Building Standalone Applications in JavaScript

Asked 16 years, 4 months ago Modified 14 years, 9 months ago

Viewed 34k times

23

With the increased power of JavaScript frameworks like YUI, JQuery, and Prototype, and debugging tools like Firebug, doing an application entirely in browser-side JavaScript looks like a great way to make simple applications like puzzle games and specialized calculators.

Is there any downside to this other than exposing your source code? How should you handle data storage for this kind of program?

Edit: yes, Gears and cookies can be used for local storage, but you can't easily get access to files and other objects the user already has around. You also can't save data to a file for a user without having them invoke some browser feature like printing to PDF or saving page as a file.

javascript deployment web-applications browser

Share

Improve this question

edited Aug 23, 2008 at 17:22

Follow

Chris
6,842 ● 6 ● 53 ● 67

asked Aug 18, 2008 at 19:52
Ben Combee
17.4k ● 6 ● 42 ● 42

2 I was working at Palm when I asked this, and what did Palm introduce in early 2009? webOS, a system where all apps were written in JavaScript. :) – Ben Combee Sep 2, 2010 at 23:27

14 Answers

Sorted by: Highest score (default)

16 ✓

I've written several application in JS including a spreadsheet.

Upside:

- great language
- short code-run-review cycle
- DOM manipulation is great for UI design
- clients on every computer (and phone)

Downside:

- differences between browsers (especially IE)
- code base scalability (with no intrinsic support for namespaces and classes)
- no good debuggers (especially, again, for IE)

- performance (even though great progress has been made with FireFox and Safari)
- You need to write some server code as well.

Bottom line: Go for it. I did.

Share Improve this answer

Follow

answered Aug 18, 2008 at 21:19

Avner

5,831 ● 2 ● 31 ● 34

5 IE8 Developer Tools actually has a good javascript debugger. I know IE=PureEvil but, just thought I'd point it out. The hell that has been IE is getting better. – jaywon Nov 21, 2009 at 4:53

7

Another option for developing simple desktop like applications or games in JavaScript is Adobe AIR. You can build your app code in either HTML + JavaScript or using Flash/Flex or a combination of both. It has the advantage of being cross-platform (actually cross-platform, Linux, OS X, and Windows. Not just Windows and OS X).

Heck, it may be the only time in your career as a developer that you can write a web page and ONLY target ONE browser.

Share Improve this answer

Follow

answered Aug 18, 2008 at 21:37

ScottKoon

3,493 ● 6 ● 28 ● 29

1 Since june 2011 Adobe AIR no more supports Linux. blogs.adobe.com/open/2011/06/… – AlexStack Aug 11, 2011 at 11:48

▲ 5 ▼

SproutCore is a wholly JavaScript-hosted application framework, borrowing concepts particularly from Cocoa (such as KVO) and Ruby on Rails (such as using a CLI generator for your models, views and controllers). It includes Prototype, but builds plenty of stuff such as sophisticated controls on top of that. Its Photos demo is arguably impressive (especially in Safari 3.1).

Greg already pointed you to Gears; in addition, HTML 5 will come with a standardized means of local storage. Safari 3.1 ships with an implementation where you have a per-site SQLite database with user-settable size maximums, as well as a built-in database browser with SQL querying. Unfortunately, it will be a long time until we can expect broad browser support. Until then, Gears is indeed an alternative (but not for Safari… yet!). For simpler storage, there is of course always cookies.

Share Improve this answer Follow

answered Aug 18, 2008 at 20:12

Sören Kuklau
19.9k ● 8 ● 55 ● 90

▲

The downside to this would be that you are at the mercy of them having js enabled. I'm not sure that this is a big

2

deal now. Virtually every browser supports js and has it enabled by default.

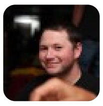
Of course the other downside would be performance. You are again at the mercy of the client handling all the intensive work. This also may not be that big of a deal, and would be dependent on the type of app you are building.

I've never used Gears, but it looks like it is worth a shot. The backup plan would be to run some server side script through ajax that dumps your data somewhere.

Not completely client side, but oh well.

Share Improve this answer Follow

answered Aug 18, 2008 at 20:02



mk.

26.2k ● 13 ● 39 ● 41

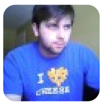
2

Nihilogic (not my site) does a lot of stuff with Javascript. They even have several games that they've made in Javascript.

I've also seen a neat roguelike game made in Javascript. Unfortunately, I can't remember what it was called...

Share Improve this answer Follow

answered Aug 18, 2008 at 20:15



Ryan Fox

10.2k ● 5 ● 39 ● 48

2

If you want to write a standalone JavaScript application, look at XULrunner. It's what Firefox is built on, but it is also built so that you can distribute it as an application runtime. You will write some of the interface in JavaScript and use JavaScript for your code.

Share Improve this answer Follow

answered Oct 28, 2008 at 0:33

pc1oad1etter

8,617 ● 10 ● 50 ● 64

1

Gears might provide the client-side persistent data storage you need. There isn't a terribly good way of not exposing your source code, though. You could obfuscate it but that only helps somewhat.

I've done simple apps like this for stuff like a Sudoku solver.

Share Improve this answer Follow

answered Aug 18, 2008 at 19:55

Greg Hewgill

990k ● 191 ● 1.2k ● 1.3k

1

You might run into performance issues given that you're completely at the mercy of the client's Javascript interpreter. Gears would be a nice way of data storage, but I don't think it has penetrated the market that much. You could just use cookies if you're not fussy about that kind of thing.

Share Improve this answer

Follow

answered Aug 18, 2008 at 20:31

conmulligan

7,138 ● 6 ● 34 ● 45

1

I'm with ScottKoon here, Adobe AIR is great. I've really only made one really nice (imho) widget thus far, but I did so using jQuery and Prototype.js, which floored in such wonderful ways because I didn't have to learn a whole new event model. Adobe AIR is really sweet, the memory foot print isn't too bad, upgrading to a new version is built into AIR so it's almost automatic, and best of all it's cross-platform...they even have an alpha-version for Linux, but it works pretty well already on my Eee.

Share Improve this answer

Follow

answered Sep 2, 2008 at 16:53

AndrewBurton

407 ● 7 ● 16

Since june 2011 Adobe AIR no more supports Linux. blogs.adobe.com/open/2011/06/… – AlexStack Aug 11, 2011 at 11:47

1

Standalone games in GWT:

1. http://gpokr.com/
2. http://kdice.com/

Share Improve this answer

Follow

answered Aug 18, 2008 at 20:50

Kevin

13.1k ● 11 ● 56 ● 77

1

In regard to saving files from a javascript application:

I am really excited about the possibilities of client-side applications. Flash 10 introduced the ability to create files for save right in the browser. I thought it was super cool, so I built a javascript+flash component to wrap the saving feature. Right now it only works for creating text based files (vcard, ical, xml, html, css, etc.)

1. **Downloadify Home Page**
2. **Source Code & Documentation on Github**
3. **See It In Use at Starter for jQuery**

I am looking to add support for non-text files soon, but this is a start.

Share Improve this answer Follow

answered Nov 21, 2009 at 4:55

Doug Neiner
66.1k ● 13 ● 110 ● 118

0

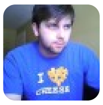
My RSS feeds have served me well- I found that Javascript roguelike!

It's called The Tombs of Asciiroth.

Share Improve this answer

Follow

answered Aug 19, 2008 at 17:03



Ryan Fox

10.2k ● 5 ● 39 ● 48

0

Given that you're going to be writing some server code anyway, it makes sense to keep storage on the server for a lot of domains (address books, poker scores, gui configuration, etc.,.) For anything the size of what you'll get in Webkit or Gears, you can probably also keep it on your server.

The advantage of keeping it on your server is two-fold:

1. You can integrate it fairly simply as a Model layer in a typical MVC framework, and,
2. Users get a consistent view without being tied to their browser/PC, or in a less-than-ideal environment (Internet Cafés).

The server code for handling this can also be fairly trivial, particularly if it's written with this task in mind, so it's not a huge cognitive burden.

Share Improve this answer

Follow

answered Sep 16, 2008 at 22:59

Brian Cully

536 ● 2 ● 11

Go with qooxdoo. They recently realsed 1.0, although most users of it say it was ripe for 1.0 at least two versions ago.

0

I compared qooxdoo with YUI and ext, and I think qooxdoo is the way to go for programmers - YUI isn't that polished as qooxdoo, from a programmer's point of view and ext has a not so friendly licensing model.

A few of the strong points (for me) of qooxdoo are:

- extremely clean code
- the nicest OO programming model I've seen among Javascript frameworks
- an extremely rich UI widget library

It also features a test runner for unit tests, an API doc generator and reader, a logging facility, and several useful features for debugging, grouped under something called Inspector.

The only downside is that there aren't readymade themes (something like skins) for qooxdoo. But creating your own theme is quite easy.

Share Improve this answer Follow

answered Mar 9, 2010 at 14:16

Me myself and I
1