

# ASP.NET MVC alternatives? [closed]

Asked 16 years, 1 month ago   Modified 12 years, 5 months ago

Viewed 6k times



11



**Closed.** This question does not meet [Stack Overflow guidelines](#). It is not currently accepting answers.

💡 Questions asking us to **recommend or find a tool, library or favorite off-site resource** are off-topic for Stack Overflow as they tend to attract opinionated answers and spam. Instead, [describe the problem](#) and what has been done so far to solve it.

Closed 11 years ago.

[Improve this question](#)

So, I've spent enough time using ASP.NET webforms to know that I'd almost rather go back to doing classic ASP than use them. But I'm hesitant to move to ASP.NET MVC until it becomes more mature. Are there any open source alternatives?

The main thing I'm looking for is something that's easy to learn and to get a prototype up and running with. If it's any help, my main preference is a python "mix and match" approach (like say web.py/SQLAlchemy/whatever templating engine suits my fancy at the time).

[asp.net-mvc](#)[open-source](#)[webforms](#)[Share](#)[Improve this question](#)[Follow](#)

edited Jul 13, 2012 at 6:49

[tereško](#)

58.4k ● 25 ● 100 ● 150

asked Nov 19, 2008 at 3:38

[Jason Baker](#)

198k ● 138 ● 382 ● 520

---

Out of curiosity, did you select ASP.NET MVC? – [Beep beep](#)

Jun 2, 2011 at 9:41

---

## 5 Answers

Sorted by:

Highest score (default)

**11**

We have used [MonoRail](#) RC2 for our small business's online store for the past 18 months. It replaced a 7 year old disaster of classic ASP pages. MonoRail RC2 has worked well for us, serving an average of ~14,000 page requests per day. It enabled me to develop the site very quickly, was free, and holds up well. For that, I am thankful to the MonoRail team.

I just used the MonoRail bit. I chose iBATIS.NET over ActiveRecord as I had to write creative SQL to maintain compatibility with a 7-year-old turd of a database. So I can't speak for some of the other Castle libraries.

Some advantages of MonoRail include the following:

- It's pretty easy to get in there and modify things. For example, the default routing implementation didn't preserve the query string when rerouting (I needed this to preserve backwards compatibility with the old URL formats used by the now defunct classic ASP site for SEO reasons) and it didn't support issuing HTTP 301 Permanent Redirect headers for that scenario. So I implemented whatever the MonoRail interface for that was, plugged it into my config file, and off I went.
- It's still ASP.NET, so you can still use Forms Authentication, the totally awesome [HTTP/HTTPS Switcher at Codeproject](#), and caching.
- Relatively few surprises. After working with it for over year, I haven't had too many days where I've cursed at MonoRail. That's a pretty good litmus test. Some of the helper classes (FormHelper) can behave a little strangely, the wizard framework is outright bizarre, and parameter binding can sometimes throw you for a loop, but it doesn't happen often.
- Choice of View Engines (templates). I put this here because most people seem to think choice here is a good thing although I usually think that it is not.

MonoRail, however, is not without its problems:

- Lack of direction in development. The number of changes between RC2 and RC3 was beyond ridiculous; a lot of protected virtual methods disappeared, a lot of helpers changed (which is a big

deal when your view engine isn't statically typed), even the mechanism for unit testing controllers and views changed. For this reason, we're probably just going to stay on RC2 forever. Now that ASP.NET MVC is out, it's unclear how healthy the community behind MonoRail will stay (though ayende and hammett are as enthusiastic and active as ever).

- NVelocity, the "de facto" view engine for MonoRail (at least at the time that we started development), is a promising templating language with an unhealthy implementation and maintenance outlook. (Does it work? well enough. But being a CTRL+C CTRL+V port from the Java version, don't read the source for that library because your eyes will bleed.)
- NVelocity and RC2 shipped with an extremely severe threading bug where multiple users accessing the site at the same time could get served pages intended for the other. It's fixed in the latest release (which, due to the release nature of the Castle projects, is very difficult to upgrade to), and we managed to work around it. But it was a very disturbing and unexpected issue to encounter, one that would be highly unlikely to be encountered on a Microsoft framework. Caveat emptor.

MonoRail provided an excellent opportunity for us in June 2007 by giving us a way to migrate an existing site on the Microsoft stack to the .NET platform in a way that avoided WebForms (which is great for intranet sites, but not so great when you need fine-grained control of your HTML

output on a public-facing Web site, in my opinion). (Fine, the real reason is I just despise the WebForms postback model.) ASP.NET MVC was not even a gleam in Microsoft's eye at that point in time.

Now that ASP.NET MVC does exist, however, and given that Microsoft is positioning it as an alternative to WebForms, I know that I personally will strongly consider it for any future project. MonoRail is a great project, it served us well, and I'm grateful to the open source community for it, but I think of it fondly as a heavily used, worn tool that is retired to a lower drawer in my workbench. Without it, ASP.NET MVC might not exist today.

Share Improve this answer

Follow

edited Jul 13, 2012 at 7:09



adt

4,350 ● 5 ● 37 ● 55

answered Nov 19, 2008 at 5:38



Nicholas Piasecki

25.6k ● 5 ● 82 ● 92



6



Personally I have tried both ASP.NET MVC and MonoRail by CastleProject. Although I really enjoy the other CastleProject libraries, I have found that I enjoy the ASP.NET MVC implementation model better than the CastleProject MonoRail model. Now that ASP.NET MVC has released that they will be including jQuery in with the releases, I am truly excited. Ultimately, I think it depends on what other libraries you use. If you use NHibernate,

ActiveRecord, and Castle Windsor then you will probably enjoy the MonoRail libraries. If you don't use any of those libraries, or prefer the Microsoft Enterprise Libraries (currently the company standard where I work) then you will probably find the ASP.NET MVC fits your needs better. With the focus on ASP.NET MVC coming from Scott Guthrie himself I doubt it will be going away any time soon. In fact, the more people using it and signing it's praises the more likely it is to become the defacto standard.

Share Improve this answer

Follow

answered Nov 19, 2008 at 5:30



[okcodemonkey](#)

79 ● 2



1

Another alternative, with which I have no experience, is [ProMesh](#). Personally, I moving to ASP.NET MVC.

Share Improve this answer

Follow

answered Nov 19, 2008 at 4:08



[tvanfoss](#)

532k ● 102 ● 699 ● 798





One alternative that seems interesting is [MonoRail](#) although I haven't tested it out fully.

0

Share Improve this answer

Follow

answered Nov 19, 2008 at 3:39



[Jason Baker](#)

198k ● 138 ● 382 ● 520



0

ASP.NET MVC should start to become more accepted as being mature in rapid fashion. Now that it is beta, and supposedly nearly feature complete, the rate at which people adopt it will continue to grow, and likely more sharply. With the RTM/RTW release promised to be in the near future, now is the best time to start to adopt it so that you can hit the ground running with it.



If there are specific shortcomings that you see in ASP.NET MVC, you should definitely let Microsoft know about it. [Scott Guthrie](#) is very receptive to feedback and the [MVC Contrib](#) project is both open to suggestions and has a great collection of enhancements available through their library.

Share Improve this answer

Follow

answered Nov 29, 2008 at 17:46



[Joseph Ferris](#)

12.7k ● 3 ● 48 ● 73