

# Tester le registre NPM en utilisant des requêtes HTTP brutes

Pour tester votre registre NPM en utilisant des requêtes HTTP brutes avec des outils comme `curl` ou `Insomnia`, vous pouvez vérifier son fonctionnement en vérifiant la connectivité, l'authentification et la capacité à servir des données de package—tout cela sans dépendre du client NPM. Comme votre registre nécessite un nom d'utilisateur et un mot de passe, vous devrez gérer l'authentification manuellement dans vos requêtes. Voici un guide complet pour tester votre registre en utilisant des requêtes HTTP brutes, en supposant qu'il s'agit d'un registre compatible avec NPM (par exemple, Verdaccio, Nexus, ou une implémentation personnalisée) qui prend en charge l'authentification basée sur des jetons ou de base.

---

## Étapes pour tester le registre avec des requêtes HTTP brutes

### 1. Authentification et obtention d'un jeton (si nécessaire)

La plupart des registres NPM utilisent une authentification basée sur des jetons, vous obligeant à vous connecter pour obtenir un jeton avant de faire d'autres requêtes. Certains registres peuvent également prendre en charge l'authentification de base directement. Voici comment vous authentifier en utilisant `curl` :

**Utilisation de `curl` pour se connecter** Envoyez une requête PUT à l'endpoint d'authentification du registre pour obtenir un jeton :

```
curl -X PUT \
  -H "Content-Type: application/json" \
  -d '{"name": "<username>", "password": "<password>"}' \
  <registry-url>/-/user/org.couchdb.user:<username>
```

- **Remplacer :**

- `<username>` : Votre nom d'utilisateur de registre.
- `<password>` : Votre mot de passe de registre.
- `<registry-url>` : L'URL complète de votre registre (par exemple, `https://my-registry.example.com`).

- **Réponse attendue** : Si réussi, vous obtiendrez une réponse JSON avec un jeton :

```
{
  "token": "your-auth-token"
}
```

- **Enregistrer le jeton** : Copiez la valeur `your-auth-token` pour une utilisation dans les requêtes ultérieures.

**Note** : Si votre registre utilise un endpoint d'authentification différent ou une méthode (par exemple, l'authentification de base ou une API personnalisée), consultez sa documentation. Si elle prend en charge l'authentification de base directement, vous pouvez sauter cette étape et inclure `-u "<username>:<password>"` dans les requêtes ultérieures.

---

## 2. Ping du registre

Testez la connectivité de base au registre en envoyant une requête GET à son URL racine ou à un endpoint de ping.

### Utilisation de `curl` pour ping

```
curl -H "Authorization: Bearer your-auth-token" <registry-url>
```

- **Remplacer** :

- `your-auth-token` : Le jeton de l'Étape 1.
- `<registry-url>` : Votre URL de registre.

- **Réponse attendue** : Une réponse réussie (HTTP 200) pourrait retourner la page d'accueil du registre ou un simple message d'état (par exemple, `{"db_name":"registry"}` pour les registres basés sur CouchDB).

- **Alternative** : Certains registres offrent un endpoint `/-/ping` :

```
curl -H "Authorization: Bearer your-auth-token" <registry-url>/-/ping
```

**Si vous utilisez l'authentification de base** : Si votre registre n'utilise pas de jetons et prend en charge l'authentification de base :

```
curl -u "<username>:<password>" <registry-url>
```

---

## 3. Récupérer les métadonnées du package

Vérifiez que le registre peut servir les métadonnées du package en demandant des détails pour un package spécifique.

## Utilisation de `curl` pour obtenir des métadonnées

```
curl -H "Authorization: Bearer your-auth-token" <registry-url>/<package-name>
```

- **Remplacer :**

- <package-name> : Un package que vous savez exister sur votre registre (par exemple, `lodash` s'il proxy le registre public, ou un package privé comme `my-org-utils`).

- **Réponse attendue :** Un objet JSON avec les métadonnées du package, y compris les versions, les dépendances et les URL des archives. Par exemple :

```
{
  "name": "lodash",
  "versions": {
    "4.17.21": {
      "dist": {
        "tarball": "<registry-url>/lodash/-/lodash-4.17.21.tgz"
      }
    }
  }
}
```

### Si vous utilisez l'authentification de base :

```
curl -u "<username>:<password>" <registry-url>/<package-name>
```

- **Succès :** Une réponse 200 OK avec des métadonnées confirme que le registre sert correctement les données de package.

---

## 4. Télécharger une archive de package (facultatif)

Pour tester pleinement le registre, téléchargez une archive de package pour vous assurer qu'il peut livrer les fichiers de package réels.

### Utilisation de `curl` pour télécharger une archive

1. À partir des métadonnées de l'Étape 3, trouvez l'URL de l'archive pour une version spécifique (par exemple, `<registry-url>/lodash/-/lodash-4.17.21.tgz`).
2. Téléchargez-la :

```
curl -H "Authorization: Bearer your-auth-token" -O <tarball-url>
```

- **Remplacer** : <tarball-url> par l'URL des métadonnées.
- **Flag -O** : Enregistre le fichier avec son nom d'origine (par exemple, lodash-4.17.21.tgz).
- **Si vous utilisez l'authentification de base** :

```
curl -u "<username>:<password>" -O <tarball-url>
```

- **Succès** : Le fichier se télécharge avec succès, et vous pouvez l'extraire (par exemple, avec tar -xzf <filename>) pour vérifier son contenu.
- 

## Test avec Insomnia

Si vous préférez un outil GUI comme Insomnia, suivez ces étapes :

### 1. Configurer l'authentification

- Créez une nouvelle requête dans Insomnia.
- Allez à l'onglet **Auth** :
  - **Bearer Token** : Si vous avez obtenu un jeton à l'Étape 1, sélectionnez "Bearer Token" et collez your-auth-token.
  - **Basic Auth** : Si le registre utilise l'authentification de base, sélectionnez "Basic Auth" et entrez votre <username> et <password>.

### 2. Ping du registre

- **Méthode** : GET
- **URL** : <registry-url> ou <registry-url>/-/ping
- Cliquez sur **Send**.
- **Réponse attendue** : Un statut 200 OK avec un corps de réponse simple.

### 3. Récupérer les métadonnées du package

- **Méthode** : GET
- **URL** : <registry-url>/<package-name>
- Assurez-vous que l'authentification est définie dans l'onglet Auth.
- Cliquez sur **Send**.
- **Réponse attendue** : Un statut 200 OK avec des métadonnées de package en JSON.

## 4. Télécharger une archive

- **Méthode** : GET
  - **URL** : L'URL de l'archive à partir des métadonnées (par exemple, `<registry-url>/<package-name>/-/<package-name>`)
  - Cliquez sur **Send and Download** pour enregistrer le fichier localement.
  - **Succès** : Le fichier se télécharge, confirmant que le registre sert les packages.
- 

## Dépannage

- **401 Non autorisé** :
    - Vérifiez votre jeton ou vos informations d'identification.
    - Assurez-vous que l'en-tête `Authorization` ou l'authentification de base est correctement formatée.
  - **404 Introuvable** :
    - Vérifiez que le package existe sur le registre.
    - Si vous testez des packages publics (par exemple, `lodash`), confirmez que le registre proxy le registre NPM public.
  - **Erreurs de connexion** :
    - Vérifiez à nouveau l'URL du `<registry-url>` (par exemple, incluez `https://` si nécessaire).
    - Pour les registres HTTPS, si vous obtenez des erreurs SSL avec `curl`, ajoutez `-k` pour contourner les vérifications de certificat (pour le test uniquement).
  - **Comportement du registre personnalisé** :
    - Si ces étapes échouent, consultez la documentation de votre registre pour des endpoints ou des méthodes d'authentification spécifiques.
- 

## Conclusion

En suivant ces étapes avec `curl` ou `Insomnia`, vous pouvez tester les capacités de connectivité, d'authentification et de service de packages de votre registre NPM en utilisant des requêtes HTTP brutes. Commencez par l'authentification (si nécessaire), puis pingez le registre, récupérez les métadonnées pour un package connu et téléchargez éventuellement une archive. Cette approche garantit que votre registre est pleinement opérationnel au niveau HTTP.