

# Testing NPM registry Using Raw HTTP Requests

To test your NPM registry using raw HTTP requests with tools like `curl` or Insomnia, you can verify its functionality by checking connectivity, authentication, and the ability to serve package data—all without relying on the NPM client. Since your registry requires a username and password, you'll need to handle authentication manually in your requests. Below is a complete guide to test your registry using raw HTTP requests, assuming it's an NPM-compatible registry (e.g., Verdaccio, Nexus, or a custom implementation) that supports token-based or basic authentication.

---

## Steps to Test the Registry with Raw HTTP Requests

### 1. Authenticate and Obtain a Token (if Needed)

Most NPM registries use token-based authentication, requiring you to log in to get a token before making further requests. Some registries might also support basic authentication directly. Here's how to authenticate using `curl`:

**Using `curl` to Log In** Send a PUT request to the registry's authentication endpoint to obtain a token:

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

- **Replace:**

- `<username>`: Your registry username.
- `<password>`: Your registry password.
- `<registry-url>`: The full URL of your registry (e.g., `https://my-registry.example.com`).

- **Expected Response:** If successful, you'll get a JSON response with a token:

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

- **Save the Token:** Copy the `your-auth-token` value for use in subsequent requests.

**Note:** If your registry uses a different authentication endpoint or method (e.g., basic auth or a custom API), check its documentation. If it supports basic auth directly, you can skip this step and include `-u "<username>:<password>"` in later requests instead.

## 2. Ping the Registry

Test basic connectivity to the registry by sending a GET request to its root URL or a ping endpoint.

### Using `curl` to Ping

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

- **Replace:**

- `your-auth-token`: The token from Step 1.
- `<registry-url>`: Your registry URL.

- **Expected Response:** A successful response (HTTP 200) might return the registry's homepage or a simple status message (e.g., `{"db_name": "registry"}` for CouchDB-based registries).

- **Alternative:** Some registries offer a `/-/ping` endpoint:

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

**If Using Basic Auth:** If your registry doesn't use tokens and supports basic authentication:

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

---

## 3. Retrieve Package Metadata

Verify that the registry can serve package metadata by requesting details for a specific package.

### Using `curl` to Get Metadata

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

- **Replace:**

- `<package-name>`: A package you know exists on your registry (e.g., `lodash` if it proxies the public registry, or a private package like `my-org-utils`).

- **Expected Response:** A JSON object with the package's metadata, including versions, dependencies, and tarball URLs. For example:

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

### If Using Basic Auth:

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

- **Success:** A 200 OK response with metadata confirms the registry is serving package data correctly.

## 4. Download a Package Tarball (Optional)

To fully test the registry, download a package tarball to ensure it can deliver the actual package files.

### Using `curl` to Download a Tarball

1. From the metadata in Step 3, find the `tarball` URL for a specific version (e.g., `<registry-url>/lodash/-/lodash-4.17.21.tgz`).
2. Download it:

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

- **Replace:** `<tarball-url>` with the URL from the metadata.
- **-O Flag:** Saves the file with its original name (e.g., `lodash-4.17.21.tgz`).
- **If Using Basic Auth:**

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

- **Success:** The file downloads successfully, and you can extract it (e.g., with `tar -xzf <filename>`) to verify its contents.

## Testing with Insomnia

If you prefer a GUI tool like Insomnia, follow these steps:

### 1. Set Up Authentication

- Create a new request in Insomnia.
- Go to the **Auth** tab:
  - **Bearer Token**: If you obtained a token in Step 1, select “Bearer Token” and paste `your-auth-token`.
  - **Basic Auth**: If the registry uses basic auth, select “Basic Auth” and enter your `<username>` and `<password>`.

### 2. Ping the Registry

- **Method**: GET
- **URL**: `<registry-url>` OR `<registry-url>/-/ping`
- Click **Send**.
- **Expected Response**: A 200 OK status with a simple response body.

### 3. Retrieve Package Metadata

- **Method**: GET
- **URL**: `<registry-url>/<package-name>`
- Ensure authentication is set in the Auth tab.
- Click **Send**.
- **Expected Response**: A 200 OK status with package metadata in JSON.

### 4. Download a Tarball

- **Method**: GET
  - **URL**: The tarball URL from the metadata (e.g., `<registry-url>/<package-name>/-/<package-name>-<version>.tgz`).
  - Click **Send and Download** to save the file locally.
  - **Success**: The file downloads, confirming the registry serves packages.
- 

## Troubleshooting

- **401 Unauthorized**:

- Check your token or credentials.
- Ensure the `Authorization` header or basic auth is correctly formatted.

- **404 Not Found:**

- Verify the package exists on the registry.
- If testing public packages (e.g., `lodash`), confirm the registry proxies the public NPM registry.

- **Connection Errors:**

- Double-check the `<registry-url>` (e.g., include `https://` if needed).
- For HTTPS registries, if you get SSL errors with `curl`, add `-k` to bypass certificate checks (for testing only).

- **Custom Registry Behavior:**

- If these steps fail, consult your registry's documentation for specific endpoints or authentication methods.
- 

## Conclusion

By following these steps with `curl` or `Insomnia`, you can test your NPM registry's connectivity, authentication, and package-serving capabilities using raw HTTP requests. Start by authenticating (if required), then ping the registry, retrieve metadata for a known package, and optionally download a tarball. This approach ensures your registry is fully operational at the HTTP level.