**ASP.NET 6框架揭秘**

[第1章 编程体验 1](#_Toc93310789)

[1.1 控制台程序 1](#_Toc93310790)

[1.1.1 构建开发环境 1](#_Toc93310791)

[1.1.2　命令行构建 .NET应用 2](#_Toc93310792)

[1.2 ASP.NET应用 6](#_Toc93310793)

[1.2.1 launchSettings.json 7](#_Toc93310794)

[1.2.2 Minimal API 9](#_Toc93310795)

[1.2.3 中间件 10](#_Toc93310796)

[1.2.4 配置选项 14](#_Toc93310797)

[1.2.5 诊断日志 16](#_Toc93310798)

[1.2.6 路由 17](#_Toc93310799)

[1.3 MVC 18](#_Toc93310800)

[1.3.1 定义Controller 18](#_Toc93310801)

[1.3.2 引入视图 19](#_Toc93310802)

[1.4 gRPC 20](#_Toc93310803)

[1.4.1 定义服务 20](#_Toc93310804)

[1.4.2 实现和承载 22](#_Toc93310805)

[1.4.3 调用服务 24](#_Toc93310806)

[1.5 Dapr 25](#_Toc93310807)

[1.5.1 构建开发环境 25](#_Toc93310808)

[1.5.2 服务调用 26](#_Toc93310809)

[1.5.3 状态管理 29](#_Toc93310810)

[1.5.4 发布订阅 32](#_Toc93310811)

[1.5.5 Actor模型 34](#_Toc93310812)

[第2章 依赖注入（上篇） 37](#_Toc93310813)

[2.1 控制反转 37](#_Toc93310814)

[2.1.1 流程控制的反转 37](#_Toc93310815)

[2.1.2 好莱坞法则 39](#_Toc93310816)

[2.1.3 流程定制 40](#_Toc93310817)

[2.2 IoC模式 40](#_Toc93310818)

[2.2.1 模板方法 40](#_Toc93310819)

[2.2.2 工厂方法 41](#_Toc93310820)

[3.2.3　抽象工厂 43](#_Toc93310821)

[2.3 依赖注入 44](#_Toc93310822)

[2.3.1 由容器提供对象 45](#_Toc93310823)

[2.3.2 三种注入方式 46](#_Toc93310824)

[2.3.3 Service Locator模式 48](#_Toc93310825)

[2.4 一个简易版的依赖注入容器 49](#_Toc93310826)

[2.4.1 编程体验 49](#_Toc93310827)

[3.4.2　设计与实现 53](#_Toc93310828)

[3.4.3　扩展方法 58](#_Toc93310829)

[第3章 依赖注入（下篇） 62](#_Toc93310830)

[3.1 利用容器提供服务 62](#_Toc93310831)

[3.1.1 服务的注册与消费 62](#_Toc93310832)

[3.1.2 生命周期 64](#_Toc93310833)

[3.1.3 服务注册的验证 66](#_Toc93310834)

[3.2 服务注册 69](#_Toc93310835)

[3.2.1 ServiceDescriptor 69](#_Toc93310836)

[3.2.2 IServiceCollection 70](#_Toc93310837)

[3.3 服务的消费 73](#_Toc93310838)

[3.3.1 IServiceProvider 74](#_Toc93310839)

[3.3.2 服务实例的创建 75](#_Toc93310840)

[3.3.3 生命周期 76](#_Toc93310841)

[3.3.4 ActivatorUtilities 81](#_Toc93310842)

[3.4 扩展 85](#_Toc93310843)

[3.4.1. 适配 85](#_Toc93310844)

[3.4.2. IServiceProviderFactory<TContainerBuilder> 86](#_Toc93310845)

[3.4.3. 整合第三方依赖注入框架 86](#_Toc93310846)

[第4章 文件系统 91](#_Toc93310847)

[4.1 抽象的文件系统 91](#_Toc93310848)

[4.1.1 树形层次结构 91](#_Toc93310849)

[4.1.2 读取文件内容 93](#_Toc93310850)

[4.1.3 监控文件的变化 95](#_Toc93310851)

[4.2 文件与目录 95](#_Toc93310852)

[4.2.1 IChangeToken 96](#_Toc93310853)

[4.2.2 IFileProvider 97](#_Toc93310854)

[4.2.3 两个特殊的文件系统 99](#_Toc93310855)

[4.3 物理文件系统 101](#_Toc93310856)

[4.3.1 PhysicalFileInfo 101](#_Toc93310857)

[4.3.2 PhysicalDirectoryInfo 102](#_Toc93310858)

[4.3.3 PhysicalDirectoryContents 102](#_Toc93310859)

[4.3.4 NotFoundDirectoryContents 102](#_Toc93310860)

[4.3.5 PhysicalFilesWatcher 103](#_Toc93310861)

[4.4 内嵌文件系统 104](#_Toc93310862)

[4.4.1 将项目文件变成内嵌资源 104](#_Toc93310863)

[4.4.2 读取资源文件 105](#_Toc93310864)

[4.4.3 EmbeddedFileProvider 106](#_Toc93310865)

[第5章 配置选项（上篇） 109](#_Toc93310866)

[5.1 读取配置信息 109](#_Toc93310867)

[5.1.1 编程模型三要素 109](#_Toc93310868)

[5.1.2 以键值对的形式读取配置 109](#_Toc93310869)

[5.1.3 读取结构化的配置 111](#_Toc93310870)

[5.1.4 将结构化配置直接绑定为对象 113](#_Toc93310871)

[5.1.5 将配置定义在文件中 113](#_Toc93310872)

[5.1.6 根据环境动态加载配置文件 114](#_Toc93310873)

[5.1.7 配置内容的同步 116](#_Toc93310874)

[5.2 配置模型 117](#_Toc93310875)

[5.2.1 数据结构及其转换 117](#_Toc93310876)

[5.2.2 IConfiguration 118](#_Toc93310877)

[5.2.3 IConfigurationProvider 120](#_Toc93310878)

[5.2.4 IConfigurationSource 121](#_Toc93310879)

[5.2.5 IConfigurationBuilder 121](#_Toc93310880)

[5.2.6 ConfigurationManager 122](#_Toc93310881)

[5.3 配置绑定 123](#_Toc93310882)

[5.3.1 绑定配置项的值 123](#_Toc93310883)

[5.3.2 绑定复合对象 125](#_Toc93310884)

[5.3.3 绑定集合 126](#_Toc93310885)

[5.3.4 绑定字典 129](#_Toc93310886)

[5.4 配置的同步 129](#_Toc93310887)

[5.4.1 配置数据流 130](#_Toc93310888)

[5.4.2 ConfigurationReloadToken 130](#_Toc93310889)

[5.4.3 ConfigurationRoot 131](#_Toc93310890)

[5.4.4 ConfigurationSection 132](#_Toc93310891)

[5.5 多样性的配置源 133](#_Toc93310892)

[5.5.1 MemoryConfigurationSource 133](#_Toc93310893)

[5.5.2 EnvironmentVariablesConfigurationSource 134](#_Toc93310894)

[5.5.3 CommandLineConfigurationSource 136](#_Toc93310895)

[5.5.4 FileConfigurationSource 139](#_Toc93310896)

[5.5.5 StreamConfigurationSource 147](#_Toc93310897)

[5.5.6 ChainedConfigurationSource 148](#_Toc93310898)

[第6章 配置选项（下篇） 151](#_Toc93310899)

[6.1 Options模式 151](#_Toc93310900)

[6.1.1 将配置绑定为Options对象 151](#_Toc93310901)

[6.1.2 提供具名的Options 152](#_Toc93310902)

[6.1.3 配置源的同步 153](#_Toc93310903)

[6.1.4 直接初始化Options对象 155](#_Toc93310904)

[6.1.5 根据依赖服务的Options设置 157](#_Toc93310905)

[6.1.6 验证Options的有效性 158](#_Toc93310906)

[6.2 Options模型 159](#_Toc93310907)

[6.2.1 OptionsManager<TOptions> 159](#_Toc93310908)

[6.2.2 IOptionsFactory<TOptions> 161](#_Toc93310909)

[6.2.3 IOptionsMonitorCache<TOptions> 167](#_Toc93310910)

[6.2.4 IOptionsMonitor<TOptions> 168](#_Toc93310911)

[6.3 依赖注入 170](#_Toc93310912)

[6.3.1 服务注册 170](#_Toc93310913)

[6.3.2 IOptions<TOptions>与IOptionsSnapshot<TOptions> 175](#_Toc93310914)

[6.3.3 集成配置系统 177](#_Toc93310915)

[第7章 诊断日志（上篇） 179](#_Toc93310916)

[7.1 各种诊断日志形式 179](#_Toc93310917)

[7.1.1 调试日志 179](#_Toc93310918)

[7.1.2 跟踪日志 180](#_Toc93310919)

[7.1.3 事件日志 181](#_Toc93310920)

[8.1.4　诊断日志 184](#_Toc93310921)

[7.2 Debugger调试日志 187](#_Toc93310922)

[7.2.1 Debugger 187](#_Toc93310923)

[7.2.2 Debug 188](#_Toc93310924)

[7.3 TraceSource跟踪日志 189](#_Toc93310925)

[7.3.1 跟踪日志模型三要素 189](#_Toc93310926)

[7.3.2 预定义TraceListener 196](#_Toc93310927)

[7.3.3 Trace 199](#_Toc93310928)

[7.4 EventSource事件日志 202](#_Toc93310929)

[7.4.1 EventSource 202](#_Toc93310930)

[7.4.2 EventListener 207](#_Toc93310931)

[7.4.3 活动跟踪 210](#_Toc93310932)

[7.5 DiagnosticSource诊断日志 213](#_Toc93310933)

[7.5.1 标准的观察者模式 213](#_Toc93310934)

[7.5.2 AnonymousObserver<T> 215](#_Toc93310935)

[7.5.3 强类型的事件订阅 218](#_Toc93310936)

[7.5.4 针对活动的跟踪 219](#_Toc93310937)

[第8章 诊断日志（中篇） 221](#_Toc93310938)

[8.1 统一日志编程模式 221](#_Toc93310939)

[8.1.1 日志输出 221](#_Toc93310940)

[8.1.2 日志过滤 225](#_Toc93310941)

[8.1.3 日志范围 230](#_Toc93310942)

[8.1.4 LoggerMessage 231](#_Toc93310943)

[8.2 日志模型详解 233](#_Toc93310944)

[8.2.1 日志模型三要素 233](#_Toc93310945)

[8.2.2 ILogger 233](#_Toc93310946)

[8.2.3 ILoggerProvider 235](#_Toc93310947)

[8.2.4 ILoggerFactory 235](#_Toc93310948)

[8.2.5 LoggerMessage 238](#_Toc93310949)

[8.3 日志范围 240](#_Toc93310950)

[8.3.1 调用链跟踪 240](#_Toc93310951)

[8.3.2 服务范围堆栈 249](#_Toc93310952)

[8.3.3 Activity的应用 251](#_Toc93310953)

[8.4 依赖注入 252](#_Toc93310954)

[8.4.1 核心服务 252](#_Toc93310955)

[8.4.2 配置 254](#_Toc93310956)

[8.4.3 日志过滤规则 257](#_Toc93310957)

[第9章 诊断日志（下篇） 259](#_Toc93310958)

[9.1 控制台 259](#_Toc93310959)

[9.1.1 ConsoleFormatter 259](#_Toc93310960)

[9.1.2 ConsoleLogger 265](#_Toc93310961)

[9.1.3 ConsoleLoggerProvider 267](#_Toc93310962)

[9.1.4 服务注册 268](#_Toc93310963)

[9.2 调试器 272](#_Toc93310964)

[9.2.1 DebugLogger 272](#_Toc93310965)

[9.2.2 DebugLoggerProvider 273](#_Toc93310966)

[9.3 TraceSource日志 274](#_Toc93310967)

[9.3.1 TraceSourceLogger 274](#_Toc93310968)

[9.3.2 TraceSourceLoggerProvider 276](#_Toc93310969)

[9.4 EventSource日志 277](#_Toc93310970)

[9.4.1 LoggingEventSource 277](#_Toc93310971)

[9.4.2 EventSourceLogger 278](#_Toc93310972)

[9.4.3 EventSourceLoggerProvider 281](#_Toc93310973)

[9.4.4 日志范围 282](#_Toc93310974)

[第10章 对象池 287](#_Toc93310975)

[13.1 利用对象池复用对象 287](#_Toc93310976)

[13.1.1 对象的“借”与“还” 287](#_Toc93310977)

[13.1.2 依赖注入 288](#_Toc93310978)

[13.1.3 池化对象策略 289](#_Toc93310979)

[13.1.4 对象池的大小 290](#_Toc93310980)

[13.1.5 对象的释放 291](#_Toc93310981)

[13.2 池化对象管理 293](#_Toc93310982)

[10.2.1 IPooledObjectPolicy<T> 293](#_Toc93310983)

[10.2.2 ObjectPool<T> 293](#_Toc93310984)

[10.2.3 ObjectPoolProvider 298](#_Toc93310985)

[13.3 扩展应用 299](#_Toc93310986)

[10.3.1 池化集合 299](#_Toc93310987)

[10.3.2 池化StringBuilder 301](#_Toc93310988)

[10.3.3 ArrayPool<T> 302](#_Toc93310989)

[10.3.4 MemoryPool<T> 303](#_Toc93310990)

[第11章 缓存 305](#_Toc93310991)

[11.1 将数据缓存起来 305](#_Toc93310992)

[11.1.1 将数据缓存在内存中 305](#_Toc93310993)

[11.1.2 将数据缓存在Redis中 306](#_Toc93310994)

[11.1.3 将数据缓存在SQL Server中 309](#_Toc93310995)

[11.2 本地内存缓存 310](#_Toc93310996)

[11.2.1 ICacheEntry 311](#_Toc93310997)

[11.2.2 MemoryCacheEntryOptions 313](#_Toc93310998)

[11.2.3 IMemoryCache 314](#_Toc93310999)

[11.3 分布式缓存 321](#_Toc93311000)

[11.3.1 IDistributedCache 321](#_Toc93311001)

[11.3.2 Redis缓存 322](#_Toc93311002)

[11.3.3 SQL Server缓存 324](#_Toc93311003)

[第12章 HTTP调用 327](#_Toc93311004)

[12.1 HttpClient的工厂 327](#_Toc93311005)

[12.1.1 手工创建HttpClient 327](#_Toc93311006)

[12.1.2 使用IHttpClientFactory工厂 328](#_Toc93311007)

[12.1.3 直接注入HttpClient 329](#_Toc93311008)

[12.1.4 定制HttpClient 329](#_Toc93311009)

[12.1.5 强类型客户端 330](#_Toc93311010)

[12.1.6 失败重试 331](#_Toc93311011)

[12.2 HttpMessageHandler管道 332](#_Toc93311012)

[12.2.1. HttpMessageHandler 332](#_Toc93311013)

[12.2.2. DelegatingHandler 333](#_Toc93311014)

[12.2.3. 诊断日志 336](#_Toc93311015)

[12.2.4. 复用HttpClientHandler 338](#_Toc93311016)

[12.3 HttpClient的构建 339](#_Toc93311017)

[12.3.1. HttpMessageHandlerBuilder 340](#_Toc93311018)

[12.3.2. HttpClientFactoryOptions 341](#_Toc93311019)

[12.3.3. IHttpMessageHandlerBuilderFilter 346](#_Toc93311020)

[12.3.4. IHttpClientFactory 347](#_Toc93311021)

[12.4 依赖注入 355](#_Toc93311022)

[12.4.1. 基础服务注册 355](#_Toc93311023)

[12.4.2. 定制HttpClient 357](#_Toc93311024)

[12.4.3. 强类型客户端 360](#_Toc93311025)

[第13章 数据保护 365](#_Toc93311026)

[13.1 加密与哈希 365](#_Toc93311027)

[13.1.6 数据加密与解密 365](#_Toc93311028)

[13.1.7 设置加密内容的有效期 367](#_Toc93311029)

[13.1.8 撤销秘钥 368](#_Toc93311030)

[13.1.9 “瞬时”加解密 369](#_Toc93311031)

[13.1.10 密码哈希 370](#_Toc93311032)

[13.2 加密模型 372](#_Toc93311033)

[13.2.1. IAuthenticatedEncryptor 372](#_Toc93311034)

[13.2.2. IKey 372](#_Toc93311035)

[13.2.3. IKeyRing 373](#_Toc93311036)

[13.2.4. IKeyRingProvider 375](#_Toc93311037)

[13.2.5. IDataProtector 376](#_Toc93311038)

[13.3 密钥管理 383](#_Toc93311039)

[13.3.1. KeyManagementOptions 383](#_Toc93311040)

[13.3.2. Key 387](#_Toc93311041)

[13.3.3. IKeyManager 389](#_Toc93311042)

[13.3.4. KeyRingProvider 394](#_Toc93311043)

[13.4 依赖注入 400](#_Toc93311044)

[13.4.1. 注册基础服务 400](#_Toc93311045)

[13.4.2. 密钥管理配置 404](#_Toc93311046)

[13.4.3. 扩展配置 405](#_Toc93311047)

[第14章 服务承载 409](#_Toc93311048)

[14.1 服务承载 409](#_Toc93311049)

[14.1.1 性能指标收集服务 409](#_Toc93311050)

[14.1.2 依赖注入 410](#_Toc93311051)

[14.1.3 配置选项 413](#_Toc93311052)

[14.1.4 承载环境 415](#_Toc93311053)

[10.1.5　日志 417](#_Toc93311054)

[14.2 承载模型 419](#_Toc93311055)

[14.2.1 IHostedService 419](#_Toc93311056)

[14.2.2 IHost 420](#_Toc93311057)

[14.2.3 IHostBuilder 424](#_Toc93311058)

[14.3 承载流程 429](#_Toc93311059)

[14.3.1 服务宿主 429](#_Toc93311060)

[14.3.2 承载设置 432](#_Toc93311061)

[14.3.3 创建宿主 437](#_Toc93311062)

[14.3.4 静态类型Host 442](#_Toc93311063)

[第15章 管道（上篇） 445](#_Toc93311064)

[15.1 管道式的请求处理 445](#_Toc93311065)

[15.1.1 承载方式的变迁 445](#_Toc93311066)

[15.1.2 中间件 449](#_Toc93311067)

[15.1.3 定义强类型中间件 452](#_Toc93311068)

[15.1.4 按照约定定义中间件 453](#_Toc93311069)

[15.2 依赖注入 454](#_Toc93311070)

[15.2.1 服务注册 454](#_Toc93311071)

[15.2.2 服务注入 455](#_Toc93311072)

[15.2.3 生命周期 457](#_Toc93311073)

[15.3 配置 461](#_Toc93311074)

[15.3.1 初始化配置 462](#_Toc93311075)

[15.3.2 以键值对形式读取和修改配置 462](#_Toc93311076)

[15.3.3 注册配置源 463](#_Toc93311077)

[15.4 承载环境 464](#_Toc93311078)

[15.4.1 IWebHostEnvironment 464](#_Toc93311079)

[15.4.2 通过配置定制承载环境 465](#_Toc93311080)

[15.4.3 设置监听地址 468](#_Toc93311081)

[15.4.4 针对环境的编程 469](#_Toc93311082)

[第16章 应用承载（中篇） 473](#_Toc93311083)

[16.1 中间件委托链 473](#_Toc93311084)

[16.1.1 HttpContext 473](#_Toc93311085)

[16.1.2 中间件 474](#_Toc93311086)

[16.1.3 中间件管道的构建 474](#_Toc93311087)

[16.2 服务器 476](#_Toc93311088)

[16.2.1 IServer 476](#_Toc93311089)

[16.2.2 针对服务器的适配 476](#_Toc93311090)

[16.2.3 HttpListenerServer 478](#_Toc93311091)

[16.3 承载服务 480](#_Toc93311092)

[16.3.1 WebHostedService 480](#_Toc93311093)

[16.3.2 WebHostBuilder 481](#_Toc93311094)

[16.3.3 应用构建 482](#_Toc93311095)

[第17 章 应用承载（下篇） 485](#_Toc93311096)

[17.1 共享上下文 485](#_Toc93311097)

[17.1.1 HttpContext 485](#_Toc93311098)

[17.1.2 服务器适配 488](#_Toc93311099)

[17.1.3 获取上下文 490](#_Toc93311100)

[17.1.4 上下文的创建与释放 491](#_Toc93311101)

[17.1.5 RequestServices 492](#_Toc93311102)

[17.2 IServer + IHttpApplication 493](#_Toc93311103)

[17.2.1 IServer 494](#_Toc93311104)

[17.2.2 HostingApplication 494](#_Toc93311105)

[17.2.3 诊断日志 496](#_Toc93311106)

[17.3 中间件委托链 501](#_Toc93311107)

[17.3.1 IApplicationBuilder 501](#_Toc93311108)

[17.3.2 弱类型中间件 503](#_Toc93311109)

[17.3.3 强类型中间件 506](#_Toc93311110)

[17.3.4 注册中间件 507](#_Toc93311111)

[17.4 应用的承载 508](#_Toc93311112)

[17.4.1 GenericWebHostServiceOptions 508](#_Toc93311113)

[17.4.2 GenericWebHostService 510](#_Toc93311114)

[17.4.3 GenericWebHostBuilder 511](#_Toc93311115)

[17.4.4 ConfigureWebHostDefaults 518](#_Toc93311116)

[17.5 Minimal API 520](#_Toc93311117)

[17.5.1 WebApplication 521](#_Toc93311118)

[17.5.2 WebApplication的构建 522](#_Toc93311119)

[17.5.3 工厂方法 530](#_Toc93311120)

[第18章 服务器 532](#_Toc93311121)

[18.1 自定义服务器 532](#_Toc93311122)

[18.1.1 IServer 532](#_Toc93311123)

[18.1.2 请求和响应特性 534](#_Toc93311124)

[18.1.3 StreamBodyFeature 535](#_Toc93311125)

[18.1.4 HttpListenerServer 535](#_Toc93311126)

[18.2 KestrelServer 538](#_Toc93311127)

[18.2.1 注册终结点 538](#_Toc93311128)

[18.2.2 限制约束 547](#_Toc93311129)

[18.2.3 其他设置 556](#_Toc93311130)

[18.2.4 设计与实现 557](#_Toc93311131)

[18.3 HTTP.SYS 567](#_Toc93311132)

[18.3.1 HTTP.SYS简介 568](#_Toc93311133)

[18.3.2 UseHttpSys 568](#_Toc93311134)

[18.3.3 HttpSysOptions 569](#_Toc93311135)

[18.4 IIS 571](#_Toc93311136)

[18.4.1 ASP.NET Core Module 571](#_Toc93311137)

[18.4.2 In-Process部署模式 571](#_Toc93311138)

[18.4.3 Out-of-Process部署模式 573](#_Toc93311139)

[18.4.4 <aspnetcore>配置 576](#_Toc93311140)

[第19章 静态文件 579](#_Toc93311141)

[19.1 搭建文件服务器 579](#_Toc93311142)

[19.1.1 发布物理文件 579](#_Toc93311143)

[19.1.2 呈现目录结构 581](#_Toc93311144)

[19.1.3 显示默认页面 582](#_Toc93311145)

[19.1.4 映射媒体类型 584](#_Toc93311146)

[19.2 处理文件请求 585](#_Toc93311147)

[19.2.1 条件请求 586](#_Toc93311148)

[19.2.2 区间请求 589](#_Toc93311149)

[19.2.3 StaticFileMiddleware 591](#_Toc93311150)

[19.3 处理目录请求 594](#_Toc93311151)

[19.3.1 DirectoryBrowserMiddleware 594](#_Toc93311152)

[19.3.2 DefaultFilesMiddleware 598](#_Toc93311153)

[第20章 路由 602](#_Toc93311154)

[20.1 路由映射 602](#_Toc93311155)

[20.1.1 注册终结点 602](#_Toc93311156)

[20.1.2 设置内联约束 604](#_Toc93311157)

[20.1.3 可缺省路由参数 605](#_Toc93311158)

[20.1.4 特殊的路由参数 606](#_Toc93311159)

[20.1.5 主机名绑定 607](#_Toc93311160)

[20.1.6 更加自由的定义方式 608](#_Toc93311161)

[20.2 路由分发 609](#_Toc93311162)

[20.2.1 路由模式 609](#_Toc93311163)

[20.2.2 路由终结点 614](#_Toc93311164)

[20.2.3 中间件 620](#_Toc93311165)

[20.2.4 处理器适配 626](#_Toc93311166)

[20.2.5 Minimal API 634](#_Toc93311167)

[20.3 路由约束 635](#_Toc93311168)

[20.3.1 预定义的IRouteConstraint 636](#_Toc93311169)

[20.3.2 InlineConstraintResolver 636](#_Toc93311170)

[20.3.3 自定义约束 637](#_Toc93311171)

[第21章 异常处理 640](#_Toc93311172)

[21.1 呈现错误信息 640](#_Toc93311173)

[21.1.1 开发者异常页面 640](#_Toc93311174)

[21.1.2 定制异常页面 642](#_Toc93311175)

[21.1.3 针对响应状态码定制错误页面 643](#_Toc93311176)

[21.2 开发者异常页面 645](#_Toc93311177)

[21.2.1 IDeveloperPageExceptionFilter 645](#_Toc93311178)

[21.2.2 显示编译异常信息 646](#_Toc93311179)

[21.2.3 DeveloperExceptionPageMiddleware 649](#_Toc93311180)

[21.3 异常处理器 650](#_Toc93311181)

[21.3.1 ExceptionHandlerMiddleware 651](#_Toc93311182)

[21.3.2 IExceptionHandlerPathFeature特性 652](#_Toc93311183)

[21.3.3 清除缓存 654](#_Toc93311184)

[21.3.4 404响应 656](#_Toc93311185)

[21.4 响应状态码页面 657](#_Toc93311186)

[21.4.1 StatusCodePagesMiddleware 658](#_Toc93311187)

[21.4.2 阻止处理异常 658](#_Toc93311188)

[21.4.3 注册中间件 660](#_Toc93311189)

[第22章 响应缓存 665](#_Toc93311190)

[22.1 缓存响应内容 665](#_Toc93311191)

[22.1.1 基于路径的响应缓存 665](#_Toc93311192)

[22.1.2 引入其他缓存维度 666](#_Toc93311193)

[22.1.3 缓存屏蔽 670](#_Toc93311194)

[22.2 HTTP-Cache 672](#_Toc93311195)

[22.2.1 私有缓存和共享缓存 672](#_Toc93311196)

[22.2.2 响应的提取 672](#_Toc93311197)

[22.2.3 新鲜度检验 673](#_Toc93311198)

[22.2.4 显式缓存控制 674](#_Toc93311199)

[22.3 中间件 675](#_Toc93311200)

[22.3.1 缓存上下文 675](#_Toc93311201)

[22.3.2 缓存策略 676](#_Toc93311202)

[22.3.3 缓存键 677](#_Toc93311203)

[22.3.4 缓存的读写 677](#_Toc93311204)

[22.3.5 ResponseCachingMiddleware 678](#_Toc93311205)

[22.3.6 注册中间件 680](#_Toc93311206)

[第23章 会话 682](#_Toc93311207)

[23.1 利用会话保留“语境” 682](#_Toc93311208)

[23.1.1 设置和提取会话状态 682](#_Toc93311209)

[23.1.2 查看存储的会话状态 683](#_Toc93311210)

[23.1.3 查看Cookie 685](#_Toc93311211)

[23.2 会话状态的读写 686](#_Toc93311212)

[23.2.1 ISession 686](#_Toc93311213)

[23.2.2 DistributedSession 687](#_Toc93311214)

[23.2.3 ISessionStore 688](#_Toc93311215)

[23.3 会话中间件 688](#_Toc93311216)

[23.3.1 SessionOptions 689](#_Toc93311217)

[23.3.2 ISessionFeature 690](#_Toc93311218)

[23.3.3 SessionMiddleware 690](#_Toc93311219)

[第24章 HTTPS策略 693](#_Toc93311220)

[24.1 HTTPS终结点的切换 693](#_Toc93311221)

[24.1.1 构建HTTPS站点 693](#_Toc93311222)

[24.1.2 HTTPS重定向 695](#_Toc93311223)

[24.1.3 浏览器自动重定向 696](#_Toc93311224)

[24.1.4 HSTS选项配置 698](#_Toc93311225)

[24.2 HTTPS重定向 700](#_Toc93311226)

[24.2.1 HttpsRedirectionOptions 700](#_Toc93311227)

[24.2.2 HttpsRedirectionMiddleware中间件 700](#_Toc93311228)

[24.2.3 中间件注册 702](#_Toc93311229)

[24.3 HSTS 703](#_Toc93311230)

[24.3.1 HstsOptions 703](#_Toc93311231)

[24.3.2 HstsMiddleware 703](#_Toc93311232)

[24.3.3 中间件注册 704](#_Toc93311233)

[第25章 重定向 705](#_Toc93311234)

[25.1 基于规则的重定向 705](#_Toc93311235)

[25.1.1 客户端重定向 705](#_Toc93311236)

[25.1.2 服务端重定向 706](#_Toc93311237)

[25.1.3 IIS重写规则 707](#_Toc93311238)

[25.1.4 Apache 重写规则 708](#_Toc93311239)

[25.1.5 HTTPS重定向 710](#_Toc93311240)

[25.2 重定向中间件 711](#_Toc93311241)

[25.2.1 重定向规则 712](#_Toc93311242)

[25.2.2 RewriteMiddleware 712](#_Toc93311243)

[25.3 预定义规则 714](#_Toc93311244)

[25.3.1 “万能”规则 714](#_Toc93311245)

[25.3.2 客户端重定向 715](#_Toc93311246)

[25.3.3 服务端重定向 716](#_Toc93311247)

[25.3.4 WWW重定向 716](#_Toc93311248)

[25.3.5 HTTPS重定向 718](#_Toc93311249)

[25.3.6 IIS重写规则 718](#_Toc93311250)

[25.3.7 Apache重写规则 720](#_Toc93311251)

[第26章 限流 722](#_Toc93311252)

[26.1 控制并发量 722](#_Toc93311253)

[26.1.1 设置并发和等待请求阈值 722](#_Toc93311254)

[26.1.2 基于队列的策略 725](#_Toc93311255)

[26.1.3 基于栈的策略 726](#_Toc93311256)

[26.2 并发限制中间件 727](#_Toc93311257)

[26.2.1 等待队列策略 727](#_Toc93311258)

[26.2.2 ConcurrencyLimiterMiddleware 727](#_Toc93311259)

[26.2.3 处理拒绝请求 728](#_Toc93311260)

[26.3 等待队列策略 729](#_Toc93311261)

[26.3.1 基于队列的处理策略 729](#_Toc93311262)

[26.3.2 基于栈的处理策略 731](#_Toc93311263)

[第27章 736](#_Toc93311264)

[27.1 认证、登录与注销 736](#_Toc93311265)

[27.1.1 认证票据 736](#_Toc93311266)

[27.1.2 基于Cookie的认证 737](#_Toc93311267)

[27.1.3 强制认证 739](#_Toc93311268)

[27.1.4 登录与注销 739](#_Toc93311269)

[27.2 身份与用户 740](#_Toc93311270)

[28.2.1 IIdentity 740](#_Toc93311271)

[28.2.2 IPrincipal 744](#_Toc93311272)

[27.3 认证模型 746](#_Toc93311273)

[27.3.1 认证票据 746](#_Toc93311274)

[27.3.2 认证处理器 749](#_Toc93311275)

[27.3.3 认证服务 754](#_Toc93311276)

[27.3.4 服务注册 757](#_Toc93311277)

[27.3.5 AuthenticationMiddleware 760](#_Toc93311278)

[27.4 Cookie认证方案 762](#_Toc93311279)

[27.4.1 AuthenticationHandler<TOptions> 762](#_Toc93311280)

[27.4.2 CookieAuthenticationHandler 766](#_Toc93311281)

[27.4.3 注册CookieAuthenticationHandler 773](#_Toc93311282)

[第28章 授权 776](#_Toc93311283)

[28.1 基于“角色”的授权 776](#_Toc93311284)

[28.1.1 基于“要求”的授权 776](#_Toc93311285)

[28.1.2 预定义授权策略 779](#_Toc93311286)

[28.1.3 基于终结点的自动化授权 781](#_Toc93311287)

[28.2 基于 “要求” 的授权 782](#_Toc93311288)

[28.2.1 IAuthorizationHandler 782](#_Toc93311289)

[28.2.2 预定义授权处理器 783](#_Toc93311290)

[28.2.3 授权检验 786](#_Toc93311291)

[28.3 基于“策略”的授权 791](#_Toc93311292)

[28.3.1 授权策略的构建 791](#_Toc93311293)

[28.3.2 授权策略的注册 792](#_Toc93311294)

[28.3.3 授权检验 793](#_Toc93311295)

[28.4 授权与路由 794](#_Toc93311296)

[28.4.1 IAuthorizeData 794](#_Toc93311297)

[28.4.2 IAllowAnonymous 796](#_Toc93311298)

[28.4.3 IPolicyEvaluator 796](#_Toc93311299)

[28.4.4 IAuthorizationMiddlewareResultHandler 799](#_Toc93311300)

[28.4.5 AuthorizationMiddleware 800](#_Toc93311301)

[第29章 跨域资源共享 803](#_Toc93311302)

[29.1 处理跨域资源 803](#_Toc93311303)

[29.1.1 跨域调用API 803](#_Toc93311304)

[29.1.2 提供者显式授权 805](#_Toc93311305)

[29.1.3 基于策略的资源授权 807](#_Toc93311306)

[29.1.4 将CORS规则应用到路由上 807](#_Toc93311307)

[29.2 CORS规范 808](#_Toc93311308)

[29.2.1 同源策略 808](#_Toc93311309)

[29.2.2 针对资源的授权 809](#_Toc93311310)

[29.2.3 获取授权的方式 810](#_Toc93311311)

[29.2.4 用户凭证 812](#_Toc93311312)

[29.3 CORS中间件 812](#_Toc93311313)

[29.3.1 CORS策略 812](#_Toc93311314)

[29.3.2 CORS与路由 814](#_Toc93311315)

[29.3.3 CORS授权 815](#_Toc93311316)

[29.3.4 CorsMiddleware 817](#_Toc93311317)

[第30章 健康检查 820](#_Toc93311318)

[30.1 检查应用的健康状况 820](#_Toc93311319)

[30.1.1 确定当前应用是否可用 820](#_Toc93311320)

[30.1.2 定制健康检查逻辑 821](#_Toc93311321)

[30.1.3 改变响应状态码 822](#_Toc93311322)

[30.1.4 细粒度的健康检查 823](#_Toc93311323)

[30.1.5 定制响应内容 824](#_Toc93311324)

[30.1.6 过滤IHealthCheck对象 825](#_Toc93311325)

[30.2 设计与实现 826](#_Toc93311326)

[30.2.1 IHealthCheck 826](#_Toc93311327)

[30.2.2 HealthCheckService 831](#_Toc93311328)

[30.2.3 HealthCheckMiddleware 834](#_Toc93311329)

[30.3 发布健康报告 837](#_Toc93311330)

[30.3.1 定期发布健康报告 838](#_Toc93311331)

[30.3.2 IHealthCheckPublisher 838](#_Toc93311332)

[30.3.3 HealthCheckPublisherHostedService 839](#_Toc93311333)