

# HTML5 cache / offline applications

## INTRODUCTION

Since the 90's, a long time before HTML5 ever existed, all browsers have been implementing a native way of caching files. However, the implementations and heuristics differed from one browser to another, and there was no easy way to control (at the application layer) which parts will be cached.



The HTML5 specification brings a new way to handle caching. As stated in the ["offline Web applications"](#) section of the document: *"In order to enable users to continue interacting with Web applications and documents even when their network connection is unavailable — for instance, because they are traveling outside of their ISP's coverage area — authors can provide a manifest which lists the files that are needed for the Web application to work offline and which causes the user's browser to keep a copy of the files for use offline."*

By "offline applications", this means:

1. **That a Web site can be browsed while no internet connection is available.** But it also says that:
2. **The Web page may be an "application", not just a static page.**

Meaning, it may be a video game that usually retrieves data from a Web server, saves high scores, etc. It may be a mail application (like Gmail) that shows the most recent emails, checks for new ones, and has features for composing and sending new emails, etc.

In these cases, having a controllable "cache API" helps implement things like:

- When offline, you can read and compose mails, but you cannot get new emails or send new ones.
- A more complex behavior would be that, when offline, you can write emails that will be stored locally on the client side and sent only when the connectivity is available.
- Or, in a static Web site, everything can be retrieved from the cache, so that a user is able to read the Web site offline, but the login page will not be available... instead, a disclaimer page will be displayed, etc.

The HTML5 cache will try to address parts of these needs, and is pretty useful for mobile Web applications, where connections are not always available/reliable.

**Note:** Native mobile applications developers often push the following argument that "a Web app cannot run offline". **This is a false statement!**

Using the HTML5 cache is both useful for mobile applications, as these device can often be offline, and for traditional Web developers who do not target specifically mobile devices. Any Web site (even made 100% of static pages/images/videos/etc.) can benefit from being intelligently cached, as the HTML5 cache can seriously decrease the load time for any given site, especially if the same browser/person visits the Web site regularly, as illustrated by these pictures below (borrowed from [a MSDN blog post about offline applications](#) that is worth reading):

## First Run



Fetch Info  
From Network

Cache Resources  
Locally

Fetch Info  
From Cache

No Network  
Available

## Later Runs



## CURRENT SUPPORT

Current support is very good, [as stated by the caniuse.com Web site](http://caniuse.com). All major browsers, desktop or mobile, support the HTML5 cache since 2012.

## EXTERNAL RESOURCES (OPTIONAL, FOR THOSE WHO WANTS TO GO DEEPER ON THAT TOPIC)

### Tutorials

- [The W3C specification about Offline Web Applications / cache](#)
- [Article from html5rocks.com about the cache API](#)
- [Appcache facts: good resume of the cache API + best practices](#)

### Tools

- [Cache manifest validator tool](#)
- [ManifestR bookmarklet, a tool for automatically generating a cache manifest file](#)