

Intro to JMAP

A spiritual successor to IMAP & Submission

Daniel Gultsch

XMPP Office Hours

April 27, 2021

Shortcomings of IMAP

- Fractured ecosystem of various extensions
- Custom wire format
- Requires persistent TCP connection
- MIME parsing
- Sending over SMTP/Submission

- Unified Spec
- Threading + FTS + Submission
- MIME parsing included
- Accessible via HTTP
- Third Party Push (APNS, FCM, ...)

- Server-to-server untouched (SMTP)
- Operates on the same datastore (Maildir, ...)
- Works alongside IMAP clients
- Inherits legacy (Keywords **and** roles)

JMAP is more

- Multipurpose sync protocol
- IETF Standard (JMAP Working Group)
- Base protocol + Email done
- Contacts, Calendars, Tasks in the works

JMAP is even more

- Core Protocol
 - Remote Procedure Calls
 - Standard Methods to sync *Things*
- Extensions
 - Define *Things*
 - Additional Methods

Remote Procedure Call Protocol

- Multiple 'Method Invocations' per request
- Method Invocation: Name, Parameters, ID

```
{  
  "using": [ "urn:example:office:hours" ],  
  "methodCalls": [  
    [ "method1", {  
      "arg1": "arg1data",  
      "arg2": {"complex": "thing"}  
    }, "call-id-01" ],  
    [ "method2", {  
      "arg1": "arg1data"  
    }, "call-id-02" ]  
  ]  
}
```

- Thing/get
 - Get **Things** by IDs
 - Returns current state (arbitrary short string)
- Thing/changes
 - IDs of **Things** created, updated, deleted since last state
- Thing/set
 - Create, update, delete **Things**
- Thing/query
 - Searches in **Things**
 - Returns IDs

■ Things

- Mailbox
- Thread
- Email
- Identity

■ Methods

- Mailbox/get standard /get method but ID parameter can be omitted to return all
- Email/query standard /query but with additional parameter collapseThreads
- ...

Question

If `Thing/changes` and `Thing/query` only return IDs do I need a round trip to retrieve **Things**?

Answer

No. Thanks to Result References multiple method invocations in the same request can be chained together.

- The value of a parameter of one method invocation can be the value(s) of the result of a previous method invocation
- 'XPath' to reference nested values ¹

¹JSON Pointer (RFC6901)

Excursus: Result of Thing/changes

```
[[ "Thing/changes", {  
  "accountId": "A1",  
  "oldState": "000001",  
  "newState": "000003",  
  "hasMoreChanges": false,  
  "created": ["e", "f"],  
  "updated": ["c"],  
  "destroyed": ["b"]  
}, "call-id-01" ]]
```

Thing/changes + Thing/get

```
[[ "Thing/changes", {  
    "accountId": "A1",  
    "sinceState": "000001"  
}, "call-id-01" ],  
[ "Thing/get", {  
    "accountId": "A1",  
    "#ids": {  
        "resultOf": "call-id-01",  
        "name": "Thing/changes",  
        "path": "/created"  
    }  
}, "call-id-02" ]]
```

```
curl -u "beta@litt.rs:***" --data-raw '{"methodCalls": [{"Mailbox/get", {}},  
↪ "0"]], "using": ["urn:ietf:params:jmap:core",  
↪ "urn:ietf:params:jmap:mail"]}' -s -H "Content-Type:application/json"  
↪ "https://litt.rs/jmap/" | jq
```

Example: Getting 20 Email threads in Inbox

```
curl -u "beta@litt.rs:***" --data-raw '{"methodCalls": [{"name": "Email/query",
↳ {"collapseThreads": true, "accountId": "beta@litt.rs",
↳ "filter": {"inMailbox": "46b9c2ce-7a7c-4736-9711-02a22491a02b"},
↳ "limit": 20, "sort": [{"isAscending": false, "property": "receivedAt"}]},
↳ "3"}, [{"name": "Email/get", "accountId": "beta@litt.rs", "#ids": {"resultOf": "3",
↳ "name": "Email/query", "path": "/ids"}, "properties": ["threadId"]},
↳ "4"}, [{"name": "Thread/get", "accountId": "beta@litt.rs", "#ids": {"resultOf": "4",
↳ "name": "Email/get", "path": "/list/*/threadId"}}, "5"}, [{"name": "Email/get",
↳ {"fetchTextBodyValues": true, "accountId": "beta@litt.rs",
↳ "#ids": {"resultOf": "5", "name": "Thread/get",
↳ "path": "/list/*/emailIds"}}, "6"]],
↳ "using": ["urn:ietf:params:jmap:core", "urn:ietf:params:jmap:mail"]}]' -s
↳ -H "Content-Type:application/json" "https://litt.rs/jmap/" | jq
```

- Powerful but complex protocol. Steep learning curve
- Stateless
- XML vs JSON: Don't handcraft either
- Protocol done. Implementations all WIP

Thank you

- Android client: [Ltt.rs](https://litt.rs)
- Library: github.com/inputmice/jmap
- Daniel Gultsch: gultsch.de, [@iNPUTmice](https://twitter.com/iNPUTmice)
- XMPP/email: daniel@gultsch.de

Can JMAP be used for IM?

- Server pushes StateChange information. Message Thing has new state 000023
- Client Messages/changes since 000022 + Message/get
- Getting a single new message always requires round trip.
- Lot's of state changes for Message Thing
- Extension that pushes **Things** directly instead of StateChange notification???