

SCIM Patch Proposal

March 6 2014
IETF 89, London
Phil Hunt

Issue

- **Simplicity**
 - Not obvious what the operation is particularly with delete
- **Complex multi-value updates**
- **Consider two options**
 - JSON Merge Patch (draft-ietf-appsawg-json-merge-patch-02)
 - JSON Patch (RFC6902)

Current Example

- Remove and Add Member

```
PATCH /Groups/acbf3ae7-8463-4692-b4fd-9b4da3f908ce
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Bearer h480djs93hd8
If-Match: W/"a330bc54f0671c9"
```

```
{
  "schemas": ["urn:scim:schemas:core:2.0:Group"],
  "members": [
    {
      "display": "Babs Jensen",
      "$ref": "https://example.com/v2/Users/2819c223-7f76-453a-919d-413861904646",
      "value": "2819c223-7f76-453a-919d-413861904646",
      "operation": "delete"
    },
    {
      "display": "James Smith",
      "$ref": "https://example.com/v2/Users/08e1d05d-121c-4561-8b96-473d93df9210",
      "value": "08e1d05d-121c-4561-8b96-473d93df9210"
    }
  ]
}
```

Current Example

- Remove All Members (remove the schema approach)

```
PATCH /Groups/acbf3ae7-8463-4692-b4fd-9b4da3f908ce
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Bearer h480djs93hd8
If-Match: W/"a330bc54f0671c9"
```

```
{
  "schemas": [ "urn:scim:schemas:core:2.0:Group" ],
  "meta": {
    "attributes": [
      "members"
    ]
  }
}
```

Possible Options

JSON Merge Patch

- Works through document comparison
 - Server infers what operations are needed
- Implied operations
 - Null incoming value implies delete
 - Attributes values provided implies replace

Merge Format

PATCH /target HTTP/1.1

Host: example.org

Content-Type: application/merge-patch+json

```
{  
  "a": "z",  
  "c": {  
    "f": null  
  }  
}
```

All Values of attr a are replaced with value "z"

sub-attribute "f" is removed

Example record

```
{
  "title": "Goodbye!",
  "author" : {
    "givenName" : "John",
    "familyName" : "Doe"
  },
  "tags": [ "example", "sample" ],
  "content": "This will be
unchanged"
}
```


Example Merge Patch

```
PATCH /my/resource HTTP/1.1
Host: example.org
Content-Type: application/merge-patch+json; charset="UTF-8"
```

```
{
  "title": "Hello!",
  "phoneNumber": "+01-123-456-7890",
  "author": {
    "familyName": null
  },
  "tags": [ "example" ]
}
```

- The resulting JSON document would be:

```
{
  "title": "Hello!",
  "author" : {
    "givenName" : "John"
  },
  "tags": [ "example" ],
  "content": "This will be unchanged",
  "phoneNumber": "+01-123-456-7890"
}
```

JSON Merge Summary

- Operations implied not specific
 - This is both good and bad!
- Complex multi-values are difficult to handle
 - Only allows for whole value replacement
 - Cannot remove attribute by sub-attribute value
- May have positive benefits in cross-domain scenarios where state is not fully known

JSON PATCH (RFC6902)

- Issue: use of 6901 JSON Pointers for PATH

```
{ "op": "add|remove|replace|copy|  
move|test", "path": "/a/b/c",  
"value": [ "foo", "bar" ] }
```

- Proposal – use SCIM PATH

```
{ "op": "add|remove|replace|test",  
"path": "<schema>:<attr>[<subattr-  
filter]", "value": [ "foo",  
"bar" ] }
```

JSON Patch and SCIM

- Multi-op
 - Each operation performed as a step
- Atomicity at REQUEST level
 - any failure must roll back all operations to pre-request state
- Test
 - allows an operation to be rolled back if data condition not met during update
- Consistent & simple JSON format for operations
- Surprisingly concise when applied to SCIM!
- Path can leverage our recent complex filter notation
 - attr[subattr eq "val"]

Proposal

- Use 6902
 - re-profile to use SCIM attributes and complex filters for path instead of 6901
 - Support for add|remove|replace|test
 - copy and move not useful since SCIM not array based
 - path becomes

```
"path" = "members"
```

```
"path" = "name.familyName"
```

```
"path" = "addresses[type eq \"work\"]"
```

```
"path" = "members[value eq  
  \"2819c223-7f76-453a-919d-413861904646\"]"
```

```
"path" = "members[value eq  
  \"2819c223-7f76-453a-919d-413861904646\"].displayName"
```

PATCH Add

```
PATCH /Groups/acbf3ae7-8463-4692-b4fd-9b4da3f908ce
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Bearer h480djs93hd8
If-Match: W/"a330bc54f0671c9"
```

```
{
  "op": "add",
  "path": "members",
  "value": [
    {
      "display": "Babs Jensen",
      "$ref": "https://example.com/v2/Users/2819c223-...
919d-413861904646",
      "value": "2819c223-7f76-453a-919d-413861904646"
    }
  ]
}
```

PATCH Remove Value

```
PATCH /Groups/acbf3ae7-8463-4692-  
b4fd-9b4da3f908ce
```

```
Host: example.com
```

```
Accept: application/json
```

```
Content-Type: application/json
```

```
Authorization: Bearer h480djs93hd8
```

```
If-Match: W/"a330bc54f0671c9"
```

```
{  
  "op": "remove",  
  "path": "members[value eq \"2819c223-7f76-...  
413861904646\"]"  
}
```

PATCH Remove Examples

- Remove all

```
{ "op": "remove", "path": "members" }
```

- Remove by query (New!)

```
{  
  "op": "remove",  
  "path": "address[type eq \"work\" and value ew \"example.com\"]"  
}
```


PATCH Multit-Op Remove & Add

```
[
  {
    "op": "remove",
    "path": "members[value eq \"2819c223-...919d-413861904646\"]"
  },
  {
    "op": "add",
    "path": "members",
    "value": [
      {
        "display": "James Smith",
        "$ref": "https://example.com/v2/Users/08e1d05d-...
8b96-473d93df9210",
        "value": "08e1d05d-121c-4561-8b96-473d93df9210"
      }
    ]
  }
]
```

PATCH Replace

```
{
  "op": "replace",
  "path": "members",
  "value": [
    {
      "display": "Babs Jensen",
      "$ref": "https://example.com/v2/Users/2819c223-...
919d-413861904646",
      "value": "2819c223-7f76-453a-919d-413861904646"
    },
    {
      "display": "James Smith",
      "$ref": "https://example.com/v2/Users/08e1d05d-121c-...
473d93df9210",
      "value": "08e1d05d-121c-4561-8b96-473d93df9210"
    }
  ]
}
```

Old Replace

```
{
  "schemas": ["urn:scim:schemas:core:2.0:User"],
  "addresses": [
    {
      "type": "work",
      "streetAddress": "100 Universal City Plaza",
      "locality": "Hollywood",
      "region": "CA",
      "postalCode": "91608",
      "country": "US",
      "formatted": "100 Universal City Plaza\nHollywood, CA 91608 US",
      "primary": true,
      "operation": "delete"
    },
    {
      "type": "work",
      "streetAddress": "911 Universal City Plaza",
      "locality": "Hollywood",
      "region": "CA",
      "postalCode": "91608",
      "country": "US",
      "formatted": "911 Universal City Plaza\nHollywood, CA 91608 US",
      "primary": true
    }
  ]
}
```

Replace Examples

- Replace work address

```
{
  "op": "replace",
  "path": "addresses[type eq \"work\"]",
  "value":
  {
    "type": "work",
    "streetAddress": "911 Universal City Plaza",
    "locality": "Hollywood",
    "region": "CA",
    "postalCode": "91608",
    "country": "US",
    "formatted": "911 Universal City Plaza\nHollywood, CA 91608 US",
    "primary": true
  }
}
```

- Update work address (NEW!)

```
{
  "op": "replace",
  "path": "addresses[type eq \"work\"].streetAddress",
  "value": "911 Universal City Plaza"
}
```

Test Operation (New!)

- Use as an attribute match condition

```
[
  {
    "op": "test",
    "path": "addresses[type eq \"work
\"] .streetAddress",
    "value": "100 Universal City Plaza"
  },
  {
    "op": "replace",
    "path": "addresses[type eq \"work
\"] .streetAddress",
    "value": "911 Universal City Plaza"
  }
]
```

JSON Patch Summary

- Multiple discrete operations
 - Actions are clear and precise
- Fully support complex attributes using sub-attribute queries
 - May cause issues for some implementations
- Requires overriding 6901 JSON Path
 - May cause issues if using generic libraries
- Not all 6902 operations can be supported
- 6902 style SCIM operations appear to be smaller / more concise (surprise)

Conclusion

- Question
 - Is profiled 6902 JSON Patch the way to go?
 - Should more exploration of JSON Merge Patch be done
 - Should we explore other proposals?