

## Yadis: Service Discovery for Identities like OpenID (Part 2)

[Continuation of Part-1...](#)

### Service Descriptions - The Yadis Resource Descriptor (XRD)

After all this fuss and running about, it's easy to miss the point of Yadis - getting that final Yadis XRD document that describes the Services associated with a given ID. Sticking with our OpenID example, we know that in order to start authenticating a user on an OpenID Server, we need a few bits of information:

1. The OpenID Server URL; the URL to which HTTP requests will be made.
2. The OpenID Delegate: The ID of the current user on the OpenID Server (not their alias!)
3. The OpenID Service Types offered: For OpenID, this can include Signon 1.0, Signon 1.1 and Simple Registration (sreg) 1.0.

Here's an example XRD document a website might fetch when performing Service Discovery:

```
<?xml version="1.0" encoding="UTF-8"?>
<xrds:XRDS
  xmlns:xrds="xri://$xrds"
  xmlns:openid="http://openid.net/xmlns/1.0"
  xmlns="xri://$xrd*(v*2.0)">
  <XRD>
    <Service priority="0">
      <Type>http://openid.net/signon/1.0</Type>
      <Type>http://openid.net/signon/1.1</Type>
      <Type>http://openid.net/sreg/1.0</Type>
      <URI>http://www.example.com/server</URI>
      <openid:Delegate>http://username.example.com/</openid:Delegate>
    </Service>
  </XRD>
</xrds:XRDS>
```

I'll skip the details of parsing XML - you can use PHP dom or SimpleXML in PHP5. The XML document follows the XRD format, included in a current OASIS specification. At it's most basic, the XRD document (noting these requirements are for Yadis 1.0 only) must contain a single XRD element composed of one or more Service elements. Each Service element defines a Service, as detailed in its child Type elements.

The Type element must always contain a valid URL or XRI (don't worry yet about the XRI bit - it's a proposed addition compatible with the current URI and IRI specifications). The URL should point to a Service specification, and contain a Version. If you check above, our example OpenID Provider is offering Signon 1.0-1.1 and Simple Registration 1.0.

The URI element must contain a valid URL - this is the URL the services described by the Type values are provided from, i.e. it's where a website supporting OpenID logins would send its association and/or authentication requests.

Finally, there is an optional element "openid:Delegate". This namespaced element contains the OpenID URL a user's OpenID Provider knows them as (remember, any other URL can be an alias but the OpenID Provider does not know or care about such aliases).

## Conclusion

At the end of this Yadis introduction, I'll refer anyone who's stuck with it to the official Yadis Specification 1.0. It's not a huge document, but has a few nuances I've likely skipped mentioning.


<http://yadis.org/papers/yadis-v1.0.pdf>

Next up I'll jump into the details of OpenID!

Posted by Pádraic Brady in PHP General at 02:55

Great tutorial.

Do you have any plans for supporting Yadis discovery via SMTP in your library? Anonymous on Feb 22 2007, 04:14

At the moment the Zend Framework Proposal (check the mailing list, and the proposal section of the ZF wiki soon) will focus entirely on the core 1.0 Yadis spec. Once that has been fully implemented, I'll then look into supporting extensions. So short answer is yes...eventually. 

If the SMTP extension has any specific requirements by all means throw them and any details at me - the more fully I'm aware of what an extension would need, the less likely it will require any major refactoring of the core code to implement. Anonymous on Feb 22 2007, 06:15

Great! 

All that's needed is to tell URLs, XRIs and emails apart (shouldn't be too difficult -- already hacked the JanRain implementation), connect to an SMTP server and send a XRDS command with the email address. A successful response would be a URL of the XRDS document.

I still didn't get to hacking PostFix to make it all work, but I hope that will be done within a couple of months. Anonymous on Feb 22 2007, 07:24

Your comment will already have one impact - I need to make sure the code specifies the discovery ID as ID (not URL/XRI which are getting a bit meaningless once you consider IRI and email...).

The class constructor can be amended to check whether that if a Yadis ID is not a valid URL, that it's one of the others. To be honest I have not checked whether the current Zend\_Uri class is compatible with IRI's - should look into that. Anonymous on Feb 22 2007, 16:50