Resumable Uploads

draft-ietf-httpbis-resumable-upload

IETF 117: July 26, 2023

Jonathan Flat

Implementation

tusdotnet server

NIOResumableUpload server

URLSession client on Apple platforms

Server implementation observations

Two approaches:

1. Additional service defines a resumable upload, then sends the complete upload to a backend service
2. Additional service handles resumption by wrapping backend logic

Feedback from those with backend integration experience?

Upload-Complete header field (#2500)

Use `Upload-Complete` instead of `Upload-Incomplete` (current)

Client:

- `Upload-Complete: ?1` This request completes the upload
- `Upload-Complete: ?0` This request does not complete the upload

Server:

- `Upload-Complete: ?1` The upload is complete
- `Upload-Complete: ?0` The upload is not complete

Symmetric, so server can simply echo the field on success

Upload-Complete header field (#2500)

Discussion:

- Can we lock down `Upload-Complete` as the preferred name?
- Other thoughts on symmetric vs. asymmetric naming?

Multi-version support (#2597)

Current:

- Client sends the version it supports (e.g. `Upload-Draft-Interop-Version: 3`)
- Server advertises resumable upload if it supports that version

What if the client updates to a newer version than the server?

- Client loses resumable upload capability

Can we make this transition smoother as the draft evolves?

Multi-version support (#2597)

Discussion:

- Interest in solving this multi-version issue?
- Insight/experience with draft versioning like this?

Adopting byte range PATCH (#2501)

draft-wright-http-patch-byterange (call for adoption in HTTPAPI)

Eliminates the need for an `Upload-Offset` field

Upload Appending Procedure

- `Upload-Offset: 50` becomes `Content-Range: bytes 50-199/200`
- If `Content-Length` is unknown, then `Content-Range: bytes 50-/*`

Offset Retrieving Procedure

- Server responds with `Content-Length: 50`

Adopting byte range PATCH (#2501)

Discussion:

- Thoughts on adopting byte range PATCH?
- Do we want to depend on a draft in another WG?

Upload progress via informational responses (#2291)

Server could send multiple 1xx responses with the current `Upload-Offset`

Allows the client to release memory associated with the transmitted data

Discussion:

- Should this be included in the draft?

Fetch API proposal (WHATWG #1626)

Opened issue in `whatwg/fetch`

- Asked for feedback on the draft
- Proposed integration with the Fetch API

Labeled “`needs implementer interest`”

Other open issues

- Error handling for Upload Creation Procedure (#2596)
- Server should reject upload resumptions with inconsistent Content-Length (#2544)
- How to get the response after upload is complete (#2312)
- Resumable upload with forms? (#2247)
- Prioritization of concurrent uploads (#2242)