

# NSURLCache and Byte Ranges

**!** This thread has been locked. Questions are automatically locked after two months of inactivity, or sooner if deemed necessary by a moderator.



709

**Update** In macOS 10.13 and friends we made a change such that `NSURLSession` will no longer cache responses with a 206 Partial Content status. While this doesn't completely resolve this issue, it does mean that it crops up much less frequently in practice.

The cache used by `NSURLSession` (`NSURLCache`) completely ignores byte ranges. This means that if you make a byte range request and it gets put in the cache, and then later make a full request (or a different byte range request), that second request can be satisfied by the cache, giving you back completely the wrong data. Not good.

This is a known, long-standing limitation of `NSURLCache`, but it's never been properly documented. We do plan to enhance `NSURLCache` to do the right thing here but there's no sign of that happening any time soon (r. 12581302).

In the meantime, here's my advice...

If you're making a byte range request, ensure that the response doesn't go into the cache. The easiest way to do that is to implement the `-NSURLSessionDataTask:willCacheResponse:completionHandler:` delegate method and return nil. This ensures that a cached response to your byte range request doesn't confuse anyone else.

If you're making byte range requests and want to take advantage of caching you have two options:

- Cache externally — For example, if you're using byte range requests to download to a file, you can use the file itself as a place to store the results.
- Cache full resources — If you want to cache responses in the standard cache used by `NSURLSession`, make sure you cache the full resource. For example, if you're using multiple connections to download a resource, store the results yourself and then, on the last connection, when you have all the data for the resource, have `-NSURLSessionDataTask:willCacheResponse:completionHandler:` return an `NSCachedURLResponse` that represents the entire resource.

Alternatively, you could force the entire resource into the cache using `-storeCachedResponse:forRequest:`.

We have another bug on file (r. 12661119) requesting that this advice be added to the standard developer documentation.

Share and Enjoy

Quinn "The Eskimo!" @ Developer Technical Support @ Apple  
`let myEmail = "eskimo" + "1" + "@" + "apple.com"`

CFNetwork

Asked 3 years ago by eskimo

Reply to this question

This site contains user submitted content, comments and opinions and is for informational purposes only. Apple disclaims any and all liability for the acts, omissions and conduct of any third parties in connection with or related to your use of the site. All postings and use of the content on this site are subject to the Apple Developer Forums Participation Agreement.