

OpenID Connect को OAuth 2.0 का उपयोग करके परीक्षण

आपका OpenID रजिस्ट्री को कूल या इन्सोमनिया जैसी टूल्स के साथ रॉ हट्टप रिक्वेस्ट्स का उपयोग करके टेस्ट करने के लिए, आप इसका कार्यक्षमता को जांचने के लिए कनेक्टिविटी, ऑथेंटिकेशन, और पैकेज डेटा सर्व करने की क्षमता को सत्यापित कर सकते हैं—सभी बिना OpenID क्लाइंट पर निर्भर करने के। क्योंकि आपका रजिस्ट्री एक यूजरनेम और पासवर्ड की आवश्यकता करता है, आपको अपने रिक्वेस्ट्स में ऑथेंटिकेशन को मैनुअल रूप से हैंडल करना होगा। नीचे एक पूर्ण गाइड है जो रॉ हट्टप रिक्वेस्ट्स का उपयोग करके अपने रजिस्ट्री को टेस्ट करने के लिए है, मानते हुए कि यह एक OpenID-समर्थित रजिस्ट्री है (जैसे कि वेडैक्वियो, नेक्सस, या एक कस्टम इम्प्लीमेंटेशन) जो टोकन-आधारित या बेसिक ऑथेंटिकेशन का समर्थन करता है।

रजिस्ट्री को रॉ हट्टप रिक्वेस्ट्स के साथ टेस्ट करने के चरण

1. ऑथेंटिकेशन और टोकन प्राप्त करना (अगर आवश्यक हो)

बहुत से OpenID रजिस्ट्रियाँ टोकन-आधारित ऑथेंटिकेशन का उपयोग करती हैं, जो आपको आगे के रिक्वेस्ट्स करने से पहले लॉगिन करने के लिए एक टोकन प्राप्त करने की आवश्यकता होती है। कुछ रजिस्ट्रियाँ भी बेसिक ऑथेंटिकेशन का समर्थन कर सकते हैं। यहाँ का उपयोग करके ऑथेंटिकेशन करने का तरीका है:

का उपयोग करके लॉगिन करना एक टोकन प्राप्त करने के लिए रजिस्ट्री के ऑथेंटिकेशन एंडपॉइंट पर एक OpenID रिक्वेस्ट भेजें:

```
-X PUT \
-H " - : / " \
-d '{" ": "< >", " ": "< >"}' \
< - >/-/ / . . :< >
```

बदलें:

- < >: आपका रजिस्ट्री यूजरनेम।
- < >: आपका रजिस्ट्री पासवर्ड।
- < - >: आपका रजिस्ट्री का पूर्ण यूआरएल (जैसे कि https:// - . .)।

अपेक्षित रिस्पॉन्स: अगर सफल, तो आप एक OpenID रिस्पॉन्स प्राप्त करेंगे जिसमें एक टोकन होगा:

```
{
  " ": " - - "
}
```

टोकन को सेभ करें: - - का मान को अगले रिक्वेस्ट्स में उपयोग करने के लिए कॉपी करें।

नोट: अगर आपका रजिस्ट्री एक अलग ऑथेंटिकेशन एंडपॉइंट या विधि का उपयोग करता है (जैसे कि बेसिक ऑथ या एक कस्टम एपीआई), तो इसके दस्तावेज़ को चेक करें। अगर यह बेसिक ऑथ को सीधे समर्थन करता है, तो आप इस चरण को छूट सकते हैं और अगले रिक्वेस्ट्स में -u "< >:< >" शामिल कर सकते हैं।

2. रजिस्ट्री को पिंग करना

रजिस्ट्री के साथ बेसिक कनेक्टिविटी को टेस्ट करने के लिए, एक `ping` रिक्वेस्ट भेजें जिसका लक्ष्य रूट यूआरएल या एक पिंग एंडपॉइंट है।

का उपयोग करके पिंग करना

```
-H " : - - " < - >
```

❑ बदलें:

- - - : चरण 1 से टोकन।
- < - >: आपका रजिस्ट्री यूआरएल।

❑ **अपेक्षित रिस्पॉन्स:** एक सफल रिस्पॉन्स (200) रजिस्ट्री के होमपेज या एक सरल स्टेटस संदेश (जैसे कि `{ " _ " : " " }` को `ping`-आधारित रजिस्ट्रियों के लिए) लौट सकता है।

❑ **अल्टर्नेटिव:** कुछ रजिस्ट्रियाँ एक `/-/` एंडपॉइंट का समर्थन कर सकते हैं:

```
-H " : - - " < - >/-/
```

बेसिक ऑथ का उपयोग करने के लिए: अगर आपका रजिस्ट्री टोकन का उपयोग नहीं करता और बेसिक ऑथेंटिकेशन का समर्थन करता है:

```
-u "< >: < >" < - >
```

3. पैकेज मेटाडेटा प्राप्त करना

रजिस्ट्री द्वारा पैकेज मेटाडेटा को सर्व करने की क्षमता को सत्यापित करने के लिए, एक विशिष्ट पैकेज के लिए विवरणों का अनुरोध करें।

का उपयोग करके मेटाडेटा प्राप्त करना

```
-H " : - - " < - >/< - >
```

❑ बदलें:

- < - >: एक पैकेज जिसका आप जानते हैं कि यह आपका रजिस्ट्री पर मौजूद है (जैसे कि अगर यह सार्वजनिक रजिस्ट्री को प्रॉक्सी करता है, या एक निजी पैकेज जैसे कि - -)।

❑ **अपेक्षित रिस्पॉन्स:** एक ऑब्जेक्ट जिसमें पैकेज के मेटाडेटा, जिसमें संस्करण, निर्भरताएं, और टारबॉल यूआरएल शामिल हैं। उदाहरण के लिए:

```
{
  " ": " ",
  " ": {
    "4.17.21": {
      " ": {
        " ": "< - >/ /-/-4.17.21. "
      }
    }
  }
}
```

बेसिक ऑथ का उपयोग करने के लिए:

```
-u "< >:< >" < - >/< - >
```

□ **सफलता:** एक 200 □ रिस्पॉन्स के साथ मेटाडेटा, यह सत्यापित करता है कि रजिस्ट्री पैकेज डेटा को सही से सर्व कर रहा है।

4. एक पैकेज टारबॉल डाउनलोड करना (वैकल्पिक)

रजिस्ट्री को पूरी तरह से टेस्ट करने के लिए, एक पैकेज टारबॉल डाउनलोड करें ताकि यह वास्तविक पैकेज फाइलों को डिलीवर कर सके।

का उपयोग करके एक टारबॉल डाउनलोड करना

- चरण 3 में मेटाडेटा से, एक विशिष्ट संस्करण के लिए यूआरएल को ढूंढें (जैसे कि < - >/ /-/-4.17.21.).
- इसे डाउनलोड करें:

```
-H " : - - " -0 < - >
```

□ **बदलें:** < - > को मेटाडेटा से यूआरएल से बदलें।

□ **-0 फ्लैग:** फाइल को उसके मूल नाम के साथ सेभ करता है (जैसे कि -4.17.21.).

□ **बेसिक ऑथ का उपयोग करने के लिए:**

```
-u "< >:< >" -0 < - >
```

□ **सफलता:** फाइल सफलतापूर्वक डाउनलोड होती है, और आप इसे (जैसे कि -xzf < > के साथ) निकाल सकते हैं ताकि इसकी सामग्री को सत्यापित कर सकें।

इन्सोमनिया के साथ टेस्टिंग

अगर आप एक `test` टूल जैसे कि इन्सोमनिया का उपयोग करना पसंद करते हैं, तो इन चरणों का पालन करें:

1. ऑथेंटिकेशन सेट अप करना

- इन्सोमनिया में एक नया रिक्वेस्ट बनाएं।
- **ऑथ** टैब पर जाएं:
 - **बियरर टोकन**: अगर आपने चरण 1 में एक टोकन प्राप्त किया है, तो "बियरर टोकन" चुनें और `-` को पेस्ट करें।
 - **बेसिक ऑथ**: अगर रजिस्ट्री बेसिक ऑथ का उपयोग करता है, तो "बेसिक ऑथ" चुनें और `<` `>` और `<` `>` दर्ज करें।

2. रजिस्ट्री को पिंग करना

- **मैथड**: `GET`
- **यूआरएल**: `< - >` या `< - >/-/`
- **सेंड** पर क्लिक करें।
- **अपेक्षित रिस्पॉन्स**: एक 200 `OK` स्टेटस के साथ एक सरल रिस्पॉन्स बॉडी।

3. पैकेज मेटाडेटा प्राप्त करना

- **मैथड**: `GET`
- **यूआरएल**: `< - >/< - >`
- ऑथ टैब में ऑथेंटिकेशन सेट किया जाए, यह सुनिश्चित करें।
- **सेंड** पर क्लिक करें।
- **अपेक्षित रिस्पॉन्स**: एक 200 `OK` स्टेटस के साथ पैकेज मेटाडेटा `JSON` में।

4. एक टारबॉल डाउनलोड करना

- **मैथड**: `GET`
- **यूआरएल**: मेटाडेटा से टारबॉल यूआरएल (जैसे कि `< - >/< - >/-/< - >-< >.`)।
- **सेंड और डाउनलोड** पर क्लिक करें ताकि फाइल को स्थानीय रूप से सेभ करें।
- **सफलता**: फाइल डाउनलोड होती है, यह सत्यापित करती है कि रजिस्ट्री पैकेजों को सर्व कर रहा है।

समस्या निदान

- **401 अनधिकृत**:
 - आपका टोकन या क्रेडेंशियल्स चेक करें।

- सुनिश्चित करें कि हेडर या बेसिक ऑथ सही रूप में फॉर्मेट किया गया है।

❑ 404 न मिला:

- सुनिश्चित करें कि पैकेज रजिस्ट्री पर मौजूद है।
- अगर सार्वजनिक पैकेजों (जैसे कि) को टेस्ट कर रहे हैं, तो सुनिश्चित करें कि रजिस्ट्री सार्वजनिक `public` रजिस्ट्री को प्रॉक्सी करता है।

❑ कनेक्टिविटी त्रुटियां:

- `< - >` को दोबारा चेक करें (जैसे कि `https://` की आवश्यकता हो तो शामिल करें)।
- `localhost` रजिस्ट्रियों के लिए, अगर `localhost` के साथ `localhost` त्रुटियां होती हैं, तो `-k` को जोड़ें ताकि सर्टिफिकेट चेक्स को बाईपास करें (केवल टेस्टिंग के लिए)।

❑ कस्टम रजिस्ट्री व्यवहार:

- अगर इन चरणों में विफलता होती है, तो अपने रजिस्ट्री के दस्तावेज़ को चेक करें ताकि विशिष्ट एंडपॉइंट या ऑथेंटिकेशन विधियों के लिए।

निष्कर्ष

या इन्सोमनिया के साथ इन चरणों का पालन करके, आप अपने `localhost` रजिस्ट्री के कनेक्टिविटी, ऑथेंटिकेशन, और पैकेज-सर्व करने की क्षमता को रॉ हटप रिक्वेस्ट्स का उपयोग करके टेस्ट कर सकते हैं। ऑथेंटिकेशन (अगर आवश्यक हो) से शुरू करें, फिर रजिस्ट्री को पिंग करें, एक जाने-माने पैकेज के लिए मेटाडेटा प्राप्त करें, और वैकल्पिक रूप से एक टारबॉल डाउनलोड करें। यह रास्ता सुनिश्चित करता है कि आपका रजिस्ट्री `localhost` स्तर पर पूरी तरह से कार्यक्षम है।