Specific RPP design challenges

IETF 122 Bangkok, Mar 2025

Pawel Kowalik, DENIC eG

Warning

The following slides contain some of typical design problems with RESTful APIs,

where people tend to have strong opinions…

Hypermedia or not hypermedia

```
POST /domains/ HTTP/1.1
Host: rpp.example.com
Content-Type: application/rpp+json

{
      "name": "example.example.com",
      ...
      "registrant": [
            "CTC42977"
      ]
      ...
}
```

Hypermedia or not hypermedia (2)

```
POST /domains/ HTTP/1.1
Host: rpp.example.com
Content-Type: application/rpp+json

{
      "name": "example.example.com",
      ...
      "registrant": [
            "/contacts/CTC42977"
      ]
      ...
}
```

Hypermedia or not hypermedia (3)

```
POST /domains/ HTTP/1.1
Host: rpp.example.com
Content-Type: application/rpp+json

{
        "name": "example.example.com",
        ...
        "links": [
                {
                        "href": "/contacts/CTC42977",
                        "rel": ”registrant”
                }
        ]
        ...
}
```

Hypermedia or not hypermedia (4)

```
POST /domains/ HTTP/1.1
Host: rpp.example.com
Content-Type: application/rpp+json

{
        "name": "example.example.com",
        ...
        "links": [
                {
                        "href": ”https://rpp.foo.example/contacts/CTC42977",
                        "rel": ”registrant”
                }
        ]
        ...
}
```

Create: POST vs. PUT dillema

- PUT is appropriate to create a resource with a known URL
  - /domains/foo.com is a known URL
  - PUT gives some guarantees of idempotency, e.g. sending twice the same representation one would still get the same registered domain (unless something happens in between – same as with EPP)
- PUT will also work for host objects
- For contacts it would depend if ID is client assigned or server assigned
    - If client assigns the ID, then PUT is still appropriate
    - If server assigns the ID, then POST on a collection would be appropriate
- Challenge: use POST everywhere for convenience, or PUT where appropriate?

Modelling ”check” command

HEAD vs. PUT/POST with Expect

- HEAD SHOULD be actually same as GET, just skipping the response body. Some frameworks make it just like that implicitly.
- it’s rather uncommon for HEAD request to have a body.
- EPP Check is aimed to tell, if a Create command would succeed.
- The equivalent would be PUT/POST with Expect HTTP Header. Still not same, as EPP Check would only require domain name, not the whole representation. Close enough anyway?
- Challenge: is Expect header well known and used?

Rich queries

- EPP Command-Response Extension type allows for large payloads in the commands, also for Info Command, which would be mapped to HTTP GET
- Challenge: HTTP GET would rather not contain body
- Possible approaches:
  - everything in query string and hope it's long enough for real life use-cases
  - HTTP QUERY
    - draft-ietf-httpbis-safe-method-w-body
  - Special resource like _query and POST
  - RPC-Style overloaded POST on collection (with different media type?)

Processing resources

In RESTful design there is only limited set of standard interface verbs. If it's not enough one can either go into RCP-style overloaded POST, or decompose the resource model to define "processing resources".

Example - Transfers:

PUT /domain/example.com/transfer (EPP op="request)

GET /domain/example.com/transfer (EPP op="query")

PUT /domain/example.com/transfer/approval (EPP op="approve")

PUT /domain/example.com/transfer/rejection (EPP op="reject")

DELETE /domain/example.com/transfer (EPP op="cancel")

Processing resources (2)

Renewal:

PUT /domain/example.com/renewal

Going futher… shall all “processes” be represented as resources, especially in “pending” state?

pendingRenew will allow for:

GET /domain/example.com/renewal

DELETE /domain/example.com/renewal

Processing resources (3)

pendingCreate

GET /domain/example.com/creation

DELETE /domain/example.com/ creation

… and so on for Update, Restore, Delete

A bonus: OPTIONS HTTP verb could then tell the client whether the process can be cancelled or not.

Authcode

Authcode is an authorization information for certain use cases:

- Reading objects of different owner
- Transferring objects

- EPP keeps it in the payload… but for HTTP GET there is no payload
  - Query string is not appropriate for any confidential information
- Header
  - Own header, like RPP-Authcode ?
  - Register Authentication Scheme Name “rpp” and have potentially 2 “Authorization” headers?

Thank you

- Contact: pawel.kowalik@denic.de
- LinkedIn: https://www.linkedin.com/in/pawelk/