Quick Start with WebAC

In this quick start, you will create a sample resource and an ACL for that resource, verify that access to that resource is correctly restricted, and finally modify the ACL to allow you to update the resource.

Prerequisites

- a running Fedora repository
- curl

The commands in this guide assume that your Fedora repository root is http://localhost:8080/fcrepo/rest.

Steps

Create these two files in a local directory:

foo.ttl

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.

<> dc:title "Hello, World!".
```

acl.ttl

```
@prefix acl: <http://www.w3.org/ns/auth/acl#>.

<#authz> a acl:Authorization;
   acl:accessTo </fcrepo/rest/foo>;
   acl:agent "user1";
   acl:mode acl:Read.
```

Upload these files into the repository:

```
curl -X PUT http://localhost:8080/fcrepo/rest/foo -u admin1:password3 \
    -H "Content-Type: text/turtle" --data-binary @foo.ttl
curl -X PUT http://localhost:8080/fcrepo/rest/foo/fcr:acl -u admin1:password3 \
    -H "Content-Type: text/turtle" --data-binary @acl.ttl
```

Now `user1` is able to read the resource at `http://localhost:8080/rest/foo`, but `user2` cannot. To test this, try the following two commands:

```
curl -i http://localhost:8080/fcrepo/rest/foo -u user1:password1
curl -i http://localhost:8080/fcrepo/rest/foo -u user2:password2
```

The first request should succeed with a **200 OK** response code, and the second should fail with a **403 Forbidden**.

To demonstrate that `user1` indeed only has read-only access to `foo`, we can try updating `foo`. Create a file named **foo.sparql** with the following contents:

foo.sparql

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>

INSERT DATA { <> dc:description "Quick Start with WebAC and Fedora 4" . }
```

Then run this to attempt to update `foo`:

```
curl -i -X PATCH http://localhost:8080/fcrepo/rest/foo -u user1:password1 \
      -H "Content-Type: application/sparql-update" \
      --data-binary @foo.sparql
```

This request should fail with a **403 Forbidden** response, since `user1` has read-only access to `foo`. To add write access for `user1`, we will need to update the `acl/authz` resource as `admin`. Create a file named **authz.sparql** with the following contents:

authz.sparql

```
PREFIX acl: <http://www.w3.org/ns/auth/acl#>

INSERT DATA { <#authz> acl:mode acl:Write . }
```

Run this command to update the ACL authorization:

```
curl -i -X PATCH http://localhost:8080/fcrepo/rest/acl/authz -u admin1:password3 \
      -H "Content-Type: application/sparql-update" \
      --data-binary @authz.sparql
```

If the update to the authorization was successful, you will see a **204 No Content** response.

Now you should be able to re-run the earlier command to update the `foo` resource as `user1`:

```
curl -i -X PATCH http://localhost:8080/fcrepo/rest/foo -u user1:password1 \
      -H "Content-Type: application/sparql-update" \
      --data-binary @foo.sparql
```

Now this should return a **204 No Content** response. To verify that the update happened, you can also go to http://localhost:8080/fcrepo/rest/foo in your web browser, and confirm that it has both **dc:title** and **dc:description** properties.

ACLs for the Repository Root

⚠ When creating an ACL to protect the repository root, you **must** include a trailing slash in the Authorizations's `acl:accessTo` predicate, otherwise the Authorization will not match the request URI, and won't get applied.

Non-Working Version

```
<#rootAuthz> a acl:Authorization;
    acl:accessTo <https://localhost:8080/fcrepo/rest> .
```

Working Version

```
<#rootAuthz> a acl:Authorization;
    acl:accessTo <https://localhost:8080/fcrepo/rest/> .
    # note this trailing slash ---------------------^
```