# Acme.mianshiBookProject项目说明文档

开发环境： Visual Studio 需要支持 .NET 8.0+ 的开发. Node v18，Yarn v1.20 或 npm v6+ (已跟随Node一起安装)

**项目名称**：Acme.mianshiBookProject

**框架**：

1.领域层：包含Domain;Domain.Shared 实体的定义，聚合根和领域服务的实现

2.应用层:Application 和 Application.Contracts。写服务，接口，相关DTO

3.用户界面层： Acme.mianshiBookProject.Web。写UI，包含 Web 应用程序的前端代码，包括视图、控制器和其他与用户界面相关的逻辑

4.Migrator数据迁移层：进行数据迁移

6.Tests 单元测试层：进行单元测试和集成测试代码的编写

## 种子数据添加

位置：在Acme.mianshiBookProject.Domain 下的mianshiBookProjectDataSeederContributor类中添加种子数据代码，为图书和作者添加种子数据。

**数据库迁移:**

操作:开命令行终端, 切换当前目录为Acme.mianshiBookProject.EntityFrameworkCore 项目目录, 输入以下命令:dotnet ef migrations add Added\_Authors

更新数据:在数据库迁移后执行 dotnet ef database update 命令 更新数据

创建DTO：位置Acme.mianshiBookProject.Application.Contracts的对应实体文件下添加DTO

对象映射:位置Acme.mianshiBookProject.EntityFrameworkCore下的EntityFramework的mianshiBookProjectDbContext类中进行对象映射。

**单元测试和集成测试:**

**1.自动审计日志**：IAuditingStore 将审计日志对象保存到数据库中. 存储在数据库AuditLogs表中

**2.软删除** Isdeleted 初始为0 软删除后为1 查询不到该条数据 但是数据库中仍然存在。

位置：Acme.Acme.mianshiBookProject.Domain Authors 文件夹的Author类

代码块public class Author : FullAuditedAggregateRoot<Guid>

FullAuditedAggregateRoot<Guid> 继承使得实体支持软删除 (指实体被删除时, 它并没有从数据库中被删除, 而只是被标记删除), 实体也具有了 审计 属性.

**3.领域服务领域驱动设计 (DDD)**最好仅用于执行核心业务逻辑

位置：Acme.Acme.mianshiBookProject.Domain 领域层项目中的 Authors 文件夹 (命名空间) AuthorManager 类

AuthorManager 类作为领域服务，强制使用一种可控的方式创建作者和修改作者的名字.

应用服务中使用 AuthorManager 来处理与作者相关的业务逻辑。

**4.数据仓库(EfCoreAuthorRepository)**

位置：Acme.Acme.mianshiBookProject.EntityFrameworkCore 项目 (Authors 文件夹)中EfCoreAuthorRepository类

现了 IAuthorRepository 接口，并继承自 EfCoreRepository，这使得它能够利用 ABP 框架提供的标准仓储方法。

**5. AppService(IAuthorAppService)**

位置：Acme.Acme.mianshiBookProject.Application.Contracts 项目的 Authors 命名空间 (文件夹) 接口 说明：IAuthorAppService:public interface IAuthorAppService : IApplicationService

IApplicationService 是一个常规接口, 所有应用服务都继承自它, 所以 ABP 框架可以识别它们.

在 Author 实体中定义标准方法用于CRUD操作.。

**6.角色访问**

[Authorize(xxxx)]在相应的Service方法前加入检查当前账户是否有对应权限

位置:Acme.mianshiBookProject.Application的Authors文件下的AuthorAppService

## Redis缓存

在项目.csproj的文件夹中打开命令行窗口并输入命令:

abp add-package Volo.Abp.Caching.StackExchangeRedis

appsettings.json进行Redis配置

**8.后台作业**

位置：在Acme.mianshiBookProject.Domain的houtaizuoye文件夹中新建EmailSendingArgs来存储后台作业的参数

在Acme.mianshiBookProject.Application 的houtaizuoye文件夹中创建后台作业类EmailSendingJob简单的使用了 IEmailSender 发送电子邮件，创建RegistrationService向队列中添加一个发送电子邮件作业:。在Acme.mianshiBookProject.EntityFrameworkCore的mianshiBookProjectEntityFrameworkCoreModule类中进行配置

**9.signalr实现消息转发**

先安装Volo.Abp.AspNetCore.SignalR包，在添加typeof(AbpAspNetCoreSignalRModule)到模块依赖类

位置在Web里的mianshiBookProjectWebModule类里面添加，在根文件运行yarn add @abp/signalr ，在 package.json 依赖项会添加 @abp/signalr。将@section scripts { <abp-script type="typeof(SignalRBrowserScriptContributor)" /> }添加到Razor页面。在web项目下pages文件夹添加Razor页面

集线器配置：在Web下添加一个文件夹名为SignalRDemo 里面添加ChatHub类进行集线器配置

## 实际项目中可能遇到的问题和解决方案的案例

**1.数据库迁移问题**

问题：在进行数据库迁移时，可能会遇到迁移失败或数据库结构不一致的问题。

解决方案：

确保在 DbContext 中正确配置了实体和关系。

使用 Add-Migration 命令生成迁移文件，并检查生成的代码是否符合预期。

在应用程序启动时调用 Database.Migrate() 方法，确保数据库结构与模型同步。

**2. DTO映射问题**

问题：在创建或更新图书时，DTO 映射可能会出现问题，导致数据未能正确传递。

解决方案：

使用 AutoMapper 或手动映射确保 DTO 和实体之间的属性正确映射。

**3.软删除和审计日志**

问题：在实现软删除时，可能会遇到未能正确标记删除状态的问题。

解决方案：

确保实体实现了 ISoftDelete 接口，并在删除方法中设置 IsDeleted 属性。

在查询时，确保过滤掉已删除的记录。

**4. Redis 缓存策略**

问题：在使用 Redis 进行缓存时，可能会遇到缓存失效或数据不一致的问题。

解决方案：

确保正确配置 Redis 连接字符串和缓存策略。

在数据更新时，确保清除或更新相关的缓存。

**5. 后台作业功能**

问题：在实现后台作业时，可能会遇到作业未能按时执行的问题。

解决方案：

使用 ABP 的 BackgroundJobManager 来调度和管理后台作业。

确保作业逻辑实现了重试和错误处理机制。