(I used Chrome as a baseline for comparison, and other browsers may differ slightly).

|  |  |  |
| --- | --- | --- |
|  | **Angular $cacheFactory** | **Browser HTTP cache (Chrome)** |
| Storage | Javascript heap memory [(more)](https://github.com/angular/angular.js/blob/v1.2.14/src/ng/cacheFactory.js#L70) | disk [(more)](http://www.chromium.org/user-experience/user-data-directory) |
| Eviction algorithm | Least recently used (LRU) | LRU [(more)](http://www.chromium.org/developers/design-documents/network-stack/disk-cache#TOC-Eviction) |
| Time-to-live | until page refresh | Determined by HTTP headers |
| Support for revalidation | No | Yes, with ETags |
| Requires cooperation of  remote server | No | Yes |
| Example | $http({  method: 'GET',  cache: **true**,  url: 'http://date.jsontest.com/  '}) | Cache-Control: max-age=1800, private  Vary: Cookie  Content-Type: application/json  ... |

Overall, I'd say the browser has the advantage, but there are many gotchas to look out for with HTTP:

* First, the server must be coorperating by sending suitable headers (usually easy if you control the remote resource, rare for third-parties).
* If your resources require authenication, you'll need to set the Vary: header, and Cache-control: private. In the above example cookies are used, but other methods will work. For example if you authenticate with OAuth 2.0, the header would be Vary: Authorization.
* Be careful with proxies, for example Varnish [does not support "private" by default](https://www.varnish-cache.org/trac/wiki/VCLExampleHitMissHeader).
* When using HTTPS, Chrome and Safari will not cache resources from servers with self-signed certificates. This sometimes makes testing more complicated.
* Safari's implementation of Vary: is [broken for cookies](https://bugs.webkit.org/show_bug.cgi?id=71509), and possibly other headers.

angular-cache

There is a third option if you can't solve the gotchas above, but need more flexibility: [angular-cache](http://jmdobry.github.io/angular-cache/). By using this replacement for $cacheFactory, you get more storage options (localStorage and sessionStorage), and a configurable time-to-live.