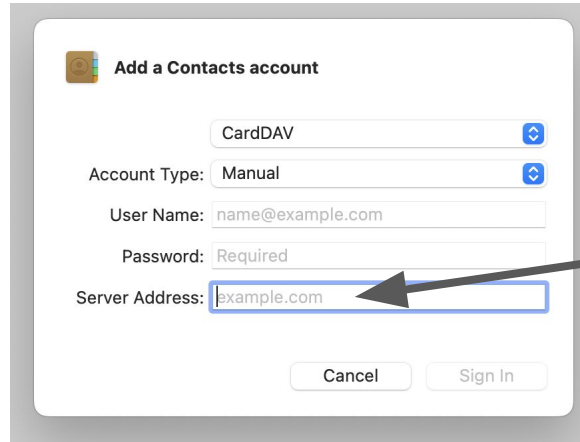


OAuth 2.0 Protected Resource Metadata now with WWW-Authenticate

<https://datatracker.ietf.org/doc/html/draft-ietf-oauth-resource-metadata-01>

What is this for?

- Enables clients to dynamically learn about and use protected resources they may have no prior knowledge of
- e.g., calendar / email apps that work with many resource servers, and
- authorization servers with no prior relationship to the calendar / email app



Add a Contacts account

CardDAV

Account Type: Manual

User Name: name@example.com

Password: Required

Server Address: example.com

Cancel Sign In

User configures client with a particular resource server

Progress Since IETF 117

- Working group adopted Resource Server Metadata
- Including WWW-Authenticate response
 - Resource returns its Resource Identifier and other WWW-Authenticate info
 - Aaron and Mike are both happy with the result
- Addressed many comments on the draft using GitHub Issues
 - Published result as draft -01
- If you're interested, see the closed issues at:

<https://github.com/oauth-wg/draft-ietf-oauth-resource-metadata/issues?q=is%3Aissue+is%3Aclosed>

Two Related Open Questions

- Should WWW-Authenticate return Resource Identifier or Resource Metadata URL?
 - RFC 8707 (Resource Indicators) uses Resource Identifier as “resource” value
 - That’s what this draft also currently does
- May WWW-Authenticate ever return a different Resource Identifier hostname than the one to which the request was made?
 - When would that be desirable?
 - When there’s difficulty hosting content at the path derived from the resource
 - What would the security implications of doing so be?
 - Currently exploring this

Next Steps

- Resolve remaining open issues
- After that, possibly ask for working group last call?

Backup Slides

Example Protected Resource Metadata Request


```
GET /.well-known/oauth-protected-resource HTTP/1.1  
Host: resource.example.com
```

Example Protected Resource Metadata Response

HTTP/1.1 200 OK

Content-Type: application/json

```
{
  "resource":
    "https://resource.example.com",
  "authorization_servers":
    ["https://as1.example.com/",
     "https://as2.example.net/"],
  "bearer_methods_supported":
    ["header", "body"],
  "resource_documentation":
    "http://resource.example.com/resource_documentation.html"
}
```



This is the metadata element that tells Clients what Authorization Server issuer URLs they can use with this Protected Resource