



QQ群：850720634

《深入浅出ASP.NET Core》学习路径

开发环境推荐

- 开发工具：Visual Studio 2019
- 操作系统：Windows 10 版本1809以上
- SDK版本：.NET Core SDK 3.1

扩展篇

- Restful服务与WebAPI的区别
- ASP.NET Core WebAPI基础介绍
- WebAPI的可视化界面帮助页
- POSTMan工具的使用
- Fidder工具的调试与使用
- 部署至IIS
- 部署至Ubuntu
- Docker中运行ASP.NET Core
- 原生Azure WebApp
- 如何迁移升级一个ASP.NET Core2.X项目到3.1

框架实践

- 打造一个学校管理的架构设计
- 多层体系架构的设计与实践
- 仓储模式的最佳实现-泛型篇
- EF Core的复杂查询
- EF Core中的多重关系映射
- 多类库之间的批量依赖注入服务实现
- 乐观并发与悲观并发的冲突处理

账户中心功能的实现

- 用户身份认证系统的设计
- 用户登录、注册、注销、密码复杂度验证的实现
- ASP.NET Core Identity 中的源代码解析
- 社交登录(github、微软)的实现
- 忘记密码、重置密码及邮箱验证功能

身份认证及安全性

- 授权与验证的关系介绍
- ASP.NET Core Identity 框架的使用
- 实现基于角色管理权限验证
- 开放式重定向漏洞处理
- ASP.NET Core中的声明授权
- RBAC(角色授权)与CABC(策略授权)的异同点
- 基于委托实现的复杂自定义授权策略

基础知识

- 了解C#和.NET的发展历程
 - 编程语言和开发平台的区别
 - 回顾.NET开发平台的发展历史
 - 未来.NET平台的发展方向
- ASP.NET Core的构成
 - ASP.NET Core概述
 - ASP.NET Core开发环境的搭建
 - ASP.NET Core项目启动流程
 - ASP.NET Core中的配置文件作用
 - ASP.NET Core中间件的工作及原理
 - 常用的中间件及其使用方式

ASP.NET Core MVC框架

- 了解MVC的设计原理
- 从零搭建MVC模板
- 依赖注入的原理和工作解析
- 常规路由与属性路由的异同点
- 新增的Taghelper介绍及其使用
- 如何进行模型绑定与模型验证

数据访问

- 仓储模式是什么以及优势
- ORM的好处与优势
- EntityFramework Core中如何访问数据源
- 数据迁移同步领域模型与数据库架构
- 单表的增删改查
- 文件的上传与处理
- 模糊查询、分组查询及分页功能的实现
- EF Core 调用原生Sql的方法

应用程序公共功能

- Http状态码的作用介绍
- 错误信息拦截与处理
- 日志中间件的介绍及其配置
- 全局统一的异常拦截与日志记录