

目录

第 1 章 全新的开发体验.....	1
1.1 Windows 平台	1
1.1.1 安装开发环境	1
1.1.2 利用命令行创建.NET Core 应用	2
1.1.3 ASP.NET Core 应用	6
1.1.4 ASP.NET Core MVC 应用	13
1.2 Mac OS 用户	17
1.2.1 构建开发环境	17
1.2.2 利用命令行创建.NET Core 程序	17
1.2.3 ASP.NET Core MVC 应用	18
1.3 Linux.....	19
1.3.1 启用 Linux 子系统.....	20
1.3.2 安装开发环境	22
1.3.3 利用命令行创建一个 ASP.NET Core 应用.....	22
1.4 Docker.....	24
第 2 章 跨平台的奥秘.....	27
2.1 历史的枷锁	27
2.1.1 Windows 下的.NET.....	27
2.1.2 非 Windows 下的.NET.....	31
2.2 复用之伤	34
2.2.1 源代码复用	34
2.2.2 程序集复用	36
2.3 全新的布局	44

2.3.1	跨平台的.NET Core	44
2.3.2	统一的 BCL	51
2.3.3	展望未来	56
第 3 章	依赖注入（上篇）	58
3.1	控制反转	58
3.1.1	流程控制的反转	58
3.1.2	好莱坞法则	61
3.1.3	流程定制	62
3.2	IoC 模式	62
3.2.1	模板方法	63
3.2.2	工厂方法	64
3.2.3	抽象工厂	66
3.3	依赖注入	68
3.3.1	由容器提供对象	68
3.3.2	3 种依赖注入方式	69
3.3.3	Service Locator 模式	72
3.4	一个简易版的依赖注入容器	74
3.4.1	编程体验	74
3.4.2	设计与实现	79
3.4.3	扩展方法	85
第 4 章	依赖注入（下篇）	89
4.1	利用容器提供服务	89
4.1.1	服务的注册与消费	89
4.1.2	生命周期	93
4.1.3	针对服务注册的验证	96
4.2	服务注册	99
4.2.1	ServiceDescriptor	99
4.2.2	IServiceCollection	101
4.3	服务的消费	105
4.3.1	IServiceProvider	105
4.3.2	服务实例的创建	106
4.3.3	生命周期	109

4.4	实现概览	113
4.4.1	ServiceProviderEngine 和 ServiceProviderEngineScope	113
4.4.2	ServiceProvider.....	115
4.4.3	注入 IServiceProvider 对象.....	117
4.5	扩展	119
4.5.1	适配	120
4.5.2	IServiceProviderFactory<TContainerBuilder>	120
4.5.3	整合第三方依赖注入框架.....	121
第 5 章	文件系统	126
5.1	抽象的文件系统	126
5.1.1	树形层次结构	126
5.1.2	读取文件内容	128
5.1.3	监控文件的变化.....	130
5.2	设计详解	131
5.2.1	IChangeToken	132
5.2.2	IFileProvider	133
5.2.3	PhysicalFileProvider.....	135
5.2.4	EmbeddedFileProvider	139
5.2.5	两个特殊的 IFileProvider 实现.....	144
5.3	远程文件系统	147
5.3.1	HttpFileInfo 与 HttpDirectoryContents	147
5.3.2	HttpFileProvider	150
5.3.3	FileProviderMiddleware	151
5.3.4	远程文件系统的应用.....	153
第 6 章	配置选项（上篇）	155
6.1	读取配置信息	155
6.1.1	配置编程模型三要素.....	155
6.1.2	以键值对的形式读取配置.....	156
6.1.3	读取结构化的配置.....	157
6.1.4	将结构化配置直接绑定为对象.....	160
6.1.5	将配置定义在文件中.....	161
6.2	配置模型	165

6.2.1	数据结构及其转换.....	166
6.2.2	IConfiguration.....	167
6.2.3	IConfigurationProvider.....	169
6.2.4	IConfigurationSource.....	171
6.2.5	IConfigurationBuilder.....	171
6.3	配置绑定.....	172
6.3.1	绑定配置项的值.....	173
6.3.2	绑定复合数据类型.....	175
6.3.3	绑定集合对象.....	177
6.3.4	绑定字典.....	180
6.4	配置的同步.....	181
6.4.1	配置数据流.....	181
6.4.2	ConfigurationReloadToken.....	182
6.4.3	ConfigurationRoot.....	183
6.4.4	ConfigurationSection.....	185
6.5	多样性的配置源.....	186
6.5.1	MemoryConfigurationSource.....	187
6.5.2	EnvironmentVariablesConfigurationSource.....	188
6.5.3	CommandLineConfigurationSource.....	191
6.5.4	FileConfigurationSource.....	194
6.5.5	StreamConfigurationSource.....	207
6.5.6	ChainedConfigurationSource.....	208
6.5.7	自定义 ConfigurationSource (S616).....	210
第 7 章	配置选项（下篇）.....	215
7.1	Options 模式.....	215
7.1.1	将配置绑定为 Options 对象.....	215
7.1.2	提供具名的 Options.....	217
7.1.3	配置源的同步.....	219
7.1.4	直接初始化 Options 对象.....	221
7.1.5	根据依赖服务的 Options 设置.....	223
7.1.6	验证 Options 的有效性.....	225
7.2	Options 模型.....	226
7.2.1	OptionsManager<TOptions>.....	226

7.2.2	IOptionsFactory<TOptions>	228
7.2.3	IOptionsMonitorCache<TOptions>	237
7.2.4	IOptionsMonitor<TOptions>	238
7.3	依赖注入	240
7.3.1	服务注册	240
7.3.2	IOptions<TOptions>与 IOptionsSnapshot<TOptions>	246
7.3.3	扩展与定制	248
7.3.4	集成配置系统	256
第 8 章	诊断日志（上篇）	258
8.1	各种诊断日志形式	258
8.1.1	调试日志	258
8.1.2	跟踪日志	259
8.1.3	事件日志	262
8.1.4	诊断日志	265
8.2	Debugger 调试日志	268
8.2.1	Debugger	268
8.2.2	Debug	270
8.3	TraceSource 跟踪日志	271
8.3.1	跟踪日志模型三要素	271
8.3.2	预定义 TraceListener	280
8.3.3	Trace	284
8.4	EventSource 事件日志	287
8.4.1	EventSource	287
8.4.2	EventListener	294
8.4.3	荷载对象序列化	298
8.4.4	活动跟踪	302
8.4.5	性能计数	306
8.5	DiagnosticSource 诊断日志	308
8.5.1	标准的观察者模式	308
8.5.2	AnonymousObserver<T>	310
8.5.3	强类型的事件订阅	313
8.5.4	针对活动的跟踪	315

第 9 章 诊断日志（下篇）	317
9.1 统一日志编程模式	317
9.1.1 将日志输出到不同的渠道	317
9.1.2 日志过滤	323
9.1.3 日志范围	329
9.1.4 LoggerMessage	331
9.2 日志模型详解	334
9.2.1 日志模型三要素	334
9.2.2 ILogger	335
9.2.3 日志范围	339
9.2.4 ILoggerProvider	342
9.2.5 ILoggerFactory	342
9.2.6 LoggerMessage	347
9.3 依赖注入	348
9.3.1 服务注册	349
9.3.2 设置日志过滤规则	351
9.4 日志输出渠道	353
9.4.1 控制台	353
9.4.2 调试器	357
9.4.3 TraceSource 日志	359
9.4.4 EventSource 日志	362
第 10 章 承载系统	377
10.1 服务承载	377
10.1.1 承载长时间运行服务	377
10.1.2 依赖注入	379
10.1.3 配置选项	382
10.1.4 承载环境	385
10.1.5 日志	388
10.2 承载模型	391
10.2.1 IHostedService	392
10.2.2 IHost	392
10.2.3 IHostBuilder	397
10.3 实现原理	402

10.3.1	服务宿主	403
10.3.2	针对配置系统的设置.....	406
10.3.3	针对依赖注入框架的设置.....	407
10.3.4	创建宿主	412
10.3.5	静态类型 Host.....	418
第 11 章	管道（上篇）	421
11.1	管道式的请求处理.....	421
11.1.1	两个承载体系.....	421
11.1.2	请求处理管道.....	423
11.1.3	中间件.....	424
11.1.4	定义强类型中间件.....	427
11.1.5	按照约定定义中间件.....	428
11.2	依赖注入.....	430
11.2.1	服务注册.....	430
11.2.2	服务的消费.....	433
11.2.3	生命周期.....	437
11.2.4	集成第三方依赖注入框架.....	443
11.3	配置	444
11.3.1	初始化配置.....	444
11.3.2	以键值对形式读取和修改配置.....	446
11.3.3	合并配置.....	448
11.3.4	注册 IConfigurationSource.....	449
11.4	承载环境.....	450
11.4.1	IWebHostEnvironment.....	450
11.4.2	通过配置定制承载环境.....	452
11.4.3	针对环境的编程.....	454
11.5	初始化	459
11.5.1	Startup	459
11.5.2	IHostingStartup	461
11.5.3	IStartupFilter	464
第 12 章	管道（中篇）	467
12.1	中间件委托链	467

12.1.1	HttpContext.....	467
12.1.2	中间件	468
12.1.3	中间件管道的构建.....	469
12.2	服务器	470
12.2.1	IServer.....	471
12.2.2	针对服务器的适配.....	471
12.2.3	HttpListenerServer.....	473
12.3	承载服务	476
12.3.1	WebHostedService	476
12.3.2	WebHostBuilder.....	476
12.3.3	应用构建	478
第 13 章	管道（下篇）	480
13.1	请求上下文	480
13.1.1	HttpContext.....	480
13.1.2	服务器适配	483
13.1.3	获取上下文	487
13.1.4	上下文的创建与释放.....	488
13.1.5	RequestServices	489
13.2	IServer + IHttpApplication	491
13.1.1	IServer.....	491
13.1.2	HostingApplication.....	492
13.1.3	诊断日志	495
13.3	中间件委托链	501
13.3.1	IApplicationBuilder	501
13.3.2	弱类型中间件	504
13.3.3	强类型中间件	507
13.3.4	注册中间件	509
13.4	应用的承载	510
13.4.1	GenericWebHostServiceOptions.....	510
13.4.2	GenericWebHostService	512
13.4.3	GenericWebHostBuilder	515
13.4.4	ConfigureWebHostDefaults	530
附录	实例演示	533

