

JMAP filenode/blobext

Bron Gondwana <brong@fastmailteam.com>

IETF125 - March 2026

FileNode

- Has been idle for a while
- Recently started implementing, have a Windows CFAPI client
- Added additional features as discovered to be necessary for that
- What's new:
 - “Role” for discovering home and trash nodes
 - “View on Web” URLs for nodes and trash
 - onExists handling
 - Immutable blobId/size/type for nodes

Roles

- Initial role set is pretty basic
 - root
 - home
 - temp
 - trash
- Do we need a registry?
- Any other roles we should create immediately?

onExists handling

- null (default) - reject if a name is already in use
- “replace” - delete the old node.
- “rename” - give this node a new name on creation (chosen by the server)
 - E.g. “filename (2).ext”

Is this JMAPpy enough?

- Ties into onDestroyRemoveChildren which applies for both replaced nodes from onExists, and destroyed nodes generally.

Immutable blobId/size/type

- Email has much immutable, calendar and contacts not so
- It's just moving the lump around under the carpet - 3 fields
 - nodeId CAN be moved to another parent or name, as well as change sharing/modified/etc, so it still has /changes.
- Should we relax this restriction?
 - Will need changes to Fastmail's data model.
 - But: more directly compatible with "inode" in POSIX.

Blobext - yet more blob magic

- Adds “convert”, “archive”, “compress”
 - Temporarily added “encrypt” but decided it’s different enough and complex enough to separate out.
- Adds “rdiff” calculation.
- Adds “chunkSize” hint
- Adds “resumable Upload” URL
- Adds “chunks” data item to Blob/get
 - Extends DataSourceObject with extra fields and “supply to check” semantics.
- Adds “noPersist” hint to Blob/upload.

Convert / Archive / Compress

Convert: resize images - useful for creating thumbnails.

- You can return a synthetic blobId immediately and either generate in the background or generate later on download attempt.

Archive:

- TODO: add “recursive” flag to Archive where directory includes all content.

Compress:

- See later - noPersist.

rdiff

- Not sure if there's an IETF-stable reference for the rdiff format
 - Do we need to do an informative doc describing it?
- Would be very useful for changed large files - you can reduce bandwidth use significantly.
- BUT MAYBE chunks with digests gets you a long way there for the common case.

ChunkSize

This one led to some debate on the list.

Fastmail server stores individual chunks by size of the uncompressed content (currently 64Mb at a time)

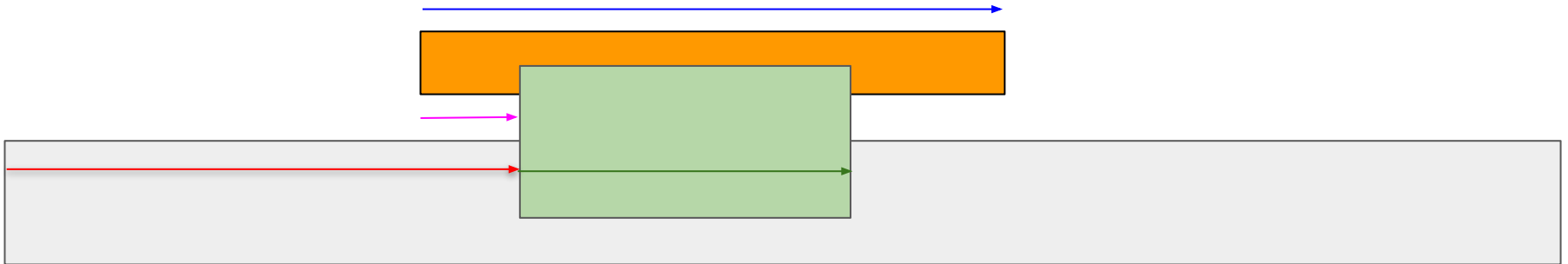
- Upload separately, combine with Blob/upload.
- digest:sha available for each chunk, can compare with local data to avoid uploading unchanged chunks.
- Don't HAVE to upload at chunk size boundaries, Blob/upload will join whatever you provide, but can optimize.

Resumable upload - needed?

Maybe the existing upload URL could support resumable uploads with autodetection without needing us to add this. I haven't used it yet.

Blob/get chunks (list of DataSourceObjects)

- New fields added to the DataSourceObject as well:
 - **blobId**: existing - source spec (can also do be raw text or base64 encoded octets on upload)
 - **size**: octet size of the whole data source (blob, or text length) - if specified, MUST match
 - **position**: octet offset of the start of this object in the containing blob, if specified, MUST ...
 - **offset**: position within the data source that copying starts from
 - **length**: number of octets within the data source to copy
 - **digest***: digest of the output range



noPersist hint

- Indicate to the server that you are never going to use this blobId after this set of MethodCalls (i.e. the only time you'll use it is a backref you have just provided)
 - This allows the server to optimise e.g. tar.gz creation without keeping the intermediate tar.
- Suppresses the “created” item for the specific creation with this hint.
- Server MAY return it anyway, if it created a persistent blob

Quota and Expiry?

Do we need to add special types for quota here?

- any blobext things that could have quotas?
- FileNode: other than disk usage quota, could also do a “maximum number of nodes” like the inode limit “df -i” on some POSIX filesystems.
- Add ‘expires’ hint to created blobs, that if they are unreferenced that will likely be cleaned up at or after a particular time.

Next Steps

- Call for adoption for blobext
 - Can't just add things to 'blob' capability; capability items are immutable once published.
- Implement more clients and servers to see that this works for other uses
 - Particularly interested in doing a basic POSIX-filesystem based server and see how much metadata it needs to be a good server.
- Not adding metadata here, do that in the generic metadata draft