

ASP.NET 6 框架揭秘

第1章 编程体验.....	1
1.1 控制台程序.....	1
1.1.1 构建开发环境.....	1
1.1.2 命令行构建 .NET 应用	2
1.2 ASP.NET 应用	6
1.2.1 launchSettings.json	7
1.2.2 Minimal API	9
1.2.3 中间件	10
1.2.4 配置选项.....	14
1.2.5 诊断日志.....	16
1.2.6 路由.....	17
1.3 MVC.....	18
1.3.1 定义 Controller	18
1.3.2 引入视图.....	19
1.4 gRPC.....	20
1.4.1 定义服务	20
1.4.2 实现和承载	22
1.4.3 调用服务	24
1.5 Dapr.....	25
1.5.1 构建开发环境.....	25
1.5.2 服务调用.....	26
1.5.3 状态管理.....	29
1.5.4 发布订阅.....	32
1.5.5 Actor 模型	34
第2章 依赖注入（上篇）	37
2.1 控制反转	37
2.1.1 流程控制的反转	37

2.1.2 好莱坞法则	39
2.1.3 流程定制	40
2.2 IoC 模式	40
2.2.1 模板方法	40
2.2.2 工厂方法	41
3.2.3 抽象工厂	43
2.3 依赖注入	44
2.3.1 由容器提供对象	45
2.3.2 三种注入方式	46
2.3.3 Service Locator 模式	48
2.4 一个简易版的依赖注入容器	49
2.4.1 编程体验	49
3.4.2 设计与实现	53
3.4.3 扩展方法	58
第3章 依赖注入（下篇）	62
3.1 利用容器提供服务	62
3.1.1 服务的注册与消费	62
3.1.2 生命周期	64
3.1.3 服务注册的验证	66
3.2 服务注册	69
3.2.1 ServiceDescriptor	69
3.2.2 IServiceCollection	70
3.3 服务的消费	73
3.3.1(IServiceProvider	74
3.3.2 服务实例的创建	75
3.3.3 生命周期	76
3.3.4 ActivatorUtilities	81
3.4 扩展	85

3.4.1. 适配	85
3.4.2. IServiceProviderFactory< TContainerBuilder>	86
3.4.3. 整合第三方依赖注入框架	86
第4章 文件系统.....	91
4.1 抽象的文件系统	91
4.1.1 树形层次结构	91
4.1.2 读取文件内容	93
4.1.3 监控文件的变化	95
4.2 文件与目录	95
4.2.1 IChangeToken	96
4.2.2 IFileProvider	97
4.2.3 两个特殊的文件系统	99
4.3 物理文件系统	101
4.3.1 PhysicalFileInfo	101
4.3.2 PhysicalDirectoryInfo	102
4.3.3 PhysicalDirectoryContents	102
4.3.4 NotFoundDirectoryContents	102
4.3.5 PhysicalFilesWatcher	103
4.4 内嵌文件系统	104
4.4.1 将项目文件变成内嵌资源	104
4.4.2 读取资源文件	105
4.4.3 EmbeddedFileProvider	106
第5章 配置选项（上篇）	109
5.1 读取配置信息	109
5.1.1 编程模型三要素	109
5.1.2 以键值对的形式读取配置	109
5.1.3 读取结构化的配置	111
5.1.4 将结构化配置直接绑定为对象	113

5.1.5 将配置定义在文件中	113
5.1.6 根据环境动态加载配置文件	114
5.1.7 配置内容的同步	116
5.2 配置模型	117
5.2.1 数据结构及其转换	117
5.2.2 IConfiguration	118
5.2.3 IConfigurationProvider	120
5.2.4 IConfigurationSource	121
5.2.5 IConfigurationBuilder	121
5.2.6 ConfigurationManager	122
5.3 配置绑定	123
5.3.1 绑定配置项的值	123
5.3.2 绑定复合对象	125
5.3.3 绑定集合	126
5.3.4 绑定字典	129
5.4 配置的同步	129
5.4.1 配置数据流	130
5.4.2 ConfigurationReloadToken	130
5.4.3 ConfigurationRoot	131
5.4.4 ConfigurationSection	132
5.5 多样性的配置源	133
5.5.1 MemoryConfigurationSource	133
5.5.2 EnvironmentVariablesConfigurationSource	134
5.5.3 CommandLineConfigurationSource	136
5.5.4 FileConfigurationSource	139
5.5.5 StreamConfigurationSource	147
5.5.6 ChainedConfigurationSource	148
第6章 配置选项（下篇）	151

6.1 Options 模式	151
6.1.1 将配置绑定为 Options 对象	151
6.1.2 提供具名的 Options	152
6.1.3 配置源的同步	153
6.1.4 直接初始化 Options 对象	155
6.1.5 根据依赖服务的 Options 设置	157
6.1.6 验证 Options 的有效性	158
6.2 Options 模型	159
6.2.1 OptionsManager<TOptions>	159
6.2.2 IOptionsFactory<TOptions>	161
6.2.3 IOptionsMonitorCache<TOptions>	167
6.2.4 IOptionsMonitor<TOptions>	168
6.3 依赖注入	170
6.3.1 服务注册	170
6.3.2 IOptions<TOptions> 与 IOptionsSnapshot<TOptions>	175
6.3.3 集成配置系统	177
第7章 诊断日志（上篇）	179
7.1 各种诊断日志形式	179
7.1.1 调试日志	179
7.1.2 跟踪日志	180
7.1.3 事件日志	181
7.1.4 诊断日志	184
7.2 Debugger 调试日志	187
7.2.1 Debugger	187
7.2.2 Debug	188
7.3 TraceSource 跟踪日志	189
7.3.1 跟踪日志模型三要素	189
7.3.2 预定义 TraceListener	196

7.3.3 Trace	199
7.4 EventSource 事件日志	202
7.4.1 EventSource	202
7.4.2 EventListener	207
7.4.3 活动跟踪	210
7.5 DiagnosticSource 诊断日志	213
7.5.1 标准的观察者模式	213
7.5.2 AnonymousObserver< T>	215
7.5.3 强类型的事件订阅	218
7.5.4 针对活动的跟踪	219
第8章 诊断日志（中篇）	221
8.1 统一日志编程模式	221
8.1.1 日志输出	221
8.1.2 日志过滤	225
8.1.3 日志范围	230
8.1.4 LoggerMessage	231
8.2 日志模型详解	233
8.2.1 日志模型三要素	233
8.2.2 ILogger	233
8.2.3 ILoggerProvider	235
8.2.4 ILoggerFactory	235
8.2.5 LoggerMessage	238
8.3 日志范围	240
8.3.1 调用链跟踪	240
8.3.2 服务范围堆栈	249
8.3.3 Activity 的应用	251
8.4 依赖注入	252
8.4.1 核心服务	252

8.4.2 配置	254
8.4.3 日志过滤规则	257
第9章 诊断日志（下篇）	259
9.1 控制台	259
9.1.1 ConsoleFormatter	259
9.1.2 ConsoleLogger	265
9.1.3 ConsoleLoggerProvider	267
9.1.4 服务注册	268
9.2 调试器	272
9.2.1 DebugLogger	272
9.2.2 DebugLoggerProvider	273
9.3 TraceSource 日志	274
9.3.1 TraceSourceLogger	274
9.3.2 TraceSourceLoggerProvider	276
9.4 EventSource 日志	277
9.4.1 LoggingEventSource	277
9.4.2 EventSourceLogger	278
9.4.3 EventSourceLoggerProvider	281
9.4.4 日志范围	282
第10章 对象池	287
13.1 利用对象池复用对象	287
13.1.1 对象的“借”与“还”	287
13.1.2 依赖注入	288
13.1.3 池化对象策略	289
13.1.4 对象池的大小	290
13.1.5 对象的释放	291
13.2 池化对象管理	293
10.2.1 IPooledObjectPolicy<T>	293

10.2.2	ObjectPool<T>	293
10.2.3	ObjectPoolProvider	298
13.3	扩展应用.....	299
10.3.1	池化集合	299
10.3.2	池化 StringBuilder	301
10.3.3	ArrayPool<T>	302
10.3.4	MemoryPool<T>	303
第 11 章	缓存.....	305
11.1	将数据缓存起来	305
11.1.1	将数据缓存在内存中	305
11.1.2	将数据缓存在 Redis 中	306
11.1.3	将数据缓存在 SQL Server 中	309
11.2	本地内存缓存.....	310
11.2.1	ICacheEntry	311
11.2.2	MemoryCacheEntryOptions	313
11.2.3	IMemoryCache.....	314
11.3	分布式缓存	321
11.3.1	IDistributedCache.....	321
11.3.2	Redis 缓存	322
11.3.3	SQL Server 缓存.....	324
第 12 章	HTTP 调用	327
12.1	HttpClient 的工厂	327
12.1.1	手工创建 HttpClient	327
12.1.2	使用 IHttpClientFactory 工厂	328
12.1.3	直接注入 HttpClient	329
12.1.4	定制 HttpClient	329
12.1.5	强类型客户端	330
12.1.6	失败重试	331
12.2	HttpMessageHandler 管道	332

12.2.1.	HttpMessageHandler	332
12.2.2.	DelegatingHandler	333
12.2.3.	诊断日志	336
12.2.4.	复用 HttpClientHandler	338
12.3	HttpClient 的构建	339
12.3.1.	HttpMessageHandlerBuilder	340
12.3.2.	HttpClientFactoryOptions	341
12.3.3.	IHttpMessageHandlerBuilderFilter	346
12.3.4.	IHttpClientFactory	347
12.4	依赖注入	355
12.4.1.	基础服务注册	355
12.4.2.	定制 HttpClient	357
12.4.3.	强类型客户端	360
第 13 章	数据保护	365
13.1	加密与哈希	365
13.1.6	数据加密与解密	365
13.1.7	设置加密内容的有效期	367
13.1.8	撤销密钥	368
13.1.9	“瞬时” 加解密	369
13.1.10	密码哈希	370
13.2	加密模型	372
13.2.1.	IAuthenticatedEncryptor	372
13.2.2.	IKey	372
13.2.3.	IKeyRing	373
13.2.4.	IKeyRingProvider	375
13.2.5.	IDataProtector	376
13.3	密钥管理	383
13.3.1.	KeyManagementOptions	383

13.3.2.	Key	387
13.3.3.	IKeyManager	389
13.3.4.	KeyRingProvider	394
13.4	依赖注入	400
13.4.1.	注册基础服务	400
13.4.2.	密钥管理配置	404
13.4.3.	扩展配置	405
第 14 章	服务承载	409
14.1	服务承载	409
14.1.1	性能指标收集服务	409
14.1.2	依赖注入	410
14.1.3	配置选项	413
14.1.4	承载环境	415
14.1.5	日志	417
14.2	承载模型	419
14.2.1	IHostedService	419
14.2.2	IHost	420
14.2.3	IHostBuilder	424
14.3	承载流程	429
14.3.1	服务宿主	429
14.3.2	承载设置	432
14.3.3	创建宿主	437
14.3.4	静态类型 Host	442
第 15 章	管道 (上篇)	445
15.1	管道式的请求处理	445
15.1.1	承载方式的变迁	445
15.1.2	中间件	449
15.1.3	定义强类型中间件	452

15.1.4 按照约定定义中间件.....	453
15.2 依赖注入.....	454
15.2.1 服务注册	454
15.2.2 服务注入	455
15.2.3 生命周期	457
15.3 配置.....	461
15.3.1 初始化配置.....	462
15.3.2 以键值对形式读取和修改配置	462
15.3.3 注册配置源.....	463
15.4 承载环境.....	464
15.4.1 IWebHostEnvironment.....	464
15.4.2 通过配置定制承载环境	465
15.4.3 设置监听地址	468
15.4.4 针对环境的编程.....	469
第 16 章 应用承载（中篇）	473
16.1 中间件委托链.....	473
16.1.1 HttpContext.....	473
16.1.2 中间件	474
16.1.3 中间件管道的构建	474
16.2 服务器	476
16.2.1 IServer	476
16.2.2 针对服务器的适配	476
16.2.3 HttpListenerServer	478
16.3 承载服务.....	480
16.3.1 WebHostedService.....	480
16.3.2 WebHostBuilder	481
16.3.3 应用构建	482
第 17 章 应用承载（下篇）	485

17.1 共享上下文	485
17.1.1 HttpContext.....	485
17.1.2 服务器适配.....	488
17.1.3 获取上下文.....	490
17.1.4 上下文的创建与释放.....	491
17.1.5 RequestServices	492
17.2 IServer + IHttpApplication	493
17.2.1 IServer	494
17.2.2 HostingApplication.....	494
17.2.3 诊断日志	496
17.3 中间件委托链.....	501
17.3.1 IApplicationBuilder	501
17.3.2 弱类型中间件	503
17.3.3 强类型中间件	506
17.3.4 注册中间件.....	507
17.4 应用的承载	508
17.4.1 GenericWebHostServiceOptions.....	508
17.4.2 GenericWebHostService.....	510
17.4.3 GenericWebHostBuilder	511
17.4.4 ConfigureWebHostDefaults	518
17.5 Minimal API.....	520
17.5.1 WebApplication	521
17.5.2 WebApplication 的构建	522
17.5.3 工厂方法	530
第 18 章 服务器.....	532
18.1 自定义服务器.....	532
18.1.1 IServer	532
18.1.2 请求和响应特性.....	534

18.1.3 StreamBodyFeature	535
18.1.4 HttpListenerServer	535
18.2 KestrelServer	538
18.2.1 注册终结点	538
18.2.2 限制约束	547
18.2.3 其他设置	556
18.2.4 设计与实现	557
18.3 HTTP.SYS	567
18.3.1 HTTP.SYS 简介	568
18.3.2 UseHttpSys	568
18.3.3 HttpSysOptions	569
18.4 IIS 571	
18.4.1 ASP.NET Core Module	571
18.4.2 In-Process 部署模式	571
18.4.3 Out-of-Process 部署模式	573
18.4.4 <aspnetcore> 配置	576
第 19 章 静态文件	579
19.1 搭建文件服务器	579
19.1.1 发布物理文件	579
19.1.2 呈现目录结构	581
19.1.3 显示默认页面	582
19.1.4 映射媒体类型	584
19.2 处理文件请求	585
19.2.1 条件请求	586
19.2.2 区间请求	589
19.2.3 StaticFileMiddleware	591
19.3 处理目录请求	594
19.3.1 DirectoryBrowserMiddleware	594

19.3.2 DefaultFilesMiddleware	598
第 20 章 路由.....	602
20.1 路由映射	602
20.1.1 注册终结点	602
20.1.2 设置内联约束	604
20.1.3 可缺省路由参数	605
20.1.4 特殊的路由参数	606
20.1.5 主机名绑定	607
20.1.6 更加自由的定义方式	608
20.2 路由分发	609
20.2.1 路由模式	609
20.2.2 路由终结点	614
20.2.3 中间件	620
20.2.4 处理器适配	626
20.2.5 Minimal API	634
20.3 路由约束	635
20.3.1 预定义的 IRouteConstraint	636
20.3.2 InlineConstraintResolver	636
20.3.3 自定义约束	637
第 21 章 异常处理	640
21.1 呈现错误信息	640
21.1.1 开发者异常页面	640
21.1.2 定制异常页面	642
21.1.3 针对响应状态码定制错误页面	643
21.2 开发者异常页面	645
21.2.1 IDveloperPageExceptionHandler	645
21.2.2 显示编译异常信息	646
21.2.3 DeveloperExceptionHandlerMiddleware	649
21.3 异常处理器	650

21.3.1	ExceptionHandlerMiddleware	651
21.3.2	IExceptionHandlerPathFeature 特性	652
21.3.3	清除缓存	654
21.3.4	404 响应	656
21.4	响应状态码页面	657
21.4.1	StatusCodePagesMiddleware	658
21.4.2	阻止处理异常	658
21.4.3	注册中间件	660
第 22 章	响应缓存	665
22.1	缓存响应内容	665
22.1.1	基于路径的响应缓存	665
22.1.2	引入其他缓存维度	666
22.1.3	缓存屏蔽	670
22.2	HTTP-Cache	672
22.2.1	私有缓存和共享缓存	672
22.2.2	响应的提取	672
22.2.3	新鲜度检验	673
22.2.4	显式缓存控制	674
22.3	中间件	675
22.3.1	缓存上下文	675
22.3.2	缓存策略	676
22.3.3	缓存键	677
22.3.4	缓存的读写	677
22.3.5	ResponseCachingMiddleware	678
22.3.6	注册中间件	680
第 23 章	会话	682
23.1	利用会话保留“语境”	682
23.1.1	设置和提取会话状态	682
23.1.2	查看存储的会话状态	683

23.1.3 查看 Cookie	685
23.2 会话状态的读写	686
23.2.1 ISession	686
23.2.2 DistributedSession	687
23.2.3 ISessionStore	688
23.3 会话中间件	688
23.3.1 SessionOptions	689
23.3.2 ISessionFeature	690
23.3.3 SessionMiddleware	690
第 24 章 HTTPS 策略	693
24.1 HTTPS 终结点的切换	693
24.1.1 构建 HTTPS 站点	693
24.1.2 HTTPS 重定向	695
24.1.3 浏览器自动重定向	696
24.1.4 HSTS 选项配置	698
24.2 HTTPS 重定向	700
24.2.1 HttpsRedirectionOptions	700
24.2.2 HttpsRedirectionMiddleware 中间件	700
24.2.3 中间件注册	702
24.3 HSTS	703
24.3.1 HstsOptions	703
24.3.2 HstsMiddleware	703
24.3.3 中间件注册	704
第 25 章 重定向	705
25.1 基于规则的重定向	705
25.1.1 客户端重定向	705
25.1.2 服务端重定向	706
25.1.3 IIS 重写规则	707

25.1.4 Apache 重写规则.....	708
25.1.5 HTTPS 重定向.....	710
25.2 重定向中间件.....	711
25.2.1 重定向规则.....	712
25.2.2 RewriteMiddleware.....	712
25.3 预定义规则.....	714
25.3.1 “万能”规则.....	714
25.3.2 客户端重定向.....	715
25.3.3 服务端重定向.....	716
25.3.4 WWW 重定向.....	716
25.3.5 HTTPS 重定向.....	718
25.3.6 IIS 重写规则.....	718
25.3.7 Apache 重写规则.....	720
第 26 章 限流.....	722
26.1 控制并发量.....	722
26.1.1 设置并发和等待请求阈值.....	722
26.1.2 基于队列的策略.....	725
26.1.3 基于栈的策略.....	726
26.2 并发限制中间件.....	727
26.2.1 等待队列策略.....	727
26.2.2 ConcurrencyLimiterMiddleware.....	727
26.2.3 处理拒绝请求.....	728
26.3 等待队列策略.....	729
26.3.1 基于队列的处理策略.....	729
26.3.2 基于栈的处理策略.....	731
第 27 章.....	736
27.1 认证、登录与注销.....	736
27.1.1 认证票据.....	736

27.1.2 基于 Cookie 的认证.....	737
27.1.3 强制认证	739
27.1.4 登录与注销.....	739
27.2 身份与用户	740
28.2.1 IIdentity	740
28.2.2 IPrincipal	744
27.3 认证模型.....	746
27.3.1 认证票据	746
27.3.2 认证处理器.....	749
27.3.3 认证服务	754
27.3.4 服务注册	757
27.3.5 AuthenticationMiddleware	760
27.4 Cookie 认证方案.....	762
27.4.1 AuthenticationHandler<TOptions>	762
27.4.2 CookieAuthenticationHandler	766
27.4.3 注册 CookieAuthenticationHandler	773
第 28 章 授 权	776
28.1 基于“角色”的授权	776
28.1.1 基于“要求”的授权.....	776
28.1.2 预定义授权策略.....	779
28.1.3 基于终结点的自动化授权	781
28.2 基于“要求”的授权.....	782
28.2.1 IAuthorizationHandler	782
28.2.2 预定义授权处理器	783
28.2.3 授权检验	786
28.3 基于“策略”的授权	791
28.3.1 授权策略的构建.....	791
28.3.2 授权策略的注册.....	792

28.3.3 授权检验	793
28.4 授权与路由	794
28.4.1 IAuthorizeData	794
28.4.2 IAllowAnonymous	796
28.4.3 IPolicyEvaluator	796
28.4.4 IAuthorizationMiddlewareResultHandler	799
28.4.5 AuthorizationMiddleware	800
第 29 章 跨域资源共享	803
29.1 处理跨域资源	803
29.1.1 跨域调用 API	803
29.1.2 提供者显式授权	805
29.1.3 基于策略的资源授权	807
29.1.4 将 CORS 规则应用到路由上	807
29.2 CORS 规范	808
29.2.1 同源策略	808
29.2.2 针对资源的授权	809
29.2.3 获取授权的方式	810
29.2.4 用户凭证	812
29.3 CORS 中间件	812
29.3.1 CORS 策略	812
29.3.2 CORS 与路由	814
29.3.3 CORS 授权	815
29.3.4 CorsMiddleware	817
第 30 章 健康检查	820
30.1 检查应用的健康状况	820
30.1.1 确定当前应用是否可用	820
30.1.2 定制健康检查逻辑	821
30.1.3 改变响应状态码	822

30.1.4 细粒度的健康检查	823
30.1.5 定制响应内容	824
30.1.6 过滤 IHealthCheck 对象	825
30.2 设计与实现	826
30.2.1 IHealthCheck	826
30.2.2 HealthCheckService	831
30.2.3 HealthCheckMiddleware	834
30.3 发布健康报告	837
30.3.1 定期发布健康报告	838
30.3.2 IHealthCheckPublisher	838
30.3.3 HealthCheckPublisherHostedService	839