



ASP.NET MVC 4  
Mobile Websites

**Succinctly**

by Lyle Luppés

## Chapter 1 I Love MVC 4!

"This is your last chance. After this, there is no turning back. You take the blue pill—the story ends, you wake up in your bed and believe whatever you want to believe. You take the red pill—you stay in Wonderland and I show you how deep the rabbit hole goes."

Morpheus in *The Matrix*

I've been writing websites for many years. Back in 1997, I wrote my first e-commerce website for a local printing company using Cold Fusion 1.0 and I thought that product was awesome at the time. I moved on to a company that developed a lot of websites during the infamous dot-com bubble which went bankrupt a few years later. During that time, we did a lot of Classic ASP development and I found a lot to like in that model. We alternated back and forth between ASP and Cold Fusion depending on what we needed. During those early years of websites and e-commerce 1.0, we learned a lot about how to streamline websites. With people hitting sites using a 9600 baud modem, it had to be as small and efficient as possible, and 100 KB was our upper limit for page size.

In 2002, Microsoft released ASP.NET, and developers had to learn a totally new system that was very different than the ASP we had just mastered. This new model looked a lot like the Windows Forms programming model and once we got over the learning curve, life was good for a while. We learned about the evils of the ViewState and Session management and all the things that could trip you up and bloat your page. But as connections got faster, it really didn't bother us too much that our pages started to put on a little extra weight.

One of the things that has always bothered me about ASP.NET Web Forms is the HTML it generates—have you ever taken a good look at that? Trying to trace through that code was a nightmare. When creating JavaScript, the field names and IDs were always an issue. You always had to ask yourself, "Do I have to put a 'ctl100\_' in front of this field?" But we put up with it because that's how ASP.NET worked and it was better than the alternative.

When the first versions of MVC came along, I wasn't too excited about them. It wasn't until I looked at MVC 3 and ~~the new Razor syntax that I began to get interested. As I researched it more, I learned more about the new programming model that MVC projects bring to ASP.NET. There was a style and simplicity~~ that reminded me of the good old days of Classic ASP, but with all of the new functionality and richness of a modern application.

Along with MVC 3 came the Entity Framework and Code First, and those particular ideas really resonated with me. My philosophy of programming tends to be what I refer to as "DDD"—Database-Driven Development. As I approach a project, I tend to think of things in terms of databases, models, and entities. I usually have a good idea in my head of how I can store the data for the application about the same time I start having a semi-rough screen design. MVC allowed me to follow my usual development path and opened up my development to many other possibilities. I fell in love.

I realize that MVC is not the ultimate development tool, and not everyone is going to love it. But as someone who has worked in web development in Microsoft-centered shops for many years, I have to say this is the best we've had it in a long time.