

Resumable Uploads

draft-ietf-httpbis-resumable-upload

IETF 125: March 20, 2026

Guoye Zhang, Marius Kleidl, Lucas Pardue

What changed since -10?

- Mostly editorial
- Allowing the server to skip ahead

Open issues

1. Guidance on the client retry behavior ([#3275](#))
2. Early responses ([#3398](#))
3. Lost responses after completed uploads ([#3270](#))

1. Guidance on the client retry behavior (#3275)

Current

4xx not retriabale, 5xx retriabale

Proposal

Do not retry: Upload-Complete: ?1

Retry: 409 (Conflict) + Upload-Complete: ?0

Retry with smaller chunk: 413 (Content Too Large)

Retry with delay: 429 (Too Many Requests)

2. Early responses (#3398)

Upload-Complete: ?1

Client: last chunk

Server: the response targeted by the initial resource

Possible even if the server didn't receive the complete request

Bidirectional streaming is out of scope

2. Early responses (#3398)

Upload-Complete: ?1

Client: last chunk

Server: the response targeted by the initial resource

Possible even if the server didn't receive the complete request

Bidirectional streaming is out of scope

Editorial change

3. Lost responses after completed uploads (#3270)

Current

The server SHOULD keep the upload resource available for a reasonable amount of time after the upload is complete.

But resumable uploads as defined in this document do not offer functionality to recover the missed response.

Questions

1. Is this in scope for the draft?
2. How do we retrieve the lost response?

3. Lost responses after completed uploads (#3270)

Proposal 1

Out of scope

Remove the recommendation and leave it up to the implementers

3. Lost responses after completed uploads (#3270)

Proposal 1

Out of scope

Remove the recommendation and leave it up to the implementers

Proposal 2

Reduce SHOULD to MAY

Mention a 0-length append to retrieve the lost response

Remove the completed-upload problem type

Any other issues?

Please let us know your implementation experience

Draft interop version

Apple HTTPClient / URLSession: version 6