Packetizer™

Introduction to WebFinger

Paul E. Jones

2013-10-21

What Is WebFinger?

> WebFinger is used to *discover information* about people or other entities on the Internet that are identified by a URI using standard Hypertext Transfer Protocol (HTTP) methods over a secure transport.

-- IETF RFC 7033

Expanding on that, WebFinger…

- Is a means of discovering information about people and things on the Internet
- Is designed for the web, using HTTP and JSON
- Can be used to discover information about any URI, where the URI might identify a person's account at a service provider, a web page, a physical device, or anything else
- Utilizes HTTPS to ensure that whatever is shared via WebFinger can be delivered to the requesting application securely
- Is *not* a protocol that scours the Internet digging for data

Exactly What Is “Discovered”?

- The simple answer is “any information a person wishes to share”
- On the next few slides, there are examples of the kinds of information that might be useful for different entity types
- What is important to keep in mind is that what can be “discovered” via WebFinger is *only* the information a person *wishes to share* with the public
- It is also important to note that anything shared via WebFinger is accessible by the entire world, just like anything else posted to the Internet

Information People Might Share

- The person’s name, perhaps even presented in multiple languages (e.g., Chinese and Pinyin for English speakers)
- A picture one wants to use on social networking sites
- A Bitcoin address so one can easily receive money
- A public key used to allow people to send encrypted email messages
- An OpenID Connect provider URI to allow one to easily log into websites
- A business card so that people can get contact information
- The address of one’s blog or RSS/ATOM feed

What Might Be Published About a Web Page

- Copyright information for the site or the content
- The author’s name or URI
- The date the page was published or last updated
- The title of the page

Information About a Device

- The name of the device
- The location of the device
- The MAC addresses associated with the device
- An enterprise asset identifier
- A URI that might be used to query for dynamic information about the device (e.g., the toner level in a printer or the internal CPU temperature of a server)

WebFinger Protocol

Specific Focus on User Accounts

The “acct” URI

- People are identified on the Internet many different ways, though WebFinger needs a URI and it needs to be consistent
- Users have grown accustomed to identifying people using identifiers like bob@example.com and @bob
- The “acct” URI was introduced to identify user accounts, which has a trivial syntax, is protocol-neutral, and contains a host component
- Examples:
  - acct:bob@example.com
  - acct:bob@twitter.com

Issuing a WebFinger Query

- Given a URI, the client issues a query to a “well-known” address of the *host* associated with the URI
- For example, let’s consider “acct:paulej@packetizer.com”
- Below is the HTTP query that would be issued to get information about the URI

```
GET /.well-known/webfinger?resource=acct%3Apaulej%40packetizer.com
Host: packetizer.com
```

Server Response

- The server will respond with a JSON payload called a JSON Resource Descriptor (JRD)
  - Content-Type: application/jrd+json
- Servers utilize CORS to allow queries to be successfully issued by web browsers
  - Access-Control-Allow-Origin: *

Example JRD

```
{
  "subject" : "acct:paulej@packetizer.com",
  "aliases" :
  [
    "h323:paulej@packetizer.com"
  ],
  "properties" :
  {
    "http://packetizer.com/ns/name" : "Paul E. Jones",
    "http://packetizer.com/ns/name#zh-CN" : "保罗·琼斯",
    "http://packetizer.com/ns/activated" : "2000-02-17T03:00:00Z"
  },
  "links" :
  [
    {
      "rel" : "http://webfinger.net/rel/avatar",
      "type" : "image/jpeg",
      "href" : "http://www.packetizer.com/people/paulej/images/paulej.jpg"
    },
    {
      "rel" : "http://specs.openid.net/auth/2.0/provider",
      "href" : "https://openid.packetizer.com/paulej"
    },
    {
      "rel" : "http://webfinger.net/rel/profile-page",
      "type" : "text/html",
      "href" : "http://www.packetizer.com/people/paulej/"
    },
    {
      "rel" : "http://packetizer.com/rel/blog",
      "type" : "text/html",
      "href" : "http://www.packetizer.com/people/paulej/blog/",
      "titles" :
      {
        "en-us" : "Paul E. Jones' Blog"
      }
    },
    {
      "rel" : "http://packetizer.com/rel/businesscard",
      "type" : "text/vcard",
      "href" : "http://www.packetizer.com/people/paulej/paulej.vcf"
    },
    {
      "rel" : "http://schemas.google.com/g/2010#updates-from",
      "type" : "application/atom+xml",
      "href" : "http://www.packetizer.com/people/paulej/blog/blog.xml"
    },
    {
      "rel" : "http://packetizer.com/rel/smime-cert",
      "type" : "application/pkix-cert",
      "href" : "http://www.packetizer.com/people/paulej/paulej_smime.cer"
    },
    {
      "rel" : "http://bitcoin.org/rel/address",
      "href" : "bitcoin:17XoqvUCrf12H7Vc7c7uDxib8FDMXFx2p6"
    }
  ]
}
```

Structure of the JRD

- Subject: the URI of the entity whose information is returned
- Aliases: an array of alternative URIs that return the same set of properties and links
- Properties: strings that contain unstructured information about the subject (e.g., a person’s name) and identified by property identifiers
- Links: an array of objects that point to information about the subject, further comprising
  - The link relation type
  - The expected content-type to be returned
  - The URI pointing to the content
  - Various properties related to the link
  - One or more titles for the link (e.g., the name of a blog)

The Subject

- This is merely the URI of the entity whose information is returned and would normally be the same as what was queried
- From the example:

```
    "subject" : "acct:paulej@packetizer.com"
```

Aliases

- These are an array of alternative URIs that return the same set of properties and links
- From the example:

```
"aliases" :
[
  "h323:paulej@packetizer.com"
]
```

Properties (Subject-Specific)

- Properties are strings that contain unstructured information about the subject (e.g., a person’s name) and identified by property identifiers
- From the example:

```
"properties" :
{
  "http://packetizer.com/ns/name" : "Paul E. Jones",
  "http://packetizer.com/ns/name#zh-CN" : "保罗·琼斯",
  "http://packetizer.com/ns/activated" : "2000-02-17T03:00:00Z"
}
```

Links

- Links are an array of objects that point to information about the subject
- From the example:

```
"links" :
[
  {
    "rel" : "http://webfinger.net/rel/avatar",
    "type" : "image/jpeg",
    "href" : "http://www.packetizer.com/people/paulej/images/paulej.jpg"
  },
  ...
]
```

Links Can Have Titles

- In addition to an “href” pointing to a location, links can have titles
- Links may have titles in multiple languages
- From the example:

```
{
  "rel" : "http://packetizer.com/rel/blog",
  "type" : "text/html",
  "href" : "http://www.packetizer.com/people/paulej/blog/",
  "titles" :
  {
    "en-us" : "Paul E. Jones' Blog"
  }
}
```

Links Can Point Nowhere

- A link does not necessarily have to point anywhere, but might simply have an href with a URI that is understood by the client
- From the example:

```
{
  "rel" : "http://bitcoin.org/rel/address",
  "href" : "bitcoin:17XoqvUCrf12H7Vc7c7uDxib8FDMXFx2p6"
}
```

Links Might Not Have an “href” Member

- Some links might be entirely self-describing with no “href” member
- Properties are used to convey meaning to the client
- Consider this example that might prescribe to an email client how to provision the SMTP server settings

```
{
  "rel" : "http://packetizer.com/rel/smtp-server",
  "properties" :
  {
    "http://packetizer.com/ns/host" : "dublin.packetizer.com",
    "http://packetizer.com/ns/port" : "587",
    "http://packetizer.com/ns/login-required" : "yes",
    "http://packetizer.com/ns/transport" : "starttls"
  }
}
```

Why Use WebFinger?

Just a Few Use Cases

Improving the Social Experience

- Suppose you receive an email from somebody that talks about a blog post they made
- You might request your email client to perform a WebFinger query to get more information about the sender of the email message, the response perhaps including
  - The sender’s blog URL
  - The sender’s picture
  - The sender’s social networking profile pages
  - The sender’s other contact information
- With this information, you could then visit the sender’s blog

Making It Easier to Post Comments

- Each time you visit a new web site to post a comment, you are often asked for information like your email address, your homepage, etc.
- To create a more complete profile on a site, you might also upload a picture, insert links to your others social networking sites (Twitter, your own blog, Google+ profile page, Facebook page, etc.)
- Much of this can be automated via WebFinger by providing the site with your user identifier (e.g., “paulej@packetizer.com”) and having that information in your primary account

Make It Easier to Receive Money

- Bitcoin represents a significant step forward in enabling people to send money to each other from anywhere in the world, make payments for goods and services, etc.
- The most “unfriendly” aspect of Bitcoin are the long addresses, which are strings that look like “17XoqvUCrf12H7Vc7c7uDxib8FDMXFx2p6”
    - Try reading that to somebody over the phone
- One can put his or her Bitcoin address online so that it can be queried via WebFinger, thus making it possible to request the Bitcoin client to send money to “paulej@packetizer.com”, for example

Enabling Secure Certificate Exchange

- S/MIME is a popular means of encrypting email messages
- A challenge with using S/MIME, however, is getting the public certificate of the person to whom you wish to send an encrypted email message
  - If you do not know the public certificate of the person to whom you wish to send encrypted email, you cannot send an encrypted message
- Using WebFinger, it is possible to easily discover a person's public certificate

Associating Your Different Accounts

- The web is full of sites that allow you to share your thoughts, spew a few random words, publish pictures, upload videos, etc. and people often manually insert cross-references to their other accounts
- With WebFinger, your various accounts can point back to your “primary” account so that users can learn more about you without you trying to keep all of the information up-to-date on lots of sites
- For example, if Twitter offered a WebFinger service, you could insert a single link that points back to your primary account
- Your primary account could have links to all of the user’s other accounts, including YouTube, Google+, Facebook, Flickr, etc.

More Information

- Visit http://www.packetizer.com/webfinger/ to get more information about WebFinger, including
  - Links to client and server software
  - Server configuration examples
  - Additional usage examples
  - Lists of link relation types, property values, etc.
  - Pointers to other sites with useful information about WebFinger

Packetizer™