

Choosing a JavaScript library

Intro

One of [Kauri](#)¹'s goals is not to reinvent the wheel until necessary. A modern web application framework these days needs a JavaScript library to provide common, reusable functionality and save us from writing lots of repetitive code. Libraries also provide a layer of abstraction, thus hiding browser differences.

We do not want to write one ourselves because it saves time to use an existing one, plus these libraries are more likely to contain fewer bugs because of the significant user base.

Because of the multitude of Javascript libraries, we started our search by selecting only the most-used, most-popular and up-to-date Ajax-enabled libraries.

The contestants

	Latest (non-beta) release	Size	the good	the bad	Licence	Browser compatibility
Dojo ²	1.0.2 (12/15/07)	50kb+	<ul style="list-style-type: none">• OOP• CSS3• buildsystem with compression• unittest• widgets (Dijit)	<ul style="list-style-type: none">• buggy• ugly code• big	modified BSD or Academic Free 2.1	Safari 3.0.x Opera 9.0+ IE 6.0+ FF1.5+ Konqueror 3.5+
Prototype ³	1.6.0.2 (01/25/08)	~120kb	<ul style="list-style-type: none">• support for UI• Script.aculo.us⁴ for UI• OOP	<ul style="list-style-type: none">• poor documentation• extends Object.prototype (*⁵, *⁶) not since 1.4• extends js built-in objects	MIT	IE 6.0+ FF1.0+/ Mozilla 1.7+ Safari 1.2+
jQuery ⁷	1.2.3 (02/08/08)	~29kb	<ul style="list-style-type: none">• documentation only• easy to use• fast• CSS 1-3, basic xpath• CSS selector• lightweight	<ul style="list-style-type: none">• selector +effects +some-XHR• only for simple and small projects	MIT or GPL	FF1.5+ IE 6+ Safari 2.0.2+ Opera 9+

1. <http://kauriproject.org/wiki/65-kauri.html> (The Kauri Project)

			<ul style="list-style-type: none"> • chainable • plugins • namespaced • jQuery UI for widgets • cross site Ajax • compatible with others (through jQuery.noConflict()) • xml as context object (if(\$(".status",xml).text() == "2")...) 			
Ext JS ⁸	2.0.1 (01/23/08)	500kb	<ul style="list-style-type: none"> • support • good collection of widgets • UI-centric (+ or -) • on top of jquery/ yui/ prototype or standalone • fast • client-side data model (*⁹) • CSS 3 • xpath 	<ul style="list-style-type: none"> • big¹⁰ 	<p>LGPL 3.0</p>	<p>IE 6+ FF1.5+ Safari 2+ Opera 9+</p>
Yahoo UI Library ¹¹	2.4.1 (12/19/07)	29kb+	<ul style="list-style-type: none"> • history manager • documentation • modular (fetch what you need) • event driven 	<ul style="list-style-type: none"> • slow • big • complex • verbose 	<p>BSD</p>	<p>IE 6+ FF1.5+ Safari 2+ Opera 9+</p>

			<ul style="list-style-type: none"> namespaced CSS framework Skins YUILoader¹² 			
MooTools ¹³	1.1 (05/07/07)	9kb+	<ul style="list-style-type: none"> lightweight fast easy modular Object Oriented chainable dynamic loading for images, css and javascript files 	extends many of js built-in objects	MIT	IE 6+ FF Safari Opera Camino
qooxdoo ¹⁴	0.7.3 (01/14/08)		<ul style="list-style-type: none"> buildsystem for optimazing and packaging namespaced event binding cross-browser back button support bookmarkability AOP IFrame IO¹⁵ 	non-CSS-based styling	LGPL or EPL	IE 5.5+ FF 1.0+/ Mozilla 1.3+ Opera 8+ Safari 3
GWT ¹⁶	1.4.61 (11/03/07)		<ul style="list-style-type: none"> browser history JUnit i18n Java compiled to javascript 		Apache 2.0	IE Firefox Mozilla Safari Opera

The pros and cons are (possibly biased) opinions collected from blogs and fora, or features promoted on the homepages.

Disqualified:

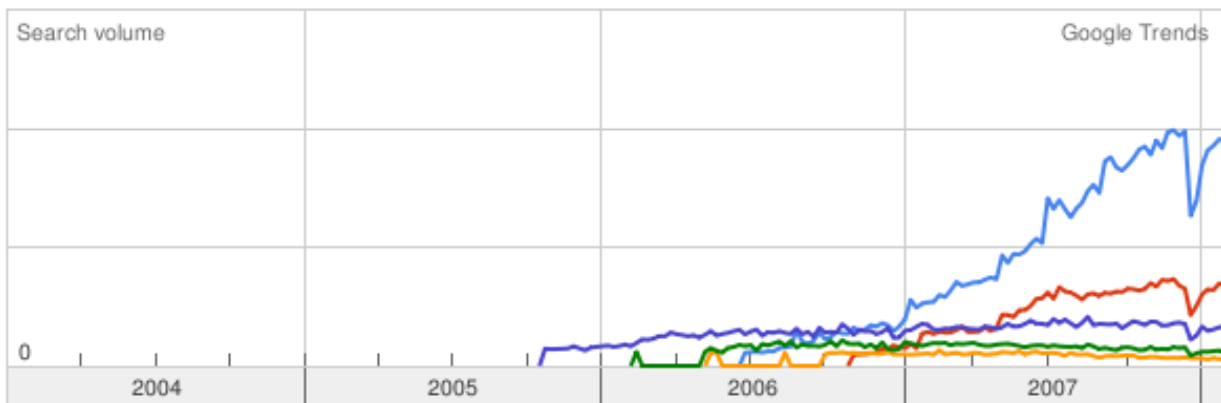
- [MochiKit](#)¹⁷: seems to be deserted
- [Rico](#)¹⁸: dying community, low support

Our requirements for a JS library within Kauri

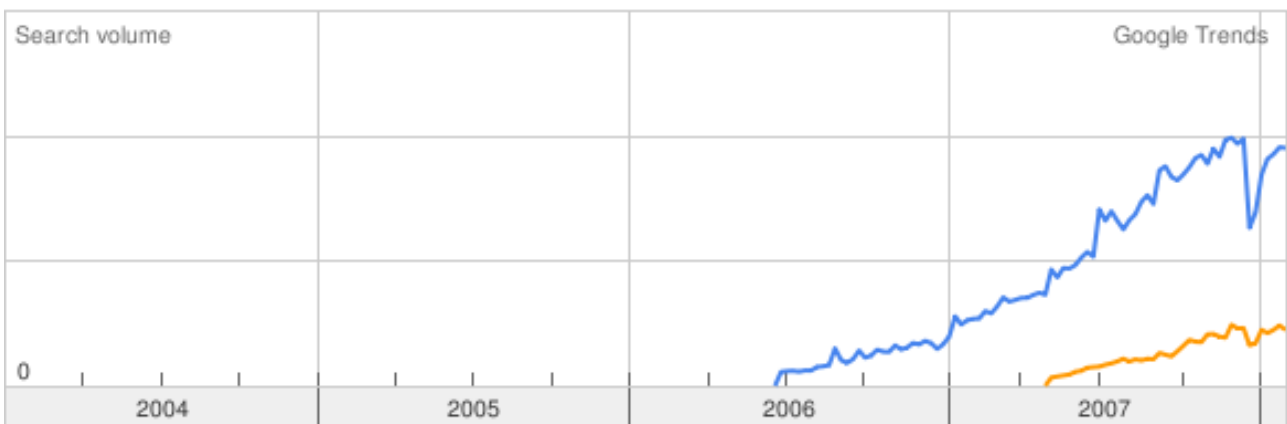
1. base for client-side restletport
2. cross-browser issues:
 - syntax
 - DOM model
 - event model
3. dynamic loading of js-files
4. event binding
5. unit testing
6. (widgets, effects)
7. integration of external widgets (Simile)
8. size, stability, maturity
9. compression
10. i18n, L10n

Popularity (by number of searches)

● jquery
 ● mootools
 ● ("yahoo ui"|"yui fra...
 ● dojo (toolkit|javasc...
 ● prototype (toolkit|j...




































● jquery
 ● qooxdoo
 ● extjs



The qooxdoo framework didn't have enough search volume to show up on the graph.











































Language

	Custom event-binding	OO	namespaced API	Namespacing	Chaining	Modular
Prototype	 19	 20		 21		
Dojo	 22	 23		 24		
jQuery	 25			 26		
MooTools	 28	 29				 (at build-time)
ExtJS	 31			 32		
qooxdoo	 33	 34		 35		
YUI	 36	 37		 38		

There are arguments why a js framework should not try to be an OO programming language (see ^{*39},^{*40} and ^{*41}).

XmlHttpRequest

All libraries contain specialized methods for Ajax calls, with support for GET and POST requests, parameters and callback functions. IFrame I/O is mostly used for file upload.

XHR API	custom HTTP request headers	Callback	Error handling	Cross-site scripting	IFrame IO	Synchronous
Prototype ⁴²						
Dojo ⁴³ * ⁴⁴					 45	
jQuery ⁴⁶						
MooTools ⁴⁷						
ExtJS ⁴⁸				 49	 50	
qooxdoo ⁵¹					 52	
YUI ⁵³			 54		 55	

39. <http://mattsnyder.com/javascript/prototype-vs-yui-round-1-oop-architecture/>

40. <http://www.geoffreymoller.com/2007/05/15/when-javascript-libraries-attack/>

41. <http://foohack.com/2007/08/yui-crockford-module-pattern-vs-prototypes-class-function/>

CSS Framework integration

- [jQuery + Blueprint](#)⁵⁶

GWT wrappers

GWT + ExtJS: [gwt-ext](#)⁵⁷, [MyGWT](#)⁵⁸

GWT + Dojo: [Tatami](#)⁵⁹

GWT + Script.aculo.us: [Script.aculo.us integration](#)⁶⁰

Java + qooxdoo: [QWT](#)⁶¹

[JavaScript Native Interface \(JSNI\)](#)⁶²

Test 1: GWT Integration

- jQuery: [Google Maps the jQuery Way](#)⁶³
- YUI: [Google Maps + Yahoo UI Lib \(YUI\) = Mashup fun](#)⁶⁴
- qooxdoo: [Google Maps widget](#)⁶⁵
- ExtJS: [Adding a Google Map to a Tab or Window](#)⁶⁶
- MooTools: [GMapsOverlay](#)⁶⁷, [google-maps-lightbox](#)⁶⁸
- Prototype: [gplotter](#)⁶⁹

Test 2: Simile

Simile Timeline and Timeplot are based on jQuery 1.2.1, which could cause conflicts.

- Dojo: [Dojo and TimeLine](#)⁷⁰
- jQuery: [How to make Timeline not conflict with jQuery](#)⁷¹
- GWT: [gwtsimiletimeline](#)⁷²
- Prototype: [Timeline](#)⁷³

Test 3: i18n support

- Dojo: has a specific i18n module (*⁷⁴).
- jQuery: no built-in i18n support. Localization is available for the datepicker-widget
- Prototype: no built-in i18n support.
- ExtJS: localization support (*⁷⁵, *⁷⁶).
- YUI: no built-in i18n support, there is however an internationalization plugin (*⁷⁷).
- MooTools: no built-in i18n support.
- qooxdoo: i18n and L10n are fully supported (*⁷⁸).
- GWT: internationalization support (*⁷⁹).

57. <http://code.google.com/p/gwt-ext/>

58. <http://mygwt.net/>

59. <http://code.google.com/p/tatami/>

60. <http://gwt.components.googlepages.com/script.aculo.usintegration>

61. <http://qooxdoo.org/documentation/contrib/contributions/qwt/about>

62. <http://code.google.com/webtoolkit/documentation/com.google.gwt.doc.DeveloperGuide.JavaScriptNativeInterface.html>

Test 4: Support for Unit testing

- qooxdoo: [testrunner](#)⁸⁰
- Prototype: there's a test framework that can be extended for custom unit tests (*⁸¹).
- jQuery: internal test framework
- Dojo: has a unit testing harness, [D.O.H.](#)⁸²
- MooTools: no unit testing
- Ext JS: no unit testing (*⁸³, *⁸⁴)
- YUI: [YUI Test](#)⁸⁵
- GWT: [JUnit integration](#)⁸⁶

Standalone js unit test tools:

- [Crosscheck](#)⁸⁷
- [Testcase](#)⁸⁸ (Prototype-based)
- [Selenium](#)⁸⁹
- [JsUnit](#)⁹⁰

Test 5: Building a custom widget

jQuery: [Writing your own plugins](#)⁹¹

Dojo: [Creating new Dojo Widget](#)⁹², [Dojo Custom Widget Tutorial](#)⁹³, [Create a Custom Javascript/AJAX Widget with Dojo](#)⁹⁴

ExtJS: [Writing Ext 2 Plugins](#)⁹⁵

Dojo, MooTools, jQuery, prototype: [Creating an AJAX Rating Widget](#)⁹⁶

The first 3 losers

- MooTools:
 - extends many of js built-in objects
 - lack of namespace
- Prototype:
 - extends js built-in objects
 - no namespacing
- qooxdoo
 - not mature enough (still beta)

The final 4 contenders

- jQuery
 - blooming community
 - beautiful and concise language

91. <http://jquery.bassistance.de/jquery-getting-started.html#plug>

92. <http://www.alexatnet.com/node/14>

93. http://www.coachwei.com/blog/_archives/2007/3/28/2841519.html

94. http://taubler.com/articles/article_Create_a_Custom_Javascript_AJAX_Widget_with_Dojo?id=5

95. http://extjs.com/learn/Tutorial:Writing_Ext_2_Plugins

96. <http://www.progressive-coding.com/tutorial.php?id=6>

- lightweight
- Dojo
 - most powerful
- Ext JS
 - possibility to combine with JUI or jQuery
- YUI
 - modular
 - documentation
 - CSS framework included

Our Winner

The JS framework we'll use and support in [Kauri](#)⁹⁷ will be [jQuery](#)⁹⁸.

Links

JavaScript Frameworks Compared

<http://www.zenperfect.com/2007/08/11/javascript-frameworks-compared/>

Top 5 javascript frameworks

<http://www.whenpenguinsattack.com/2007/04/24/top-5-javascript-frameworks/>

JavaScript Libraries By Comparison

<http://javascriptant.com/articles/24/javascript-libraries-by-comparison>

Javascript Toolkit Comparison

<http://www.ja-sig.org/wiki/display/UP3/Javascript+Toolkit+Comparison>

Why I'm moving from jQuery to ExtJs

<http://coderseye.com/2007/why-im-moving-from-jquery-to-extjs.html>

2007 Ajax Tools Usage Survey Results

http://www.surveymonkey.com/sr.aspx?sm=fXLiKcnKID6cO5bRe961aBB6NCCWytRyY3rParAYmWA_3d

A Mootools Tutorial

<http://clientside.cnet.com/wiki/mootorial>

Choosing a JavaScript library

<http://www.b-list.org/weblog/2007/jan/22/choosing-javascript-library/>

Prototype and jQuery: A code comparison

<http://ajaxian.com/archives/prototype-and-jquery-a-code-comparison>

Simplify Ajax development with jQuery

<http://www.ibm.com/developerworks/library/x-ajaxjquery.html>

Easy Ajax with jQuery

<http://www.sitepoint.com/article/ajax-jquery>

Survey of js frameworks

<http://zhenhua-guo.blogspot.com/2007/12/karajan-workflow-composition.html>

97. <http://kauriproject.org/wiki/65-kauri.html> (The Kauri Project)

98. <http://jquery.com/>