

Pruebas del registro de NPM utilizando solicitudes HTTP en bruto

Para probar tu registro de NPM utilizando solicitudes HTTP brutas con herramientas como `curl` o Insomnia, puedes verificar su funcionalidad comprobando la conectividad, la autenticación y la capacidad de servir datos de paquetes, todo sin depender del cliente de NPM. Dado que tu registro requiere un nombre de usuario y una contraseña, deberás manejar la autenticación manualmente en tus solicitudes. A continuación, se presenta una guía completa para probar tu registro utilizando solicitudes HTTP brutas, asumiendo que es un registro compatible con NPM (por ejemplo, Verdaccio, Nexus o una implementación personalizada) que admite autenticación basada en tokens o básica.

Pasos para Probar el Registro con Solicitudes HTTP Brutas

1. Autenticar y Obtener un Token (si es Necesario)

La mayoría de los registros de NPM utilizan autenticación basada en tokens, lo que requiere que inicies sesión para obtener un token antes de realizar solicitudes adicionales. Algunos registros también pueden admitir autenticación básica directamente. Aquí te mostramos cómo autenticar utilizando `curl`:

Usando curl para Iniciar Sesión Envía una solicitud PUT al punto final de autenticación del registro para obtener un token:

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

- **Reemplazar:**

- `<username>`: Tu nombre de usuario del registro.
- `<password>`: Tu contraseña del registro.
- `<registry-url>`: La URL completa de tu registro (por ejemplo, <https://my-registry.example.com>).

- **Respuesta Esperada:** Si es exitoso, obtendrás una respuesta JSON con un token:

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

- **Guardar el Token:** Copia el valor de `your-auth-token` para usarlo en solicitudes posteriores.

Nota: Si tu registro utiliza un punto final de autenticación diferente o un método (por ejemplo, autenticación básica o una API personalizada), consulta su documentación. Si admite autenticación básica directamente, puedes omitir este paso e incluir `-u "<username>:<password>"` en las solicitudes posteriores.

2. Ping al Registro

Prueba la conectividad básica al registro enviando una solicitud GET a su URL raíz o a un punto final de ping.

Usando curl para Ping

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

- **Reemplazar:**

- `your-auth-token`: El token del Paso 1.
- `<registry-url>`: La URL de tu registro.

- **Respuesta Esperada:** Una respuesta exitosa (HTTP 200) podría devolver la página principal del registro o un simple mensaje de estado (por ejemplo, `{"db_name": "registry"}` para registros basados en CouchDB).

- **Alternativa:** Algunos registros ofrecen un punto final `/-/ping`:

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

Si Usas Autenticación Básica: Si tu registro no usa tokens y admite autenticación básica:

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

3. Obtener Metadatos del Paquete

Verifica que el registro pueda servir metadatos del paquete solicitando detalles para un paquete específico.

Usando curl para Obtener Metadatos

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

- **Reemplazar:**

- <package-name>: Un paquete que sabes que existe en tu registro (por ejemplo, `lodash` si proxies el registro público, o un paquete privado como `my-org-utils`).
- **Respuesta Esperada:** Un objeto JSON con los metadatos del paquete, incluidos las versiones, dependencias y URLs de archivos tar.gz. Por ejemplo:

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

Si Usas Autenticación Básica:

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

- **Éxito:** Una respuesta 200 OK con metadatos confirma que el registro está sirviendo datos de paquetes correctamente.
-

4. Descargar un Archivo Tarball del Paquete (Opcional)

Para probar completamente el registro, descarga un archivo tarball del paquete para asegurarte de que puede entregar los archivos del paquete real.

Usando curl para Descargar un Tarball

1. A partir de los metadatos del Paso 3, encuentra la URL del tarball para una versión específica (por ejemplo, `<registry-url>/lodash/-/lodash-4.17.21.tgz`).
2. Descárgalo:

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

- **Reemplazar:** `<tarball-url>` con la URL de los metadatos.
- **Bandera -O:** Guarda el archivo con su nombre original (por ejemplo, `lodash-4.17.21.tgz`).

- **Si Usas Autenticación Básica:**

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

- **Éxito:** El archivo se descarga exitosamente y puedes extraerlo (por ejemplo, con tar -xzf <filename>) para verificar su contenido.
-

Prueba con Insomnia

Si prefieres una herramienta de interfaz gráfica como Insomnia, sigue estos pasos:

1. Configurar Autenticación

- Crea una nueva solicitud en Insomnia.
- Ve a la pestaña **Auth**:

- **Token de Portador:** Si obtuviste un token en el Paso 1, selecciona “Bearer Token” y pega your-auth-token.
- **Autenticación Básica:** Si el registro usa autenticación básica, selecciona “Basic Auth” e ingresa tu <username> y <password>.

2. Ping al Registro

- **Método:** GET
- **URL:** <registry-url> o <registry-url>/-/ping
- Haz clic en **Send**.
- **Respuesta Esperada:** Un estado 200 OK con un cuerpo de respuesta simple.

3. Obtener Metadatos del Paquete

- **Método:** GET
- **URL:** <registry-url>/<package-name>
- Asegúrate de que la autenticación esté configurada en la pestaña Auth.
- Haz clic en **Send**.
- **Respuesta Esperada:** Un estado 200 OK con metadatos del paquete en JSON.

4. Descargar un Tarball

- **Método:** GET
 - **URL:** La URL del tarball de los metadatos (por ejemplo, <registry-url>/<package-name>/-/<package-name>-<versión>.tarball)
 - Haz clic en **Send and Download** para guardar el archivo localmente.
 - **Éxito:** El archivo se descarga, confirmando que el registro sirve paquetes.
-

Solución de Problemas

- **401 No Autorizado:**

- Verifica tu token o credenciales.
- Asegúrate de que el encabezado `Authorization` o la autenticación básica esté correctamente formateado.

- **404 No Encontrado:**

- Verifica que el paquete exista en el registro.
- Si pruebas paquetes públicos (por ejemplo, `lodash`), confirma que el registro proxies el registro público de NPM.

- **Errores de Conexión:**

- Verifica la <registry-url> (por ejemplo, incluye `https://` si es necesario).
- Para registros HTTPS, si obtienes errores SSL con `curl`, agrega `-k` para omitir las comprobaciones de certificados (solo para pruebas).

- **Comportamiento de Registro Personalizado:**

- Si estos pasos fallan, consulta la documentación de tu registro para puntos finales específicos o métodos de autenticación.
-

Conclusión

Siguiendo estos pasos con `curl` o `Insomnia`, puedes probar la conectividad, autenticación y capacidades de servicio de paquetes de tu registro de NPM utilizando solicitudes HTTP brutas. Comienza autenticando (si es necesario), luego pinge el registro, recupera metadatos para un paquete conocido y opcionalmente descarga un tarball. Este enfoque asegura que tu registro esté completamente operativo a nivel HTTP.