Getting Started with JMAP
the JSON Meta Application Protocol
okay, funny story
I pitched this talk:

Getting Started with JMAP

the JSON Meta Application Protocol
Getting Started with JMAP

- what is JMAP?
- how does it work?
- why is it so great?
- how do I get started
- what comes next?
- how does Fm use it?
I made that talk!
I made that talk!

- pretty good
I made that talk!

- pretty good
- comprehensive
I made that talk!

- pretty good
- comprehensive
- nearly an hour long
I made that talk!

- pretty good
- comprehensive
- nearly an hour long
- then I remembered...
I made that talk!

- pretty good
- comprehensive
- nearly an hour long
- then I remembered...

15m slot! PDF only, no Keynote
so, I present...
if you want to hear the whole thing, *it can be arranged*
if you want to hear the whole thing, it can be arranged
Fastmail
making email better

Ricardo Signes
rjbs

e-mail, calendars, and contacts

in Melbourne, AU

and Philadelphia, US

we really love email
but first, IMAP
(connection established)
S  * OK IMAP4rev1 Service Ready
C  a001 login mrc secret
S  a001 OK LOGIN completed
C  a002 select inbox
S  * 18 EXISTS
S  * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S  * 2 RECENT
S  * OK [UNSEEN 17] Message 17 is the first unseen message
S  * OK [UIDVALIDITY 3857529045] UIDs valid
S  a002 OK [READ-WRITE] SELECT completed
(connection established)

S   * OK IMAP4rev1 Service Ready
C   a001 login mrc secret
S   a001 OK LOGIN completed
C   a002 select inbox
S   * 18 EXISTS
S   * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S   * 2 RECENT
S   * OK [UNSEEN 17] Message 17 is the first unseen message
S   * OK [UIDVALIDITY 3857529045] UIDs valid
S   a002 OK [READ-WRITE] SELECT completed
(to say nothing of the payload)
whatever you want to say about HTTP+JSON

at least it isn't *that*
You probably all know how to use HTTP
You probably all know how to use HTTP even if you don't know how it works.
You probably all know how to use HTTP even if you don't know how it works and you probably do know how it works.
omitted here:
lots of slides about how weird IMAP is
Date: Wed, 17 Jul 1996 02:23:25 -0700 (PDT)
From: Terry Gray <gray@cac.washington.edu>
Subject: IMAP4rev1 WG mtg summary and minutes
To: imap@cac.washington.edu
cc: minutes@CNRI.Reston.VA.US, John Klensin <KLENSIN@MIT.EDU>
Message-Id: <B27397-0100000@cac.washington.edu>
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; CHARSET=US-ASCII

)
Date: Wed, 17 Jul 1996 02:23:25 -0700 (PDT)
From: Terry Gray <gray@cac.washington.edu>
Subject: IMAP4rev1 WG mtg summary and minutes
To: imap@cac.washington.edu
cc: minutes@CNRI.Reston.VA.US, John Klensin <KLENSIN@MIT.EDU>
Message-Id: <B27397-0100000@cac.washington.edu>
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; CHARSET=US-ASCII

)

a004 OK FETCH completed
C a005 store 12 +flags \deleted
S * 12 FETCH (FLAGS (\Seen \Deleted))
a005 OK +FLAGS completed
Date: Wed, 17 Jul 1996 02:23:25 -0700 (PDT)
From: Terry Gray <gray@cac.washington.edu>
Subject: IMAP4rev1 WG mtg summary and minutes
To: imap@cac.washington.edu
cc: minutes@CNRI.Reston.VA.US, John Klensin <KLENSIN@MIT.EDU>
Message-Id: <B27397-0100000@cac.washington.edu>
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; CHARSET=US-ASCII

)})
a004 OK FETCH completed
C a005 store 12 +flags \deleted
S * 12 FETCH (FLAGS (\Seen \Deleted))
a005 OK +FLAGS completed

sent store, got fetch!
IMAP is a cache management protocol
imap
server

client
(cache)

- fetch
- update
- create
- delete

update cache
invalidate cache
<table>
<thead>
<tr>
<th>client means</th>
<th>client sends</th>
<th>server replies</th>
<th>server means</th>
</tr>
</thead>
<tbody>
<tr>
<td>imap server</td>
<td>fetch</td>
<td>update</td>
<td>create</td>
</tr>
<tr>
<td></td>
<td>update cache</td>
<td>invalidate cache</td>
<td>delete</td>
</tr>
</tbody>
</table>
I want to work with the inbox.
Client means

<table>
<thead>
<tr>
<th>I want to work with the inbox.</th>
<th>SELECT INBOX</th>
<th>Server replies</th>
<th>Server means</th>
</tr>
</thead>
</table>

Client sends

- update cache
- invalidate cache

Server replies

- fetch
- update
- create
- delete
I want to work with the inbox.

**client means**

**client sends**

**server replies**

**server means**

---

- SELECT INBOX
- 172 EXISTS FLAGS (…)
- update cache
- invalidate cache
<table>
<thead>
<tr>
<th>client means</th>
<th>client sends</th>
<th>server replies</th>
<th>server means</th>
</tr>
</thead>
<tbody>
<tr>
<td>I want to work with the inbox.</td>
<td>SELECT INBOX</td>
<td>1 72 EXISTS FLAGS (...)</td>
<td>Prep a cache. Here's how.</td>
</tr>
<tr>
<td>client means</td>
<td>client sends</td>
<td>server replies</td>
<td>server means</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------</td>
<td>-------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>I want to work with the inbox.</td>
<td>SELECT INBOX</td>
<td>172 EXISTS FLAGS (...)</td>
<td>Prep a cache. Here's how.</td>
</tr>
<tr>
<td>I want to see new mail.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>client means</td>
<td>client sends</td>
<td>server replies</td>
<td>server means</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>I want to work with the inbox.</td>
<td>SELECT INBOX</td>
<td>172 EXISTS FLAGS (...)</td>
<td>Prep a cache. Here's how.</td>
</tr>
<tr>
<td>I want to see new mail.</td>
<td>FETCH 12:* ENVELOPE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>client means</td>
<td>client sends</td>
<td>server replies</td>
<td>server means</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>------------------------</td>
<td>-----------------------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>I want to work with the inbox.</td>
<td>SELECT INBOX</td>
<td>172 EXISTS FLAGS (...)</td>
<td>Prep a cache. Here's how.</td>
</tr>
<tr>
<td>I want to see new mail.</td>
<td>FETCH 12:* ENVELOPE</td>
<td>* 13 FETCH ...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>* 14 FETCH ...</td>
<td></td>
</tr>
<tr>
<td>client means</td>
<td>client sends</td>
<td>server replies</td>
<td>server means</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------------------</td>
<td>----------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>I want to work with the inbox.</td>
<td>SELECT INBOX</td>
<td>172 EXISTS FLAGS (...)</td>
<td>Prep a cache. Here's how.</td>
</tr>
<tr>
<td>I want to see new mail.</td>
<td>FETCH 12:* ENVELOPE</td>
<td>* 13 FETCH ...</td>
<td>Update your cache.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* 14 FETCH ...</td>
<td></td>
</tr>
</tbody>
</table>
### Client Means

<table>
<thead>
<tr>
<th>Client Means</th>
<th>Client Sends</th>
<th>Server Replies</th>
<th>Server Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>I want to work with the inbox.</td>
<td>SELECT INBOX</td>
<td>172 EXISTS FLAGS (…)</td>
<td>Prep a cache.  Here's how.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* 13 FETCH ...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>* 14 FETCH ...</td>
<td></td>
</tr>
<tr>
<td>I want to see new mail.</td>
<td>FETCH 12:* ENVELOPE</td>
<td>* 13 FETCH ...</td>
<td>Update your cache.</td>
</tr>
<tr>
<td>I want to mark this read.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>client means</td>
<td>client sends</td>
<td>server replies</td>
<td>server means</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------</td>
<td>------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>I want to work with the inbox.</td>
<td>SELECT</td>
<td>172 EXISTS FLAGS (...)</td>
<td>Prep a cache. Here's how.</td>
</tr>
<tr>
<td></td>
<td>INBOX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I want to see new mail.</td>
<td>FETCH 12:*</td>
<td>* 13 FETCH …</td>
<td>Update your cache.</td>
</tr>
<tr>
<td></td>
<td>ENVELOPE</td>
<td>* 14 FETCH …</td>
<td></td>
</tr>
<tr>
<td>I want to mark this read.</td>
<td>STORE 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>+FLAGS \Seen</td>
<td></td>
<td></td>
</tr>
<tr>
<td>client means</td>
<td>client sends</td>
<td>server replies</td>
<td>server means</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>--------------</td>
<td>--------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>I want to work with the inbox.</td>
<td>SELECT INBOX</td>
<td>172 EXISTS FLAGS (…)</td>
<td>Prep a cache. Here's how.</td>
</tr>
<tr>
<td>I want to see new mail.</td>
<td>FETCH 12:*</td>
<td>* 13 FETCH …</td>
<td>Update your cache.</td>
</tr>
<tr>
<td></td>
<td>ENVELOPE</td>
<td>* 14 FETCH …</td>
<td></td>
</tr>
<tr>
<td>I want to mark this read.</td>
<td>STORE 12</td>
<td>* 12 FETCH …</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+FLAGS \Seen</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Client Means vs. Server Means

<table>
<thead>
<tr>
<th>Client Means</th>
<th>Client Sends</th>
<th>Server Replies</th>
<th>Server Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>I want to work with the inbox.</td>
<td>SELECT INBOX</td>
<td>172 EXISTS FLAGS (...)</td>
<td>Prep a cache. Here's how.</td>
</tr>
<tr>
<td>I want to see new mail.</td>
<td>FETCH 12:* ENVELOPE * 13 FETCH … * 14 FETCH …</td>
<td></td>
<td>Update your cache.</td>
</tr>
<tr>
<td>I want to mark this read.</td>
<td>STORE 12 +FLAGS \Seen * 12 FETCH …</td>
<td></td>
<td>Update your cache.</td>
</tr>
</tbody>
</table>
C   A03 SELECT INBOX (QRESYNC (67890007 123 41,43:211,214:541))
S   * OK [CLOSED]
S   * 314 EXISTS
S   * 15 RECENT
S   * OK [UIDVALIDITY 67890007] UIDVALIDITY
S   * OK [UIDNEXT 567] Predicted next UID
S   * OK [HIGHESTMODSEQ 130]
S   * OK [UNSEEN 7] There are some unseen messages in the mailbox
S   * FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
S   * OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen \*)]
S   * VANISHED (EARLIER) 41,43:116,118,120:211,214:540
S   * 49 FETCH (UID 117 FLAGS (\Seen \Answered) MODSEQ (128))
S   * 50 FETCH (UID 119 FLAGS (\Draft $MDNSent) MODSEQ (129))
S   ... 
S   * 100 FETCH (UID 541 FLAGS (\Seen $Forwarded) MODSEQ (130))
S   A03 OK [READ-WRITE] mailbox selected
C   A03 SELECT INBOX (QRESYNC (67890007 123 41,43:211,214:541))
S   * OK [CLOSED]
S   * 314 EXISTS
S   * 15 RECENT
S   * OK [UIDVALIDITY 67890007] UIDVALIDITY
S   * OK [UIDNEXT 567] Predicted next UID
S   * OK [HIGHESTMODSEQ 130]
S   * OK [UNSEEN 7] There are some unseen messages in the mailbox
S   * FLAGS (Answered Flagged Draft Deleted Seen)
S   * OK [PERMANENTFLAGS (Answered Flagged Draft Deleted Seen *)]
S   * VANISHED (EARLIER) 41,43:116,118,120:211,214:540
S   * 49 FETCH (UID 117 FLAGS (Seen Answered) MODSEQ (128))
S   * 50 FETCH (UID 119 FLAGS (Draft $MDNSent) MODSEQ (129))
S   ...
S   * 100 FETCH (UID 541 FLAGS (Seen $Forwarded) MODSEQ (130))
S   A03 OK [READ-WRITE] mailbox selected
A03 SELECT INBOX (QRESYNC (67890007 123 41,43:211,214:541))

* OK [CLOSED]
* 314 EXISTS
* 15 RECENT
* OK [UIDVALIDITY 67890007] UIDVALIDITY
* OK [UIDNEXT 567] Predicted next UID
* OK [HIGHESTMODSEQ 130]
* OK [UNSEEN 7] There are some unseen messages in the mailbox
* FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
* OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen \*)]
* VANISHED (EARLIER) 41,43:116,118,120:211,214:540
* 49 FETCH (UID 117 FLAGS (\Seen \Answered) MODSEQ (128))
* 50 FETCH (UID 119 FLAGS (\Draft $MDNSent) MODSEQ (129))
...
* 100 FETCH (UID 541 FLAGS (\Seen $Forwarded) MODSEQ (130))
A03 OK [READ-WRITE] mailbox selected
C A03 SELECT INBOX (QRESYNC (67890007 123 41,43:211,214:541))
S * OK [CLOSED]
S * 314 EXISTS
S * 15 RECENT
S * OK [UIDVALIDITY 67890007] UIDVALIDITY
S * OK [UIDNEXT 567] Predicted next UID
S * OK [HIGHESTMODSEQ 130] Predicted highest MODSEQ
S * OK [UNSEEN 7] There are some unseen messages in the mailbox
S * FLAGS (Answered Flagged Draft Deleted Seen)
S * OK [PERMANENTFLAGS (Answered Flagged Draft Deleted Seen *)]
S * VANISHED (EARLIER) 41,43:116,118,120:211,214:540
S * 49 FETCH (UID 117 FLAGS (Seen Answered) MODSEQ (128))
S * 50 FETCH (UID 119 FLAGS (Draft MDNSent) MODSEQ (129))
S ...
S * 100 FETCH (UID 541 FLAGS (Seen Forwarded) MODSEQ (130))
S A03 OK [READ-WRITE] mailbox selected
C A03 SELECT INBOX (QRESYNC (67890007 123 41,43:211,214:541))
S * OK [CLOSED]
S * 314 EXISTS
S * 15 RECENT
S * OK [UIDVALIDITY 67890007] UIDVALIDITY
S * OK [UIDNEXT 567] Predicted next UID
S * OK [HIGHESTMODSEQ 130] Predicted next UID
S * OK [UNSEEN 7] There are some unseen messages in the mailbox
S * FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
S * OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen \*)]
S * VANISHED (EARLIER) 41,43:116,118,120:211,214:540
S * 49 FETCH (UID 117 FLAGS (\Seen \Answered) MODSEQ (128))
S * 50 FETCH (UID 119 FLAGS (\Draft $MDNSent) MODSEQ (129))
S ...
S * 100 FETCH (UID 541 FLAGS (\Seen $Forwarded) MODSEQ (130))
S A03 OK [READ-WRITE] mailbox selected
C A03 SELECT INBOX (QRESYNC (67890007 123 41,43:211,214:541))
S * OK [CLOSED]
S * 314 EXISTS
S * 15 RECENT
S * OK [UIDVALIDITY 67890007] UIDVALIDITY
S * OK [UIDNEXT 567] Predicted next UID
S * OK [HIGHESTMODSEQ 130] Predicted next UID
S * OK [UNSEEN 7] There are some unseen messages in the mailbox
S * FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
S * OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen \*)]
S * VANISHED (EARLIER) 41,43:116,118,120:211,214:540
S * 49 FETCH (UID 117 FLAGS (\Seen \Answered) MODSEQ (128))
S * 50 FETCH (UID 119 FLAGS (\Draft $MDNSent) MODSEQ (129))
S ...
S * 100 FETCH (UID 541 FLAGS (\Seen $Forwarded) MODSEQ (130))
S A03 OK [READ-WRITE] mailbox selected

C Give me updates to inbox since 123.
S Here are many updates. Apply these and you will be at 130.
Now you understand IMAP!
Now you understand IMAP!

So, who wants to implement it?
Right!

The good stuff is good, but the bad stuff is a total buzzkill.
👍 Good

can re-sync

👎 Bad
<table>
<thead>
<tr>
<th>Good</th>
<th>Bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>can re-sync</td>
<td></td>
</tr>
<tr>
<td>domain-specific model</td>
<td></td>
</tr>
</tbody>
</table>
Good
- can re-sync
- domain-specific model

Bad
- horrible data format
👍 Good

can re-sync
domain-specific model

👎 Bad

horrible data format
weird-o transport layer
Good

can re-sync
domain-specific model

Bad

horrible data format
weird-o transport layer
no (or awful) commodity code
**Good**

- can re-sync
- domain-specific model

**Bad**

- horrible data format
- weird-o transport layer
- no (or awful) commodity code
- key features not in core
Good

- can re-sync
- domain-specific model

Bad

- horrible data format
- weird-o transport layer
- no (or awful) commodity code
- key features not in core
- too many (s
JMAP
JSON Meta Application Protocol
POST /jmap HTTP/1.1
Host: api.fastmail.com
Content-Type: application/json

{
  ...
  "Email/get",
  {
    "ids": [ "1", "2", "3" ]
  },
  ...
}

request
Content-Type: application/json

{
    ...
    "Email/get",
    {
        ...
        "list": [
            {
                ...
                "id": "1",
                "subject": "Hi, Brussels! 🇧🇪💙",
            }
        ],
    },...
}
{  "id": "Mf713f8ce8838c8fb1176ad98",
    "messageId": [ "GBSALNHAGWNZBAVAWIFONB@signes.online" ],
    "blobId": "Gf713f8ce8838c8fb1176ad98e8c7adfc383b1eb2",
    "sentAt": "2020-09-23T21:50:06-08:00",
    "bodyValues": {  
        "1": {  
            "value": "Hi Rik, it's your mother...\n"
        }
    },
    "bodyStructure": {  
        "type": "multipart/alternative"
    },
    "header:to:asAddresses": [  
        {  
            "email": "rjbs@rjbs.cloud",
            "name": "Ricardo Signes"
        }
    ],
    "replyTo": [  
        {  
            "email": "xwcznsl@domainsmadeeasy.com",
            "name": "DOMAIN REGISTRAR"
        }
    ]
}
{
  "id": "Mf713f8ce8838c8fb1176ad98",
  "messageId": [ "GBSALNHAGWNZBAVAWIFONB@signes.online" ],
  "blobId": "Gf713f8ce8838c8fb1176ad98e8c7adfc383b1eb2",
  "sentAt": "2020-09-23T21:50:06-08:00",
  "bodyValues": {
    "1": {
      "value": "Hi Rik, it's your mother...
    }
  },
  "bodyStructure": {
    "type": "multipart/alternative"
  },
  "header:to:asAddresses": [
    {
      "email": "rjbs@rjbs.cloud",
      "name": "Ricardo Signes"
    }
  ],
  "replyTo": [
    {
      "email": "xwcznsl@domainsmadeeasy.com",
      "name": "DOMAIN REGISTRAR"
    }
  ]
}
You could stop here.
You could stop here.

Let's keep going!
response

{
  ...
  "Email/get",
  {
    ...
    "state": "616",
    "list": [
      {
        ...
        "id": "1",
        "subject": "Hi, Brussels! 🇧🇪💙",
      }
    ],
  }, ...
}

request

```json
{
  ...,
  "Email/changes",
  {
    ...,
    "sinceState": "616",
  }, ...
}
```
request

{ 
  ... 
  "Email/changes", 
  { 
    ... 
    "sinceState": "616", 
  }, ... 
}

response

{ 
  ... 
  "Email/changes", 
  { 
    ... 
    "oldState": "616", 
    "newState": "717", 
    "created": [ "a", "b" ], 
    "updated": [ "c" ], 
    "destroyed": [] 
  }, ... 
}
{ ...
    "Email/query",
    {
        ...
        "filter": [
            {
                ...
                "hasKeyword": "${flagged}",
                "from": "rjbs",
            }
        ],
    }
}
response

```json
{
  "Email/query",
  {
    "ids": [ "4", "8", "15", "16" ]
  }, ...
}
```
response

```json
{ ...
  "Email/query",
  { ...
    "ids": [ "4", "8", "15", "16" ]
  }, ...
}
```

request

```json
{ ...
  "Email/get",
  { ...
    "ids": [ "4", "8", "15", "16" ]
  }, ...
}
```
Brace yourself, we need to look at more IMAP.
C  a001 SEARCH FLAGGED FROM "rjbs"
S  * SEARCH 2 84 882
S  a001 OK SEARCH completed
C  a002 FETCH 2,84,882 FULL
S  * (lots of stuff)
C a001 SEARCH FLAGGED FROM "rjbs"
[ waiting ]
S * SEARCH 2 84 882
S a001 OK SEARCH completed
C a002 FETCH 2,84,882 FULL
[ waiting ]
S * (lots of stuff)
response

```json
{
  ...,
  "Email/query",
  {
    ...,
    "ids": [ "4", "8", "15", "16" ]
  }, ...
}
```

request

```json
{
  ...,
  "Email/get",
  {
    ...,
    "ids": [ "4", "8", "15", "16" ]
  }, ...
}
```
Request 1: Email/query (filter)
Response 1: Email/query (1,2,3)
Request 2: Email/get (1,2,3)
Response 2: Email/get (1,2,3)
Request 1: Email/query (filter)  
[ waiting ]
Response 1: Email/query (1,2,3)
Request 2: Email/get (1,2,3)  
[ waiting ]
Response 2: Email/get (1,2,3)
request

```json
{
  ...
  [
    "Email/query",
    {
      "filter": [...], ...
    },
    "a",
  ],
  [
    "Email/get",
    {
      "#ids": {
        "resultOf": "a",
        "name": "Email/query",
        "path": "/ids"
      },
      "b",
    },
  ],
}
```
Request

```json
{
  ...
  [ "Email/query",
    { "filter": [...], ... },
    "a",
  ],
  [ "Email/get",
    { "#ids": { "resultOf": "a", "name": "Email/query", "path": "/ids" },
      "b",
    ],
  }
}
```
Request

```json
{
  ...
  [
    "Email/query",
    {
      "filter": [...], ... },
    "a",
  ],
  [
    "Email/get",
    {
      "#ids": {
        "resultOf": "a",
        "name": "Email/query",
        "path": "/ids" }
    },
    "b",
  ],
}
```
Request 1: a: Email/query (filter)
b: Email/get (&a)
[ waiting ]

Response 1: a: Email/query (1,2,3)
b: Email/get (1,2,3)
just a couple more things
{ "using": [ "urn:ietf:params:jmap:core", "urn:ietf:params:jmap:mail" ], "methodCalls": [ [ "Mailbox/query", { "filter": { "role": "inbox" }, "accountId": "rjbs@fastmailteam.com" }, "r1" ], [ "Mailbox/get", { "accountId": "rjbs@fastmailteam.com", "properties": [ "unreadThreads" ], "#ids": { "name": "Mailbox/query", "resultOf": "r1", "path": "/ids" } }, "r2" ] ] }
```json
{
    "using": [
        "urn:ietf:params:jmap:core",
        "urn:ietf:params:jmap:mail"
    ],
    "methodCalls": [
        [
            "Mailbox/query",
            {
                "filter": {
                    "role": "inbox"
                },
                "accountId": "rjbs@fastmailteam.com"
            },
            "r1"
        ],
        [
            "Mailbox/get",
            {
                "accountId": "rjbs@fastmailteam.com",
                "properties": ["unreadThreads"],
                "#ids": {
                    "name": "Mailbox/query", "resultOf": "r1", "path": "/ids"
                }
            },
            "r2"
        ]
    ]
}
```
```json
{
    "using": [
        "urn:ietf:params:jmap:core",
        "urn:ietf:params:jmap:mail",
    ],
    "methodCalls": [
        [
            "Mailbox/query",
            {
                "filter": {
                    "role": "inbox"
                },
                "accountId": "rjbs@fastmailteam.com"
            },
            "r1"
        ],
        [
            "Mailbox/get",
            {
                "accountId": "rjbs@fastmailteam.com",
                "properties": ["unreadThreads"],
                "#ids": {
                    "name": "Mailbox/query", 
                    "resultOf": "r1",
                    "path": "/ids"
                }
            },
            "r2"
        ]
    ]
}
```
"using": [
  "urn:ietf:params:json:core",
  "urn:ietf:params:json:mail",
]

"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars"
]
{
  "using": [ ... ],
  "methodCalls": [
    [ "Mailbox/changes", ... ],
    [ "Mailbox/get", ... ],
    [ "Email/changes", ... ],
    [ "Email/get", ... ],
    [ "Contact/changes", ... ],
    [ "Contact/get", ... ],
    [ "CalendarEvent/changes", ... ],
    [ "CalendarEvent/get", ... ]
  ]
}

"using": ["urn:ietf:params:jmap:core",
"urn:ietf:params:jmap:mail",
"https://cyrusimap.org/ns/jmap/contacts",
"https://cyrusimap.org/ns/jmap/calendars",
"https://marcolini.com/jmap/chocolat"]
"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars",
  "https://marcolini.com/jmap/chocolat"
]
"using": ["urn:ietf:params:jmap:core",
"urn:ietf:params:jmap:mail",
"https://cyrusimap.org/ns/jmap/contacts",
"https://cyrusimap.org/ns/jmap/calendars",
"https://marcolini.com/jmap/chocolat"]
"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars",
  "https://marcolini.com/jmap/chocolat"
]
"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars",
  "https://marcolini.com/jmap/chocolat"
]
"using": [ "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars",
  "https://marcolini.com/jmap/chocolat"
]
"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars",
  "https://marcolini.com/jmap/chocolat"
]
"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars",
  "https://marcolini.com/jmap/chocolat"
]
oh and also
GET /jmap/events HTTP/1.1
GET /jmap/events HTTP/1.1

event: state
data: {
"Email": "818"
}

event: state
data: {
"Email": "820", "Contact": "412"
}
GET /jmap/events HTTP/1.1

event: state
data:{"Email":"818"}

event: state
data:{"Email":"820","Contact":"412"}

event: state
data:{"Email":"833","News":"69012"}
also, RFC 8030
aka Web Push
👍 Good

can re-sync
domain-specific model

👎 Bad

horrible data format
weird-o transport layer
no (or awful) commodity code
key features not in core
too many (s
Good
- can re-sync
- domain-specific model
- ubiquitous data format
- standard transport layer
- avoids round trips
- one protocol, many jobs
- real-time sync!

Bad
- horrible data format
- weird-o transport layer
- no (or awful) commodity code
- key features not in core
- too many (s
Good

- can re-sync
- domain-specific model
- ubiquitous data format
- standard transport layer
- avoids round trips
- one protocol, many jobs
- real-time sync!

Bad

- horrible data format
- weird-o transport layer
- no (or awful) commodity code
- key features not in core
- too many (s
- not much adoption yet
Good

- can re-sync
- domain-specific model
- ubiquitous data format
- standard transport layer
- avoids round trips
- one protocol, many jobs
- real-time sync!

Bad

- horrible data format
- weird-o transport layer
- no (or awful) commodity code
- key features not in core
- too many `{s
- not much adoption yet
- too many `s and "s
okay, what now?
first, read some simple code, which you can't run, just to get a look at it:

https://github.com/fastmail/JMAP-Samples
...then...
TIME TO READ SOME RFCs!
wait, it's gonna be okay!
how to enjoy the specs
how to enjoy the specs

• first, get a basic sense of how the core methods in RFC 8620 work
how to enjoy the specs

• first, get a basic sense of how the core methods in RFC 8620 work
• that is: how to /get, how to /set, how to /query
how to enjoy the specs

• first, get a basic sense of how the core methods in RFC 8620 work

• that is: how to /get, how to /set, how to /query

• then learn the specific properties and quirks of specific data types: Mailbox, Email, &c.
RFC 8620

- request basics
- five value types
- */get
- */set
- */changes
- */query
- */queryChanges
- blobs
- push subscriptions
- event source

...but no syncable data types...
RFC 8621

- Mailbox/*
- Thread/*
- Email/*
- SearchSnippet/*
- Identity/*
- EmailSubmission/*
- VacationResponse/*
some RFC highlights
this is not an elaborate prank
Email/*

the "only" complicated part
[ "Email/get",
 { "ids": [ "e1", "e2", "e3" ],
 "properties": [ "from", "to", "subject",
 "preview",
 "mailboxIds",
 ]
 },
 "a",
 ]
["Email/get",
{
  "list": [
    {
      "id": "e1",
      "to": [
        {
          "name": "Rik",
          "email": "rik@sample.com"
        }
      ],
      "from": [
        {
          "name": "Neil",
          "email": "neil@sample.com"
        }
      ],
      "subject": "Enjoy Belgium!",
      "preview": "Don't forget to ...
    }
  ],
  "mailboxIds": {
    "m1": true
  }
}]}
["Email/get",
{
  "list": [  
  {  "id": "e1",
   "to": [ {  "name": "Rik",  "email": "rik@example.com"  
   "from": [ {  "name": "Neil",  "email": "neil@example.com"  
   "subject": "Enjoy Belgium!",  "preview": "Don't forget to ...  
   "mailboxIds": {  "m1": true  
  },  
   
  }  
  ]
  },  
  "a",
  ]
},
]
response

[ "Email/get",
{
  "list": [
    {
      "id": "e1",
      "to": [
        {
          "name": "Rik",
          "email": ...
        },
        {
          "name": "Neil",
          ...
        }
      ],
      "subject": "Enjoy Belgium!",
      "preview": "Don't forget to ...
      "mailboxIds": {
        "m1": true
      },
    }
  ]
},
"a",
]
["Email/get",
{
"list": [
{
"id": "e1",
"to": [{ "name": "Rik", "email": ... }, { "name": "Neil", ... },
"from": [{ "name": "Neil", ... }, { "name": "Neil", ... }],
"subject": "Enjoy Belgium!",
"preview": "Don't forget to ...",
"mailboxIds": { "m1": true },
}
]
},
"a",
]
response

[ "Email/get",
{
   "list": [
   {
      "id": "e1",
      "to": [ { "name": "Rik", "em..." }, ...
      "from": [ { "name": "Neil", ...
      "subject": "Enjoy Belgium!",
      "preview": "Don't forget to ...
      "mailboxIds": { "m1": true },
   }, ...
   ]
   }, ...
   "a",
} ]
[ "Email/get",
{
   "list": [
      {
         "id": "e1",
         "mailboxIds": {
            "m1": true,
            "m2": true
         }
      }
   ]
},
"a",
]
[ "Email/get",
{ "list": [
{ "id": "e1",
...

"mailboxIds": {
  "m1": true,
  "m2": true
}
},
]
}]

Ask Rik about "labels mode" later!
Email headers
RFC 8621 § 4.1.2

"properties": [
  "subject",
  "header:subject",
]

request

response
Email headers
RFC 8621 § 4.1.2

"properties": [
  "subject",
  "header:subject",
]

{ "id": "eX",
  "subject": "🇧🇪🧇",
  "header:subject": " =?UTF-8?…
}

request

response
Email headers
RFC 8621 § 4.1.2
Email headers
RFC 8621 § 4.1.2

"properties": [ ]
Email headers
RFC 8621 § 4.1.2

"properties": [
  "subject",
]
Email headers
RFC 8621 § 4.1.2

"properties": [
    "subject",
    "header:subject",
]
Email headers
RFC 8621 § 4.1.2

"properties": [
  "subject",
  "header:subject",
  "header:subject:all"
]
Email headers
RFC 8621 § 4.1.2

"properties": [
  "subject",
  "header:subject",
  "header:subject:all",
  "header:subject:asText"
]
Email headers
RFC 8621 § 4.1.2

"properties": [
  "subject",
  "header:subject",
  "header:subject:all",
  "header:subject:asText",
  "header:subject:asText:all"
]
Email headers
RFC 8621 § 4.1.2

"properties": [
    "subject",
    "header:subject",
    "header:subject:all",
    "header:subject:asText",
    "header:subject:asText:all"
]
"Email/get", { ...,
    "properties": [
        "blobId",
        "textBody",
        "htmlBody",
        "bodyValues"
    ],
},
...

RF RFC 8621 § 4.1.4
Email body
RFC 8621 § 4.1.4

"Email/get", { ...
  "properties": [
    "blobId",
    "textBody",
    "htmlBody",
    "bodyValues"
  ],
...

"Email/get", {
  ...
  "properties": [
    "blobId",
    "textBody",
    "htmlBody",
    "bodyValues"
  ],
},
...

This way lies MIME parsing!!
"Email/get", { ...,
    "properties": [
        "blobId",
        "textBody",
        "htmlBody",
        "bodyValues"
    ],
    "fetchTextBodyValues": true,
    "fetchHtmlBodyValues": true,
    ...
}
It's me, hi, I'm the body,...
JMAP makes the server be smart so the client can be stupid.
I know, you're sold!

How can you JMAP?
Lots of ways!
Lots of ways!

Ask for more later!
Fastest is to use JMAP on your **Fastmail** account.

https://fastmail.com/for-developers/
Learn more about how in our JMAP howto.

jmap.topicbox.com

way cool group email written using JMAP
Thank you!