Multipart Extension for ALTO

draft-zhang-alto-multipart-02

J. Jensen Zhang

ALTO @ IETF 104
Mar 26, 2019
Overview of Updates for ALTO Multipart Extension

● Goal of the latest revision (-02):
  ○ Clarify the motivation to use "multipart" in ALTO
  ○ Clarify the compatibility issue and solution

● Updates since the last ALTO interim meeting (revision -01):
  ○ Made terms consistent with [RFC2045], [RFC2046], and [RFC2387].
  ○ Update protocol specification and examples to follow the latest WG drafts.
  ○ Extended design space for ALTO multipart extension.
  ○ Discussed the compatibility issue and proposed the basic solution.
Design Space: 3 Cases to Apply Multipart in ALTO

- **Case 1**: Multipart Request and Multipart Response
  - **Example**: batch query for data consistency

- **Case 2**: Single Request and Fixed-Layout Multipart Response
  - **Example**: path vector query

- **Case 3**: Single Request and Flexible-Layout Multipart Response
  - **Example**: general multiple resource query (proposed in revision -00 and -01)
Example for Case 1

```
"default-networkmap": {
  "uri": "http://alto.example.com/networkmap/default",
  "media-type": "application/alto-networkmap+json"
},
"filt-netmap": {
  "uri": "http://alto.example.com/filter/networkmap",
  "media-type": "application/alto-networkmap+json",
  "accepts": "application/alto-networkmapfilter+json",
  "uses": [ "default-networkmap" ]
},
"filt-costmap": {
  "uri": "http://alto.example.com/filter/costmap",
  "media-type": "application/alto-costmap+json",
  "accepts": "application/alto-costmapfilter+json",
  "uses": [ "default-costmap" ],
  "capabilities": ...
},
"multipart-batch": {
  "uri": "http://alto.example.com/query/batch",
  "media-type": "multipart/mixed",
  "accepts": "multipart/mixed",
  "uses": [ "filt-netmap", "filt-costmap" ]
}
```

---

Request

<table>
<thead>
<tr>
<th>Method</th>
<th>URI</th>
<th>Accepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>/query/batch</td>
<td>multipart/mixed;boundary=case1</td>
</tr>
<tr>
<td>Content-Type</td>
<td>multipart/mixed</td>
<td></td>
</tr>
<tr>
<td>Accepts</td>
<td>multipart/mixed</td>
<td></td>
</tr>
</tbody>
</table>

```
--case1
Content-Type: application/alto-networkmapfilter+json

<network-map-filter>
--case1
Content-Type: application/alto-costmapfilter+json

<cost-map-filter>
--case1--
```

Response

```
--case1
Content-Type: multipart/mixed;boundary=case1

--case1
Content-Type: application/alto-networkmap+json

<network-map-obj>
--case1
Content-Type: application/alto-costmap+json

<cost-map-obj>
--case1--
```
Example for Case 2

For Case 2, IRD MUST export the layout of the multipart response message.

```
"multipart-pv": {
    "uri": "http://alto.example.com/query/pv",
    "media-type": "multipart/related;type=application/alto-costmap+json",
    "accepts": "application/alto-costmapfilter+json",
    "capabilities": {
        "cost-type-names": [ "pv-bw" ],
        "multipart-layout": [
            { "resource-id": "pv-costmap", "media-type": "application/alto-costmap+json" },
            { "resource-id": "pv-nepmap", "media-type": "application/alto-propmap+json" }
        ]
    }
}
```

- Media type for root part
- Capabilities for root part
- Layout of the multipart response message:
  - the first part is a cost map;
  - the second part is a property map.
Compatible with Incremental Update

- [I-D.ietf-alto-incr-update-sse] allows the client to subscribe multiple resources in a batch. No need extra support to allow the Multipart resource subscription.

- For Case 2 and Case 3:
  - The client subscribe the Multipart resource.
  - The server returns the full multipart message via SSE, and automatically subscribes the whole multipart tree.
  - Once data updated, the server publishes the update for each node in the tree individually.
Next Steps

- Add concrete motivating examples to show the workflow and benefit of ALTO multipart extension
- Call for reviews
Backup Slides
Quick Updates

● The document is still in the very early stage.

● Updates since IETF 102:
  ○ Update text and terms
  ○ Clarify the requirements for the query languages and programs
  ○ Update the protocol errors
  ○ Add some security considerations