais cela n'est pas recommandé pour des raisons de sécurité, surtout en production.

Utilisation de `.npmrc` : Édition Manuelle vs. Ligne de Commande Vous pouvez éditer `.npmrc` directement avec n'importe quel éditeur de texte, en vous assurant que chaque paramètre est sur une nouvelle ligne avec le format `key = value`. Les commentaires commencent par `;`, par exemple :

```
; Paramètres proxy pour le réseau d'entreprise
proxy = http://user:password@proxy_server:8080
https_proxy = http://user:password@proxy_server:8080
registry = https://registry.npmjs.org/
strict-ssl = false
```

Alternativement, utilisez les commandes NPM pour plus de commodité : - Définir le proxy : `npm config set proxy http://user:password@proxy_server:port` - Définir le registre : `npm config set registry https://registry.npmjs.org/` - Définir strict-ssl : `npm config set strict-ssl false`

Ces commandes mettent à jour le fichier `.npmrc` approprié en fonction du contexte (global ou local). Pour basculer entre global et local, utilisez `-g` pour global, par exemple, `npm config set -g proxy`

Vérification et Dépannage Pour vérifier les configurations, utilisez : - `npm config list` pour lister tous les paramètres, montrant à la fois les valeurs effectives et par défaut. - `npm config get proxy` pour récupérer un paramètre spécifique.

Pour le dépannage, surtout derrière des proxies, exécutez : `- npm install --verbose` pour voir les logs détaillés, ce qui peut aider à identifier les problèmes de connexion, tels que `ECONNREFUSED` ou `socket hang up`.

Si les paramètres ne prennent pas effet, vérifiez les variables d'environnement (`HTTP_PROXY`, `HTTPS_PROXY`) qui pourraient remplacer `.npmrc`. Elles peuvent être définies au niveau système et prennent la priorité, alors assurez-vous de la cohérence.

Meilleures Pratiques et Considérations Supplémentaires

- **Sécurité** : Évitez `strict-ssl = false` en production ; au lieu de cela, configurez des certificats de confiance.
- **Variables d'Environnement** : Soyez conscient que `NODE_ENV` et `HTTPS_PROXY` peuvent remplacer les paramètres `.npmrc`, utile pour les pipelines CI/CD mais potentiellement déroutant localement.
- **Niveaux de Log** : Ajustez `loglevel` dans `.npmrc` (par exemple, `loglevel = http`) pour un output plus détaillé pendant le débogage, avec des options allant de `silent` à `silly` (7 niveaux au total).
- **Proxies d'Entreprise** : Pour les proxies NTLM, envisagez des outils comme `Cntlm`, en configurant un serveur proxy local et en configurant `.npmrc` pour pointer vers `localhost:3128`, simplifiant l'authentification.

Configurations Exemples Voici un tableau résumant les paramètres `.npmrc` courants pour le proxy et le registre, avec des exemples :

Paramètre	Description	Exemple
proxy	URL du serveur proxy HTTP	<code>proxy = http://user:password@proxy:8080</code>
https_proxy	URL du serveur proxy HTTPS	<code>https_proxy = http://user:password@proxy:8080</code>
registry	URL du registre de packages	<code>registry = https://registry.npmjs.org/</code>
strict-ssl	Activer/désactiver la vérification SSL	<code>strict-ssl = false</code>
loglevel	Niveau de sortie console (silent à silly, 7 niveaux)	<code>loglevel = http</code>

Ce tableau aide à la référence rapide, surtout pour les développeurs nouveaux à `.npmrc`.

Détail Inattendu : Impact des Paramètres Globaux vs. Locaux Un aspect inattendu est la manière dont les paramètres globaux `.npmrc` affectent tous les projets, tandis que les paramètres locaux peuvent les remplacer pour des projets spécifiques. Cette stratification peut entraîner de la confusion si, par exemple, un proxy global est défini mais un projet local utilise un registre différent, potentiellement causant des problèmes d'authentification. Comprendre cette hiérarchie est crucial pour les environnements multi-projets.

Conclusion Configurer NPM via `.npmrc` pour les paramètres de proxy et de registre est simple avec le bon format et les considérations pour le codage et SSL. En éditant directement ou en utilisant des commandes, et en vérifiant avec `npm config list`, vous assurez une gestion fluide des packages, surtout dans les réseaux d'entreprise avec des proxies. Pour une lecture supplémentaire, consultez la documentation officielle à NPM CLI Config ou explorez les meilleures pratiques à Configuring Your `.npmrc`.

Citations Clés

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