JMAP for Migration and Data Portability

Experiences and ideas

JMAP as a generic (one-time) API for migration

Problem statement:

- Move data from legacy system to modern system
- Legacy system has no API
- API will be used for migration only (one-time API)
- Point third-party developers to an API specification

➔ JMAP as a generic protocol

☑ Powerful enough for use-case

☐ Ease of implementation for server developers (following slides)

Bending JMAP for rapid development

Some ideas:

- Focus on key objects and properties for migration use-case (e.g. no /query , no properties in /get request)
- Skip Session Object in case username == account
- Introduce simplified request (next slide)

Bending JMAP for rapid development (2)

Introduce simplified request (removes the need to implement JSON parsing on server):

```
$ curl <jmap-endpoint>?class=Preferences&method=get&accountId=<account-id>

$ curl <jmap-endpoint> --data '{
  "using": [
    "https://www.audriga.eu/jmap/preferences/",
    "https://www.audriga.eu/jmap/preferences-blocklist/",
    "urn:ietf:params:jmap:core"
  ],
  "methodCalls": [
    [
      "Preferences/get",
      {
        "accountId": "<account-id>"
      },
      "0"
    ]
  ]
}'
```

Discussion

- Benefits
    - Rapid development (reusing/extending existing JMAP concepts and libraries)
    - Resulting library extension might be useful for the JMAP ecosystem in general
- Related problems
    - We get many requests where legacy data cannot be accessed by an API (maildir archives, legacy file shares, folders full of VCards etc.)
- A “JMAP wrapper” seems to be a very useful approach to help make accessible such data (as we validated in practice!) - even if not complying with full JMAP specs
- Any thoughts?
    - Where to put document?
    - …