The HTML5 cache manifest file

HTML5 introduced a "manifest file" as an attribute to the `<html>`element. This file will tell the browser what to cache, what not to cache, and some other goodies we will see in this section.

Please find below an example of the inclusion of a manifest file (it is best practice/recommendation to use the `.appcache`suffix):

```
<html manifest="myCache.appcache">
...
</html>
```

You must include the `manifest` attribute on every page for which you want to cache resources (images included in the page, JavaScript files, CSS files). The manifest file will contain lines that will tell which image/js/css needs to be cached, which ones must never be cached, etc.

The page with the manifest is *de facto* included in the list of files that will be cached. This means that any page the user navigates to that include a manifest will be implicitly added to the application cache.

The browser does not cache a page if it does not contain the manifest attribute. The default cache behavior, prior to HTML5, will be used (depending on the browser versions). **If you want a Web site or a Web app to work offline then, please include a manifest in every HTML page!**

The manifest attribute value is a relative path (relative to the page URL), but an absolute URL can be used (not recommended) if it respects the same origin policy.

THE MANIFEST FILE MUST BE SERVED WITH THE CORRECT MIME TYPE

The HTTP server that serves your files must be configured so that `.appcache` files are served with the MIME type t`ext/cache-manifest`. For example, with the Apache server, this line must be added in the HTTP.conf configuration file (or in the .htaccess files):

```
AddType text/cache-manifest .appcache
```

WHAT DO WE PUT IN THE MANIFEST FILE?

First example: cache an HTML page that displays the current time (clock). It uses three pages: HTML, CSS and JavaScript. This example is taken from the W3C specification, you can try the online example here.

The manifest file is:

```
CACHE MANIFEST
clock.html
clock.css
clock.js
```

Lines 2-4 show that a given HTML page, in which this manifest is included, asks the browser to cache three files: the HTML page itself (`clock.html`, cached by default, but the specification recommends adding it to the manifest as best practice), a CSS file `clock.css` and a JavaScript file `clock.js`.

Note that the first line "CACHE MANIFEST" is mandatory.

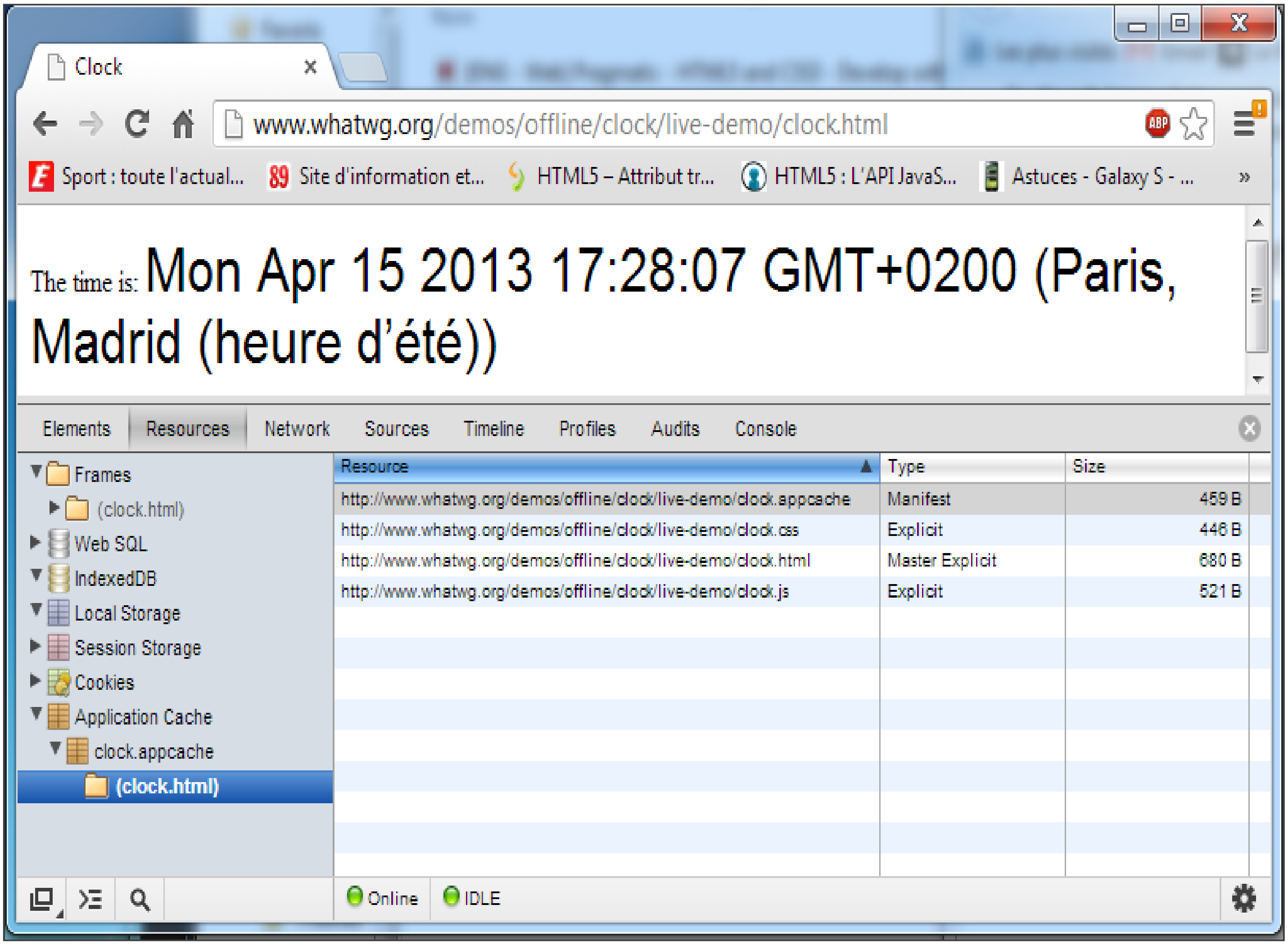
PITFALLS TO AVOID WHEN USING THE HTML5 CACHE

PITFALL #1 : When a file is available in the cache and on the remote

HTTP server, it will always be retrieved from the cache! A next section is dedicated to "updating the cache", and we will explain how to control this and update the files in the cache.

PITFALL #2: If one file cannot be retrieved and cached, zero files will be updated in the cache. There is not "partial update" possible. A best practice is to always validate you manifest file using one of the tools listed at the end of this chapter.

The first time you run this in Chrome, you may see the cache content with the dev. tools (F12):

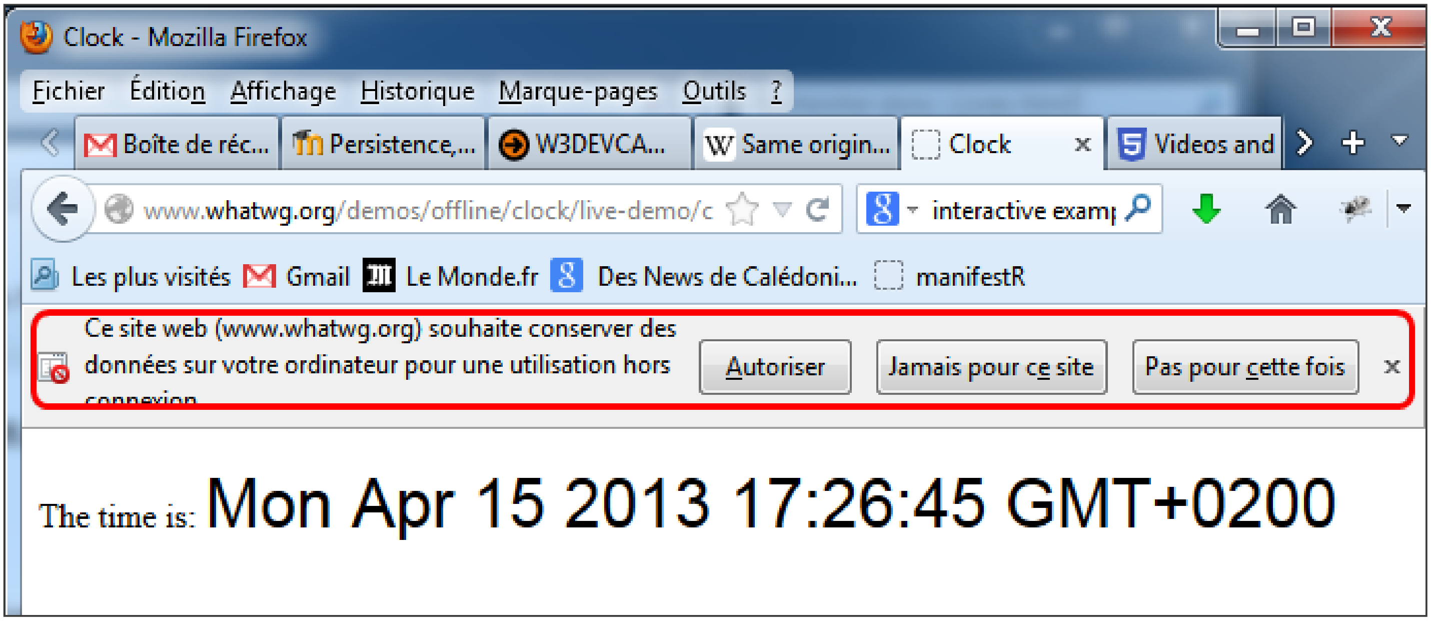


www.whatwg.org/demos/offline/clock/live-demo/clock.html

The time is: Mon Apr 15 2013 17:28:07 GMT+0200 (Paris, Madrid (heure d'été))

Resource	Type	Size
http://www.whatwg.org/demos/offline/clock/live-demo/clock.appcache	Manifest	459 B
http://www.whatwg.org/demos/offline/clock/live-demo/clock.css	Explicit	446 B
http://www.whatwg.org/demos/offline/clock/live-demo/clock.html	Master Explicit	680 B
http://www.whatwg.org/demos/offline/clock/live-demo/clock.js	Explicit	521 B

With the same example, Firefox asks if you agree to cache some data (the sentence is in French here but it says: "This Web site would like to save data on your computer for offline use. Authorize? Never? Just for this time?"):



Ce site web (www.whatwg.org) souhaite conserver des données sur votre ordinateur pour une utilisation hors connexion

The time is: Mon Apr 15 2013 17:26:45 GMT+0200

Let's have a look at another example of `manifest.appcache`(this one comes from the webdirections.com Web site), that does a little more:

```
CACHE MANIFEST
CACHE:
#images
/images/image1.png
/images/image2.png
#pages
/pages/page1.html
/pages/page2.html
#CSS
/style/style.css
#scripts
/js/script.js
FALLBACK:
/ /offline.html
NETWORK:
login.html
```

This time we notice a few additional things:

- It's possible to add comments starting with #
- There are three different sections in capital letters: CACHE, FALLBACK and NETWORK

These three sections are optional, we did not have them in the first example. But as soon as you indicate one of them, you must indicate the others. (CACHE was defaulted in the first example as we had no explicit section declarations).

The CACHE section specifies the URLs of the resources that must be cached (generally relative to the page, but they can also be absolute and external, for example for caching jQuery from a Google repository, etc.). These resources will be 1) cached when online, and 2) available from the cache when offline.

The NETWORK section is the contrary of the CACHE section: it is useful for specifying resources that should NOT be cached. These resources 1) will not be cached when online, and consequently 2) will not be available when the user is offline. EVEN IF THE BROWSER HAS CACHED THEM IN ITS OWN "PRE HTML5" cache! In the previous example, at line 23, the login.html file (the one with the login/password form...) is never cached. Indeed, entering login/password and pressing a "login/connect/signup" button is useless if you are offline.

Using a wildcard * in that section is also common practice, this means "do not cache all files that are not in the CACHE or FALLBACK section":

```
NETWORK:
*
```

Partial URLs may also be used in that section, like "/images" that means, all URLs that ends with images/*... should not be cached. Notice that wildcards and partial URLs are not allowed in the CACHE section, where all individual files must be explicitly specified.

The FALLBACK section specifies resources that will be displayed when a

resource that is not available when offline is requested. For example, a `login.html` file must not be cached nor be available when offline. In that case, accessing to `http://.../login.html` will cause `offline.html` to be displayed (and this file will be cached, this is forced by being in the FALLBACK section). The "`/ /offline.html`" in the FALLBACK section of the example says that for any resource that is not available in the cache ("/" means here "any resource"), show the `offline.html` page.

Partial URLs can be used too. For example:

```
/images/ /images/missing.png
```

... tells us that all images in the sub-directory "images" relative to the Web page that includes the manifest, if unavailable in the cache when the browser is offline, will be replaced by an image named "`missing.png`".