

## 응답헤더 설정으로 웹사이트 성능 개선하기

---

보통의 HTML 문서는 이미지와 CSS, JS 등 여러가지 정적인 파일들로 구성됩니다. 이 파일들의 통신을 형성하기 위한 규칙 중 하나로 헤더(Header)라는 것이 있습니다. 이것은 크게 요청(Request)과 수신(Response)으로 나뉩니다. 요청부는 특수한 경우를 제외하고 브라우저가 만들어 내는 것이 보통으로, 요청 형식, 브라우저 종류(User-Agent), 데이터 유형(Accept), 쿠키(Cookie), 리퍼러(Referer), 호스트(Host) 등의 정보를 서버로 전달합니다. 요청 형식에는 프로토콜(protocol)과, 문서명, 그리고 메서드(method)를 기본으로 포함합니다. "Get /document.html HTTP/1.0"과 같은 형식이죠. 웹 서버는 이렇게 받은 요청을 처리하고 데이터와 함께 응답 헤더를 씌어 보냅니다. 응답 헤더는 응답 결과, 응답 일시(Date), 서버 종류(Server), 콘텐츠 종류(Content-type), 콘텐츠 길이(Content-Length), 데이터 보관일시(Expires), 마지막 작성한 일시>Last-Modified), ETag(해시타입 구분자) 등을 포함하여 토해냅니다. 응답 결과물로는 "HTTP/1.0 200 Found"와 같은 서버 상황(status)을 코드로 작성합니다. 만약 서버로부터 요청한 파일이 존재하지 않는다면 "404 Not Found"를 내뱉어 브라우저에게 문서가 없다는 사실을 알리죠.

Inspect	Clear	All	HTML	CSS	JS	XHR	Images	Flash
Console	HTML	CSS	Script	DOM	Net	YSlow		
+	firejune.com		firejune.com	43 KB	82ms			
+	hybrid.css,mystory.css		firejune.com	5 KB	150ms			
+	back.gif		firejune.com	743 b	69ms			
+	blog_logo_0.jpg		firejune.com	18 KB	227ms			
+	535.recent.thumb.jpg		firejune.com	8 KB	382ms			

  

Headers	Response
Response Headers	
Date	Tue, 07 Aug 2007 15:11:51 GMT
Server	Apache
X-Powered-By	PHP/4.4.1
Accept-Ranges	bytes
Content-Length	7665
Etag	"9b4efc32c22d0eeabdbfc9b667a93c0"
Cache-Control	max-age=7776000
Expires	Mon, 05 Nov 2007 15:11:51 GMT
Last-Modified	Sat, 10 Mar 2007 14:05:01 GMT

자, 이제 본론으로 들어가서, 응답 헤더로부터 어떠한 성능향상을 기대할 수 있을까요? 그렇습니다. 바로 브라우저 캐싱(caching)입니다. 응답헤더에 붙어오는 ETag, Expires, Last-Modified는 더이상 같은 정보를 요청하지 않아도 된다는 중요한 정보를 브라우저에게 알려줍니다. 구글의 경우, Expires가 1년으로 설정되어 있습니다. 즉, 한번 퍼간 데이터는 1년동안 다시 요청하지 말라는 뜻이죠. 그렇다고 브라우저가 정말로 1년동안 요청행위를 금지 당하는 것은 아닙니다. 까칠한 브라우저는 페이지가 다시 열리는 시점에서 기존의 캐시정보들을 무시하고 예전에 읽었던 파일이라도 다시금 불러오거나 확인요청을 보냅니다. 캐시정보를 인지한 브라우저는 페이지에(도메인을 떠나지 않고) 머물러 있는 동안 새로 읽으라는 명령(Ctrl + F5 또는 리로드)이 있기 전에 다시금 요청하는 일은 없습니다.

이제 여러분의 사이트가 들고있는 이미지, CSS, JS, SWF 등의 파일들에는 어떠한 헤더가 붙어서 날라오는지 확인해 보세요. 아차, 그전에 확인하는 방법을 알려드려야 겠군요. 사용중인 브라우저가 IE라면 개별 파일의 헤더를 실시간 확인하는 일이 쉽지 않습니다.(불가능하지 않나요?)  
 파이어폭스의 개발 플러그인 중 하나인 **파이어버그**를 설치하면 좌측에 있는

그림처럼 쉽게 알 수 있죠. 참고로 아래는 이 곳에서 사용되는 이미지의 응답 헤더입니다.

```
Date : Tue, 07 Aug 2007 15:11:51 GMT
Server : Apache
X-Powered-By : PHP/4.4.1
Accept-Ranges : bytes
Content-Length : 7665
Content-Type : image/jpeg
Etag : "9b4efc32c22d0eeabdbbfc9b667a93c0"
Cache-Control : max-age=7776000
Expires : Mon, 05 Nov 2007 15:11:51 GMT
Last-Modified : Sat, 10 Mar 2007 14:05:01 GMT
Keep-Alive : timeout=1, max=95
Connection : Keep-Alive
```

만약에 Last-Modified와 Expires 또는 ETag가 잘못 설정되어 있거나 없으면, 새로 불러들여야 할 파일이 아님에도 브라우저는 가차없이 다시 읽어들입니다. 서버의 access 로그에 접근할 수 있는 환경이라면 확인이 더욱 빠릅니다. 그렇지 않은 경우 파이어버그의 NET 화면에서 캐시된 양을 간접적으로 확인 할 수 있습니다. 파이어폭스의 경우 ETag만으로도 캐시여부를 알아내지만 IE는 그렇지 않을 때도 있습니다. 파일의 캐시여부를 확인하기 위한 요청을 제차 날리기도 하며, 심지어는 하나의 파일을 3,4번씩 요청하는 경우도 있습니다. 예를 들어 "new"가 찍힌 아이콘이 한 페이지에 10번 중복 사용 되었다면, 최소 10번에서 40번의 요청이 일어나기도 합니다. 다른 예를 들면, 자바스크립트와 스타일시트를 사용한 애니메이션에서 마우스 커서모양에 모래시계가 연달아 나타나는 현상과 함께 모션이 부드럽지 못한 것을 들 수 있으며, 롤오버/아웃

이벤트에 이미지를 적용한 경우 계속해서 불러오는 현상이 발생합니다. 물론 방문객 브라우저의 캐시설정이 정상이라도 발생하죠.

이러한 현상은 대부분 Expires헤더로 해결 할 수 있습니다. [YSlow\(성능 개선 도구\)](#)에서는 Expires를 모든 파일에 사용할 것을 권고([Add an Expires Header](#))하고 있습니다. 지금까지 송/수신 헤더를 그리 중요하게 여기지 않았습시다만, YSlow로 인해 헤더에 대하여 공부할 수 있었던 좋은 계기가 되었고 서버가 어떠한 응답 헤더를 주느냐에 따라 브라우저와 서버 양쪽의 성능을 모두 끌어올릴 수 있다는 사실을 알았습니다. 서버로부터 응답헤더를 어떻게 커스토마이징 하는지는 다루지 않겠습니다. 서버의 종류, 언어, 버전, 모듈 등에 따라 헤더를 설정하는 방법론은 헤아릴수 없을 정도로 많거든요.(구글링 강추)