**初识ASP.NET MVC架构**

**写在前面——学习的原因**

关于B/S架构的系统一直没有正式的接触过，但知道一些对于.NET技术的B/S相关技术，刚参加工作的时候记得还都是ASP，后来有了ASP.NET、AJAX，JAVA相关的由JSP等，最近B/S的技术越来越成熟，流行的也都很多。在经过几年的C/S架构系统开发后，早就想有机会去学习一下B/S相关知识，以补偿残缺的知识结构，在经过考虑后决定去学习微软的ASP.NET MVC及其相关的HTML5、CSS、JQuery（包括JAVAScript技术）等技术。打算从MVC入手作为学习的主线，然后展开学习相关知识，MVC的学习参考书是Adam Freeman写的《精通ASP.NET MVC 4》。下面就开始介绍自己学习到的知识，后续章节会逐步添加。

**关于ASP.NET MVC**

ASP.NET MVC 是MVC模式的ASP.NET实现，它吸取了传统的ASP.NET Web From的一些缺陷，借鉴了MVC模式的一些优势。目前，具有很多优点，如：可扩展性、在HTML及HTTP方面的严密控制（如：在实现过程中不生成任何包含视图状态数据的页面，相比传统Web From页面要小很多，也减少了宽带资源的占用。）、可测试性、拥有强大的路由系统等等。另外，ASP.NET MVC是开源的，书中给出的源代码的下载地址是：http://aspnet.codeplex.com。我在GIit Hub上微软的开源代码页面页看到了相关源代码（地址：https://github.com/Microsoft，我们还可以从这上面看到除ASP.NET MVC以外的其他技术的源代码），大家可以从网上搜索一下。

MVC模式：即模型——视图——控制器（Model——View——Controller）的缩写。

* **模型（Model）**：

包含或表示用户使用的数据。可以是视图模型或域模型，视图模型只表现视图与控制器直接传递的数据；域模型包含业务领域的数据，以及处理这些数据的操作、转换和规则。我觉得可以简单的将模型理解为带有自身基本逻辑、规则以及存在于视图与控制器间直接传递的数据对象，比如可以是一些C#类型。其中域模型是MVC应用程序中最重要的部分。

* **视图（View）：**

承担将模型的某些部分渲染成用户界面（UI）。

* **控制器（Controller）：**

处理传入的请求，执行模型上的操作，并选择渲染给用户的视图。

**最后**

在我学习的过程中发现书中给出了一些MVC的约定，我打算将其单独总结到另一篇文章中（《MVC常用约定》），以便将来查看，当然在学习的过程中会逐步更新。关于ASP.NET MVC的具体技术的学习体会将在后续文章中一一介绍，在该模式下的相关的其他技术将会在相应的技术文章中进行体现，不会在ASP.NET MVC技术中做过多的阐述。

***声明：如果在本人写的学习总结中出现理解有误或书写有误的地方还希望大家能给予指出。***