

# JMAP

IETF 100 / 16 Nov 2017

# Note Well

**Any submission to the IETF intended by the Contributor for publication as all or part of an IETF Internet-Draft or RFC and any statement made within the context of an IETF activity is considered an "IETF Contribution". Such statements include oral statements in IETF sessions, as well as written and electronic communications made at any time or place, which are addressed to:**

- **The IETF plenary session**
- **The IESG, or any member thereof on behalf of the IESG**
- **Any IETF mailing list, including the IETF list itself, any working group or design team list, or any other list functioning under IETF auspices**
- **Any IETF working group or portion thereof**
- **Any Birds of a Feather (BOF) session**
- **The IAB or any member thereof on behalf of the IAB**
- **The RFC Editor or the Internet-Drafts function**

**All IETF Contributions are subject to the rules of RFC 5378 and RFC 8179.**

**Statements made outside of an IETF session, mailing list or other function, that are clearly not intended to be input to an IETF activity, group or function, are not IETF Contributions in the context of this notice. Please consult RFC 5378 and RFC 8179 for details.**

**A participant in any IETF activity is deemed to accept all IETF rules of process, as documented in Best Current Practices RFCs and IESG Statements.**

**A participant in any IETF activity acknowledges that written, audio and video records of meetings may be made and may be available to the public.**

# Agenda

- Overview of changes since IETF99: 15 min
- JMAP Core remaining issues: 20 min
- JMAP Mail remaining issues: 40 min
- Other business / GitHub tickets: 10 min

# Patch updates

- More efficient network usage
- Clients that don't want to support patch can just send full object; server processes the same
- Allows add/remove keyword/mailbox without blatching any other recent changes if server state != client state

# Patch updates

```
[
  [
    "setMessages",
    {
      "create": {},
      "update": {
        "bf5cb26879d72a8bcef36f0e2efe979609afbef1": {
          "keywords/$Flagged": true
        }
      },
      "destroy": []
    },
    "0"
  ]
]
```

# Back references for method arguments

```
[
  [
    "getMessageList",
    {
      "filter": {
        "inbox": "46738b6c-c83d-11e7-a64b-0026b9fac3ae"
      },
      "sort": [
        "date desc"
      ],
      "collapseThreads": true,
      "position": 0,
      "limit": 30
    },
    "0"
  ],
  [
    "getMessages",
    {
      "#ids": {
        "resultOf": "0",
        "path": "/ids"
      },
      "properties": [
        "threadId"
      ]
    },
    "1"
  ],
  ...
]
```

```
""
[
  "getThreads",
  {
    "#ids": {
      "resultOf": "1",
      "path": "/list/*/threadId"
    }
  },
  "2"
],
[
  "getMessages",
  {
    "#ids": {
      "resultOf": "2",
      "path": "/list/*/messageIds"
    },
    "properties": [
      "threadId",
      "mailboxIds",
      "keywords",
      "hasAttachment",
      "from",
      "to",
      "subject",
      "date",
      "size",
      "preview"
    ]
  },
  "3"
]
]
```

# Push Support

- Add web-encryption support
- Ensure RFC8030 conformity
- Event-source: long-polling option, client-determined ping times



# Authentication

- New authentication out of scope
- It's just HTTP, so HTTP auth (probably Basic/Bearer)

# **JMAP Core**

## **Remaining Issues**

# Top-level API object and versioning/extensions

- Wrap method calls array in a top-level object
- Add “using” property to object: opt-in to specifications (like Sieve)
- Server advertises supported specs in capabilities object
- Specs define their capability name (preferably a URL to the spec...)

# Method naming

- getFoo**s**
- getFooUpdates
- getFooList
- getFooListUpdates
- setFoo**s**

# Method naming

- getFooUpdates
  - => getFoosUpdates?
  - => getUpdatesToFoos?
- getFooList
  - => getFoosList?
  - => getListOfFoos?
- getFooListUpdates
  - => getUpdatesToListOfFoos?
  - => getFoosListUpdates?

# **JMAP Mail**

## **Remaining Issues**

# Renaming "Message" object type to "Email"

- Yes, more naming bike shedding
- The issue is whether people are likely to get confused by the "id" property on the Message object being a JMAP server id and NOT the RFC5322 Message-ID
- It's called out explicitly in the spec, but still...

# Message Format

- JSON representation of RFC5322
- What does the client want?
- What is an attachment?
- Single or multiple parts to body?
- Flat or tree?
- Partial body fetch for efficiency; how?



# ACLs and separating isUnread from keywords

- IMAP has a different ACL for \$Seen vs other keywords
- \$Seen is already special because  
Mailboxes#unreadThreads/unreadMessages
- Maybe map IMAP \$Seen to separate isUnread property?

# Destroying mailboxes: what happens to messages

- Mostly consensus reached on the list:
- `setMailboxes#onLastReferenceMoveTo: `String|null``
- Concerns raised that omitting is equivalent to null => deletes messages.
- Omitting should raise error instead?
- Is this inconsistent though?

**Any other business**