

UMA Resource Registration – Draft 00

Introduction

This draft defines the resource registration endpoint at the Authorization Manager side where hosts acting as OAuth clients can register resources which should be protected by this Authorization Manager. For each registered resource the host expects that the AM will be able to evaluate requests made by requesters.

Terminology

resource registration endpoint

The authorization manager's HTTP endpoint capable of receiving information identifying resource at the host.

resource discovery endpoint

The host's HTTP endpoint capable of receiving requests for resources from authorization managers and replying with a list of resources that may be registered at these AMs.

Discovery of resource registration endpoint

The host uses the [sitemeta] and [hostmeta] discovery mechanisms to learn about the URI of the resource registration endpoint at the authorization server at which the host wants to register its resources. The authorization server MUST provide a host-meta document containing:

- Link element with the following `rel` value of http://kantarainitiative.org/uma/am/resource_registration (REQUIRED)

```
<XRD>

<Host>http://server.example.com</Host>

<Link rel="http://kantarainitiative.org/uma/am/resource_registration"
href="http://server.example.com/resource_register">
<Title>Resource Registration Endpoint</Title>
</Link>

<!-- other content omitted -->

</XRD>
```

Overview

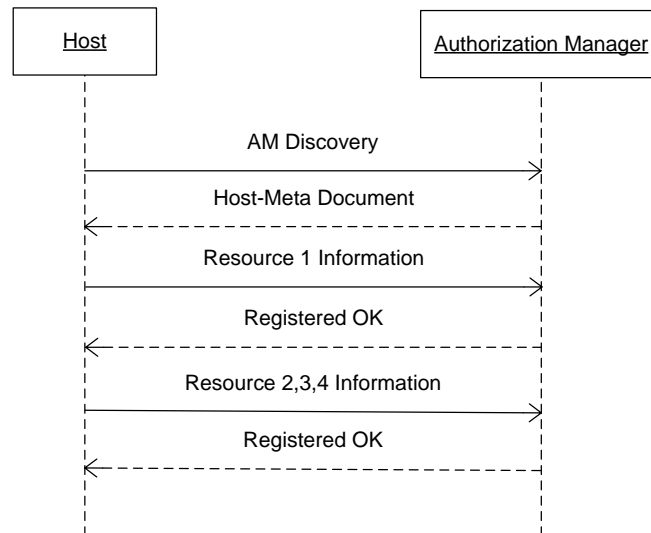
The resource registration process allows the host to register resources which should be protected by the Authorization Manager which the host has previously established a trust relationship with (i.e.

has been provisioned with the required access token for using the AM's validation endpoint and the resource registration endpoint). This specification defines two different flows for registering resources at the AM side. These flows are the Push Resource Registration flow and the Pull Resource Registration flow. This specification defines both flows.

Push Resource Registration

The Push Resource Registration flow is suitable when the Authorizing User is at the Host side and wants to register resources managed at this Host with their preferred Authorization Manager. This flow works as following:

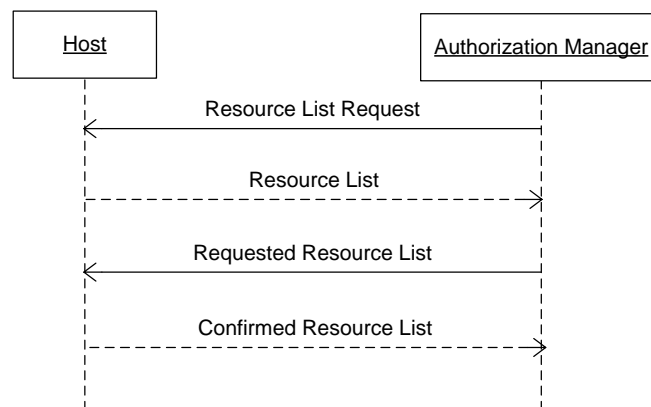
1. The host uses the resource registration endpoint to send information regarding a resource or a set of resources that a host wants to protect with the Authorization Manager.
2. The Authorization Manager confirms registration of each resource at the AM side.



Pull Resource Registration

The Pull Client Registration flow is suitable when the Authorizing User is at the AM side and wants to register resources managed at a Host (or a set of Hosts) with their preferred Authorization Manager. This flow works as following:

1. The AM sends a request to the host's resource discovery endpoint.
2. The host responds with a list of resource identifiers that this host may register with this AM.
3. The AM sends a list of resource identifiers that it wishes to protect at its side.
4. The Host responds with a list of resource identifiers that a host accepts to register at this AM.



Push Client Registration

Resource Registration Request

The Host sends a JSON formatted document to the resource registration endpoint. The host includes the following parameters in the request:

- subject (REQUIRED)
- expires (REQUIRED)
- resources (REQUIRED)

The client MAY include additional information in the request and the authorization server MAY ignore this information.

TBS: type parameter names should be different

For example, the client sends:

```

POST /register HTTP/1.1
Host: server.example.com
Content-Type: application/json
Authorization: Token token="EQ1GAWxWcKUU"

{
  "subject": "http://onlinephotogallery.com",
  "expires": "2011-01-30T09:30:00Z",
  "resources":
  [
    {
      "href": "http://onlinephotogallery.com/bob/Newcastle",
      "title": "Newcastle Photos"
    },
    {
      "href": "http://onlinephotogallery.com/bob/California",
      "title": "California Photos"
    }
  ]
}
  
```

```
]
}
```

The parameters are included in the entity body of the HTTP request using the "application/json" media type as defined by [RFC4627]. The parameters are serialized into a JSON structure by adding each parameter at the highest structure level. Parameter names and string values are included as JSON strings.

Resource Registration Response

Based on the received resource registration request, the Authorization Manager decides which resources are registered at the AM side. The AM may choose not to register all resources that a Host provides in the request. The Authorization Manager replies with a list of resources that the AM decided to register by adding the following parameters to the entity body of the HTTP response with a 200 status code (OK):

- resources_confirmed (REQUIRED)

The parameters are included in the entity body of the HTTP response using the "application/json" media type as defined by [RFC4627]. The parameters are serialized into a JSON structure by adding each parameter at the highest structure level. Parameter names and string values are included as JSON strings.

For example:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "resources_confirmed":
  [
    {
      "href":"http://onlinephotogallery.com/bob/Newcastle"
    },
    {
      "href":"http://onlinephotogallery.com/bob/California"
    }
  ]
}
```

Error Message

If the request for registration is invalid or unauthorized, the authorization server constructs the response by adding the following parameters to the entity body of the HTTP response with a 400 status code (Bad Request) using the "application/json" media type:

- error (REQUIRED)
- description (OPTIONAL)

```
HTTP/1.1 400 Bad Request
```

Content-Type: application/json

```
{  
  "error": "resources_already_registered",  
  "description": "Provided resources have been previously registered."  
}
```

TBS: We need standardized error codes.

Pull Client Registration

TBS: ...

References

[sitemeta] Defining Well-Known Uniform Resource Identifiers (URIs) (RFC5785)

<http://tools.ietf.org/html/rfc5785>

[hostmeta] Web Host Metadata

<http://tools.ietf.org/html/draft-hammer-hostmeta-13>