Globbing

Reading Resources

Resources can be commonly accessed (i.e. read) using HTTP GET requests. Solid servers are encouraged to perform content negotiation for RDF resources, depending on the value of the Accept header.
IMPORTANT: a default Content-Type: text/turtle will be used for requests for RDF resources or views (containers) that do not have an Accept header.

Streams

Being LDP (BasicContainer) compliant, Solid servers MUST return a full listing of container contents when receiving requests for containers. For every resource in a container, a Solid server may include additional metadata, such as the time the resource was modified, the size of the resource, and more importantly any other RDF type specified for the resource in its metadata. You will notice in the example below that the <profile> resource has the extra RDF type<http://xmlns.com/foaf/0.1/PersonalProfileDocument>, and also that the resource <workspace/> has the RDF type<http://www.w3.org/ns/pim/space#Workspace>.
Extra metadata can also be added, describing whether each resource in the container maps to a file or a directory on the server, using the POSIX vocabulary. Here is an example that reflects how our current server implementations handle such a request:

REQUEST:

```
GET /
Host: example.org
```

RESPONSE:

```
HTTP/1.1 200 OK
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<>
    a <http://www.w3.org/ns/ldp#BasicContainer>, <http://www.w3.org/ns/ldp#Container>,
<http://www.w3.org/ns/posix/stat#Directory> ;
    <http://www.w3.org/ns/ldp#contains> <profile>, <data/>, <workspace/> ;
    <http://www.w3.org/ns/posix/stat#mtime> "1436281776" ;
    <http://www.w3.org/ns/posix/stat#size> "4096" .

<profile>
    a <http://xmlns.com/foaf/0.1/PersonalProfileDocument>, <http://www.w3.org/ns/posix/stat#File> ;
    <http://www.w3.org/ns/posix/stat#mtime> "1434583075" ;
    <http://www.w3.org/ns/posix/stat#size> "780" .
```

Globbing (inlining on GET)

We have found that in some cases, using the existing LDP features was not enough. For instance, to optimize certain applications we needed to aggregate all RDF resources from a

container and retrieve them with a single GET operation. We implemented this feature on the servers and decided to call it "globbing". Similar to UNIX shell glob, doing a GET on any URI which ends with a * will return an aggregate view of all the resources that match the indicated pattern.
For example, let's assume that /data/res1 and /data/res2 are two resources containing one triple each, which defines their type as follows:
For res1:

```
<> a <https://example.org/ns/type#One> .
```

For res2:

```
<> a <https://example.org/ns/type#Two> .
```

If one would like to fetch all resources of a container beginning with res (e.g. /data/res1, /data/res2) in one request, they could do a GET on /data/res* as follows.
REQUEST:

```
GET /data/res* HTTP/1.1
Host: example.org
```

RESPONSE:

```
HTTP/1.1 200 OK
<res1>
    a <https://example.org/ns/type#One> .

<res2>
    a <https://example.org/ns/type#Two> .
```

Alternatively, one could ask the server to inline all resources of a container, which includes the triples corresponding to the container itself:

REQUEST:

```
GET /data/* HTTP/1.1
Host: example.org
```

RESPONSE:

```
HTTP/1.1 200 OK
<>
    a <http://www.w3.org/ns/ldp#BasicContainer> ;
    <http://www.w3.org/ns/ldp#contains> <res1>, <res2> .

<res1>
    a <https://example.org/ns/type#One> .

<res2>
    a <https://example.org/ns/type#Two> .
```

Note: the aggregation process is not currently recursive, therefore it will not apply to children containers.