

<<ASP.NET本质论>>

图书基本信息

书名：<<ASP.NET本质论>>

13位ISBN编号：9787111332855

10位ISBN编号：7111332857

出版时间：2011-3-10

出版时间：机械工业出版社华章公司

作者：郝冠军

页数：454

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<ASP.NET本质论>>

内容概要

如果你只是想系统地学习如何简单地利用asp.net快速地进行开发,《asp.net本质论》也许不是你想要的;如果你不满足于只是会利用asp.net强大的控件功能完成一些常规应用的开发,而是想深入探究asp.net的本质和精髓,实现从一个控件使用人员向系统开发人员的过渡,那么这本书是你不能错过的,也是你目前的唯一选择。

本书以asp.net应用中的请求处理过程为主线,对每一步处理所涉及的技术和原理进行了深入的剖析,同时列举了开发过程中在各处理环节可能会遇到的经典疑难问题进行了分析并给出了解决方案。

第1~4章是asp.net的核心部分,细致地剖析了asp.net中的请求处理机制、asp.net中的对象与http之间的映射关系、应用程序处理管道的处理过程、处理程序的处理机制,以及多线程技术在asp.net中的应用。

第5~8章是经典的webform部分,重点讲解了控件的原理与页面的生成机制,包括流与控件的关系、控件与页面的关系、数据绑定控件与模板的关系,以及asp.net中的各种状态管理技术。

第9章分析asp.net mvc的处理过程,以及asp.net mvc应用与经典的asp.net应用之间的关系。

第10章讨论了asp.net与iis服务器之间的关系,并分别针对不同版本的iis分析了其处理过程。

第11章对asp.net应用中的用户问题进行了分析,并就各种常见问题给出了解决方案。

<<ASP.NET本质论>>

作者简介

郝冠军，资深微软技术专家和培训讲师。
从事软件开发工作十余年，精通.NET和ASP.NET开发。
博客园和CSDN等社区的活跃技术专家，经常帮助大家解答各种疑难问题。
此外，作为微软社区精英的领导团队成员，他还负责微软社区精英的招募工作。

<<ASP.NET本质论>>

书籍目录

前言

第1章 网站应用程序

- 1.1 web应用程序的简单回顾
 - 1.1.1 资源的地址—通用资源标识符
 - 1.1.2 找到主机
 - 1.1.3 http协议
- 1.2 最简单的 web 服务器
 - 1.2.1 网络插座socket
 - 1.2.2 基于 tcplistener 的 web 服务器
 - 1.2.3 基于 httplistener 的 web 服务器
- 1.3 进入 asp.net
 - 1.3.1 web 应用程序域
 - 1.3.2 不使用gac 和bin 加载web 应用程序域
 - 1.3.3 默默无闻的工作者对象
 - 1.3.4 web 应用程序的运行时
- 1.4 对象化的http
 - 1.4.1 请求参数的对象类型 httprequest
 - 1.4.2 处理回应的对象类型 httpresponse
 - 1.4.3 辅助的常用工具类 httpserverutility
 - 1.4.4 编码与解码
 - 1.4.5 浏览器类型
- 1.5 创建自定义的 asp.net 服务器
 - 1.5.1 asp.net web 服务器
 - 1.5.2 监听程序
 - 1.5.3 部署程序集
 - 1.5.4 各种各样的cassini

1.6 本章小结

第2章 应用程序对象

- 2.1 请求的处理参数—上下文对象httpcontext
 - 2.1.1 常用成员
 - 2.1.2 底层方法
- 2.2 应用程序对象 httpapplication
 - 2.2.1 处理管道
 - 2.2.2 httpapplication 的处理管道
 - 2.2.3 处理过程的简单介绍
 - 2.2.4 httpcontext状态管理
- 2.3 处理 httpapplication 的事件
 - 2.3.1 通过 ihttpmodule 创建 httpapplication 的事件处理程序
 - 2.3.2 注册 httpmodule
 - 2.3.3 不使用配置文件注册 httpmodule
 - 2.3.4 常见的 httpmodule
 - 2.3.5 httpmodule的事件
 - 2.3.6 通过 global.asax 创建 httpapplication 的事件处理程序
 - 2.3.7 global.asax 中 httpapplication 事件的自动注册
 - 2.3.8 特殊的httpapplication 事件处理

<<ASP.NET本质论>>

- 2.4 两个特殊的事件
 - 2.5 大文件上传问题
 - 2.5.1 文件上传的规范
 - 2.5.2 asp.net 中的文件上传
 - 2.5.3 文件上传的解决方案
 - 2.5.4 通过httpmodule 接管请求参数
 - 2.5.5 自定义的请求参数对象
 - 2.5.6 读取上传数据的接口和实现
 - 2.5.7 读取上传数据流
 - 2.5.8 注册自定义的上传管理
 - 2.5.9 使用自定义的上传管理
 - 2.6 各种各样的文件上传
 - 2.6.1 无刷新的上传：jquery form
 - 2.6.2 基于客户端技术的上传进度：swfupload
 - 2.7 本章小结
- 第3章 http请求处理程序
- 3.1 处理程序
 - 3.1.1 处理程序与 httpapplication 的关系
 - 3.1.2 处理程序接口 ihttphandler 和 ihttpasynchandler
 - 3.1.3 在处理程序中使用会话
 - 3.1.4 处理程序工厂
 - 3.1.5 注册处理程序
 - 3.1.6 使用处理程序生成验证码
 - 3.2 一般处理程序
 - 3.2.1 一般处理程序工厂
 - 3.2.2 使用一般处理程序的场合
 - 3.2.3 使用一般处理程序生成验证码图片
 - 3.2.4 使用一般处理程序生成 json
 - 3.3 页面处理程序
 - 3.3.1 页面处理程序工厂
 - 3.3.2 创建页面处理程序
 - 3.3.3 生成的代码
 - 3.3.4 使用页面处理程序
 - 3.4 web 服务处理程序
 - 3.4.1 web 服务处理程序工厂
 - 3.4.2 使用 web 服务处理程序
 - 3.4.3 web 服务的常用标签
 - 3.4.4 派生自 system.web.services.webservice 类的意义
 - 3.5 mvc 处理程序
 - 3.5.1 mvc 的路由接口 iroutehandler
 - 3.5.2 自定义的 iroutehandler
 - 3.5.3 注册路由处理程序
 - 3.5.4 获取控制器的工厂接口 icontrollerfactory
 - 3.5.5 mvc 请求的处理过程
 - 3.6 资源处理程序
 - 3.6.1 资源的处理程序配置
 - 3.6.2 定义嵌入的资源

<<ASP.NET本质论>>

- 3.6.3 获取资源的地址
 - 3.6.4 使用嵌入的资源
 - 3.7 禁止的处理程序
 - 3.7.1 配置禁止访问的资源
 - 3.7.2 禁止访问 excel
 - 3.8 虚拟路径提供器
 - 3.8.1 定义虚拟路径提供器
 - 3.8.2 注册虚拟路径提供器
 - 3.8.3 压缩文件中的网站
 - 3.8.4 sharpziplib
 - 3.9 本章小结
- 第4章 asp.net中的线程与异步
- 4.1 线程基础
 - 4.1.1 线程
 - 4.1.2 自定义线程
 - 4.1.3 前台线程和后台线程
 - 4.1.4 工作者线程和 i o 线程
 - 4.1.5 线程池
 - 4.2 .net 中线程处理
 - 4.2.1 线程的创建与启动
 - 4.2.2 线程的状态
 - 4.2.3 线程的执行上下文
 - 4.2.4 异步编程模式apm
 - 4.2.5 基于事件的异步编程模式 epm
 - 4.2.6 异步线程的状态与同步问题
 - 4.2.7 处理管道中的异步问题
 - 4.3 线程池
 - 4.3.1 线程池的工作原理
 - 4.3.2 将工作者线程加入线程池
 - 4.3.3 将i o 线程加入线程池
 - 4.4 httpapplication 中的异步线程
 - 4.4.1 asp.net 中的线程池设置
 - 4.4.2 异步步骤中的异步点
 - 4.4.3 启动和完成异步步骤
 - 4.5 异步处理程序
 - 4.5.1 异步处理程序接口
 - 4.5.2 在处理程序中异步调用 web 服务
 - 4.6 异步页面
 - 4.6.1 页面异步任务的启动和完成
 - 4.6.2 异步页面任务
 - 4.6.3 异步页面中访问 web 服务三种方式
 - 4.6.4 实例—查询 qq 在线状态
 - 4.7 本章小结
- 第5章 页面即对象
- 5.1 流动的网页
 - 5.1.1 字节流
 - 5.1.2 字符编码

<<ASP.NET本质论>>

- 5.1.3 字符流
 - 5.1.4 回应对象中的流
 - 5.1.5 专门输出 html 的字符流
 - 5.2 控件—页面对象的基石
 - 5.2.1 控件类
 - 5.2.2 render和rendercontrol
 - 5.2.3 控件基类
 - 5.2.4 组合模式 composite
 - 5.2.5 id 是一个问题
 - 5.3 形形色色的控件
 - 5.3.1 html 控件
 - 5.3.2 web 控件
 - 5.3.3 webpart 控件
 - 5.4 控件实现的常用接口
 - 5.4.1 生成和回发
 - 5.4.2 控件的任意属性iattributeaccessor
 - 5.4.3 数据的回发ipostbackdatahandler
 - 5.4.4 回发服务器端事件ipostbackeventhandler
 - 5.5 页面
 - 5.5.1 页面与模板
 - 5.5.2 母版页
 - 5.5.3 页面就是一个处理程序
 - 5.5.4 页面的事件处理管道
 - 5.5.5 处理页面的事件
 - 5.6 生成的过程
 - 5.6.1 从模板到对象模型—buildprovider
 - 5.6.2 从标记到控件—controlbuilder
 - 5.6.3 进入生成阶段—controladapter
 - 5.6.4 控件适配器—controladapter
 - 5.6.5 web控件适配器—webcontroladapter
 - 5.6.6 页面适配器—pageadapter
 - 5.6.7 使用 adapter 定制表单的action
 - 5.7 自定义的 url 重写
 - 5.7.1 url 重写的原理
 - 5.7.2 使用 httpmodule 实现url 重写
 - 5.7.3 在配置文件中处理重写映射
 - 5.7.4 无扩展名请求的处理问题
 - 5.8 本章小结
- 第6章 状态
- 6.1 基本状态管理
 - 6.1.1 隐藏域
 - 6.1.2 cookie
 - 6.1.3 url
 - 6.2 视图状态viewstate
 - 6.2.1 序列化和反序列化
 - 6.2.2 控制序列化
 - 6.2.3 base64

<<ASP.NET本质论>>

- 6.2.4 视图状态属性与 istatemanager 接口
 - 6.2.5 保存和恢复的时间点
 - 6.2.6 视图状态的序列化器
 - 6.2.7 使用视图状态实现路径导航
 - 6.3 控件状态 controlstate
 - 6.4 应用程序状态 application
 - 6.5 会话状态 session
 - 6.5.1 服务器端的 session
 - 6.5.2 客户端的 sessionid
 - 6.5.3 session 保存的位置
 - 6.5.4 session 的过期问题
 - 6.5.5 压缩 session 数据
 - 6.6 httpcontext 状态
 - 6.7 cache
 - 6.7.1 缓存的原理
 - 6.7.2 .net 中的缓存管理实现
 - 6.7.3 基于文件的缓存依赖
 - 6.7.4 基于sql的缓存依赖
 - 6.7.5 组合的缓存依赖
 - 6.7.6 删除所有的缓存项目
 - 6.7.7 web 服务器端的页面缓存
 - 6.7.8 页面局部缓存
 - 6.7.9 自定义的输出缓存提供者
 - 6.8 memcached
 - 6.8.1 下载和安装 memcached
 - 6.8.2 在 asp.net 中访问 memcached
 - 6.9 统计当前在线用户
 - 6.9.1 module 的处理
 - 6.9.2 注册 module
 - 6.9.3 module 的配置参数
 - 6.9.4 module 的事件处理
 - 6.10 本章小结
- 第7章 模板和数据绑定
- 7.1 页面与绑定
 - 7.1.1 嵌入式代码块和表达式
 - 7.1.2 绑定表达式
 - 7.1.3 目标target
 - 7.1.4 容器container
 - 7.1.5 触发绑定事件的方法 databind
 - 7.2 控件内的模板
 - 7.2.1 基于模板的控件
 - 7.2.2 控件模板中的 container
 - 7.2.3 databinder
 - 7.2.4 page 中的 eval
 - 7.2.5 在属性中使用绑定表达式
 - 7.3 repeater控件
 - 7.3.1 数据的来源 datasource

<<ASP.NET本质论>>

- 7.3.2 repeater 的基石—repeateritem
- 7.3.3 绑定的过程
- 7.3.4 绑定中的事件
- 7.3.5 绑定的结果：controls 集合和 items 集合
- 7.3.6 回发中的 itemcommand 事件
- 7.4 高级数据控件
 - 7.4.1 唯一支持分栏的控件-datalist
 - 7.4.2 gridview
 - 7.4.3 listview 和 datapager
- 7.5 数据源控件
 - 7.5.1 反射
 - 7.5.2 两种数据源
 - 7.5.3 对象数据源
 - 7.5.4 业务对象的标签
 - 7.5.5 页面控件与数据源控件之间的关系
 - 7.5.6 数据源控件相关的事件点
 - 7.5.7 基于数据源控件的分页
- 7.6 本章小结
- 第8章 自定义控件
 - 8.1 自定义控件的继承体系
 - 8.2 自定义控件涉及的相关类型
 - 8.3 自定义的带有上传进度的按钮
 - 8.3.1 控件的工作原理
 - 8.3.2 选择控件的基类
 - 8.3.3 自定义的数据类型
 - 8.3.4 状态的持久化
 - 8.3.5 控件的属性
 - 8.3.6 属性转换问题—typeconverter
 - 8.3.7 编辑属性数据 uitypeditor
 - 8.3.8 保存在 aspx 中
 - 8.3.9 设计器中的显示效果
 - 8.3.10 工具栏中控件的图标
 - 8.3.11 脚本嵌入和使用
 - 8.3.12 控件的呈现
 - 8.3.13 使用自定义控件
 - 8.4 petshop 中的自定义控件
 - 8.4.1 基类
 - 8.4.2 表格
 - 8.4.3 处理当前页码参数
 - 8.4.4 datasource 属性
 - 8.4.5 事件
 - 8.4.6 生成
 - 8.5 本章小结
- 第9章 mvc
 - 9.1 asp.net mvc 是表现层的mvc
 - 9.2 在 httpapplication 中的asp.net mvc
 - 9.2.1 创建 routetable

<<ASP.NET本质论>>

- 9.2.2 urlroutingmodule事件处理
 - 9.3 从url进入mvc之门
 - 9.3.1 有意义的 url
 - 9.3.2 在 iis 6.0 和 iis 7 中的配置
 - 9.3.3 从url到route
 - 9.3.4 约束
 - 9.3.5 routing
 - 9.3.6 requestcontext 的前世今生
 - 9.3.7 在 asp.net mvc 中防盗链
 - 9.4 控制器
 - 9.4.1 控制器工厂
 - 9.4.2 使用自定义的控制器工厂
 - 9.4.3 为 controller 类传递构造函数的参数
 - 9.4.4 controller 的继承关系
 - 9.4.5 controller 中的状态管理
 - 9.4.6 基于过滤器的扩展
 - 9.4.7 选择 action
 - 9.5 模型
 - 9.5.1 绑定 model
 - 9.5.2 简单参数和复杂参数
 - 9.5.3 模型对象的元数据
 - 9.5.4 model 的验证
 - 9.5.5 自定义model 的验证
 - 9.6 执行 action
 - 9.6.1 各种actionresult
 - 9.6.2 向视图传递数据
 - 9.7 视图
 - 9.7.1 视图引擎
 - 9.7.2 经典视图—viewpage
 - 9.7.3 视图引擎—razor
 - 9.8 本章小结
- 第10章 iis 与 asp.net
- 10.1 网站
 - 10.1.1 绑定
 - 10.1.2 网站应用程序
 - 10.1.3 虚拟目录
 - 10.2 通过 isapi 扩展 iis
 - 10.2.1 isapi 扩展
 - 10.2.2 isapi 过滤器
 - 10.2.3 clr 是一个 com 组件
 - 10.2.4 asp.net 中的isapi 扩展和过滤器
 - 10.2.5 isapi rewrite
 - 10.3 iis 与 asp.net
 - 10.3.1 iis5 与 asp.net
 - 10.3.2 iis6 与 asp.net
 - 10.3.3 iis7 与 asp.net
 - 10.4 创建网站的两种方法

<<ASP.NET本质论>>

- 10.5 asp.net中的加密与解密
 - 10.5.1 machinekey
 - 10.5.2 加密服务
 - 10.5.3 配置节的加密和解密
- 10.6 本章小结
- 第11章 asp.net中的用户
 - 11.1 从iis开始
 - 11.1.1 匿名用户方式
 - 11.1.2 基本身份验证
 - 11.1.3 摘要式身份验证
 - 11.1.4 集成 windows 身份验证
 - 11.1.5 .net passport 身份验证
 - 11.2 asp.net中的用户信息
 - 11.2.1 基于 windows 验证的用户
 - 11.2.2 基于 forms 验证的用户
 - 11.2.3 基于 passport 验证的用户
 - 11.2.4 在 iis7 中使用表单验证
 - 11.3 .net 中的用户
 - 11.3.1 用户的标识 iidentity
 - 11.3.2 用户 iprincipal
 - 11.4 网站中的用户
 - 11.5 成员管理
 - 11.5.1 用户的基本信息
 - 11.5.2 成员管理的约定
 - 11.5.3 基于 sqlserver 的成员管理实现
 - 11.5.4 自定义的成员管理实现
 - 11.5.5 辅助工具类 membership
 - 11.6 用户的扩展信息—个性化数据
 - 11.6.1 个性化数据的约定
 - 11.6.2 个性化数据的属性
 - 11.6.3 实现自定义的个性化数据管理
 - 11.6.4 匿名的个性化数据
 - 11.6.5 合并匿名用户的个性化数据
 - 11.6.6 基于 sql server 的个性化数据管理
 - 11.6.7 管理个性化数据
 - 11.7 执行程序的用户
 - 11.7.1 执行网站程序的 windows 用户
 - 11.7.2 用户模拟的作用
 - 11.7.3 数据库连接串中的用户
 - 11.8 本章小结
- 附录 自定义配置参数

<<ASP.NET本质论>>

章节摘录

版权页：插图：

<<ASP.NET本质论>>

媒体关注与评论

这是一本非常好的ASP.NET学习指南和参考资料，它直指ASP.NET的技术核心，理论知识与开发实例并进，详尽地分析了ASP.NET开发过程中所涉及的重要方面，让读者懂得如何最大限度地利用ASP.NET提供的丰富功能。

对于所有ASP.NET开发者来说，这是一本不可或缺的好书。

——周银辉 资深微软技术专家 \ MVP (2008) 收到本书的初稿时，我正好在呼和浩特出差，这本书便成为了我旅途中形影不离的伙伴。

本书不仅在广度上包含了ASP.NET相关的重要知识点，而且在深度上更是透彻地分析了各个知识点的原理、要点和难点，总结了大量开发者们很可能在开发过程中遇到的问题，并给予了很好的解决方案，足见作者具有丰富的开发经验。

本书不仅是中初级开发者提升ASP.NET功力不可多得的一本好书，而且也是高级开发者案头必备的参考手册，它收录了大量的开发技巧和最佳实践，可供开发过程中查阅，对于开发稳定而高效WASP.NET应用极有帮助。

——冯瑞涛 微软MSDN中文技术论坛ASP.NET&Ajax和ADO.NET&LINQ版块版主 \ 微软.NET俱乐部活动组织者会用ASP.NET的程序员很多，但是ASP.NET方面的开发高手着实不多，而在这些高手中，能用深入浅出的语言阐述.NET本质的人更是少之又少了。

幸运的是，这本书做到了，它用极为朴实的语言，结合精心设计的案例，将ASP.NET的本质以及开发中的重点和难点娓娓道来，细致入微，是深入学习ASP.NET不可多得的经典之作。

——王晓光 微软金牌讲师 \ Mc.r \ 达内集团技术支持总监 本书是作者在ASP.NET领域多年开发和教学工作经验的结晶。

作为一本面向有一定开发经验的ASP.NET开发人员的书籍，本书不但对ASP.NET的内在运行机制进行了深入地阐述，而且对ASP.NET的组件机制进行了入微地分析。

这不仅是一本进阶指导书，能帮助ASP.NET开发者进一步了解ASP.NET的精髓和本质；而且这也是一本开发参考手册，本书中有大量实例、解决方案和最佳实践供开发者查阅和参考。 ——葛涵涛 微软(北京).NET俱乐部负责人 \ windows Phone MVP

<<ASP.NET本质论>>

编辑推荐

《ASP.NET本质论》：以最新版ASP.NET为基础，全面深入剖析ASP.NET的本质资深ASP.NET专家执笔，微软技术社和MVP联袂推荐。

ASP.NET通过一整套封装了底层处理机制的类库来提供极其高效的开发环境，使得许多开发任务只要通过简单的控件拖曳就可以实现。

但是，在这强大功能的背后，ASP.NET隐藏了高度复杂的实现机制和工作原理，使得开发者不必去关心ASP.NET底层的问题。

因为如此，大量的ASP.NET程序员，包括许多有经验的开发者，都被简单的拖曳遮蔽了视线，几乎所有的ASP.NET图书也都以控件的使用方法作为主要内容，使得很多ASP.NET程序员仅浮于技术表面，不能深入理解和掌握ASP.NET的本质和精髓，只能借助现有的控件来完成任务。

对于开发中遇到的难题，不能突破控件的局限进行深入地分析，如同雾里看花，只能束手无策。

最后导致软件质量难以保证，开发工期无法控制。

《程序员修炼之道：从小工到专家》中有一句名言：“如果你不能了解生成的代码，那就不要使用代码生成器。

”这句话同样适用于ASP.NET的学习，学习ASP.NET也需要从底层的核心原理开始。

就像看一场魔术表演，舞台上的表演千变万化，但是基本的原理却万变不离其宗，当我们掌握了这个“宗”，控件的使用也就水到渠成，ASP.NET的不断升级和演变就不再会是学习的负担，而是不断提供给程序员的神兵利器。

<<ASP.NET本质论>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>