**日常APP**

**2025-8-27**

目录

[一、 项目结构：C/S 1](#_Toc211593059)

[二、 技术及框架 1](#_Toc211593060)

[三、 Prism 1](#_Toc211593061)

[四、 WebApi 2](#_Toc211593062)

[五、 项目演示 2](#_Toc211593063)

[1、项目创建及配置 2](#_Toc211593064)

[1.1、DailyAPP： 2](#_Toc211593065)

[1.2、DailyApi: 3](#_Toc211593066)

[2、项目演示 4](#_Toc211593067)

# 项目结构：C/S

# 技术及框架

VS2022 net6.0

WPF(Prism框架, MaterialDesign框架），WebApi(EF,SqlSever,AutoMapper)

# Prism

1. 设计模式：

内置MVVM（Model-View-ViewModel）模式实现，提供BindableBase、DelegateCommand等基础组件，简化界面与业务逻辑分离。

1. 模块化开发：

允许将应用拆分为独立模块（Module），支持按需加载、独立测试和部署，适合团队协作开发。

3. 依赖注入：

集成Unity、MEF等容器，通过构造函数注入实现组件解耦，提升代码可测试性。

4．导航与区域管理：

通过IRegionManager管理界面区域（Region），支持基于URI的导航（如NavigateAsync("ViewA")），并可传递参数。

5. 事件聚合器：

提供IEventAggregator实现组件间跨模块通信，避免直接引用导致的紧耦合。

# WebApi

常用类型：

**RESTful API**：基于REST架构风格，通过URL定位资源，使用HTTP方法（GET/POST/PUT/DELETE）操作资源。

**SOAP API**：基于XML的协议，结构严格、安全性高，常用于企业级服务（如银行接口）。

**核心特点**

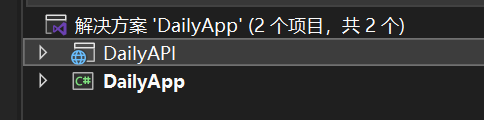
1. **跨平台/语言**：客户端（Web/移动/桌面）和服务端可使用不同技术栈（如前端JavaScript调用后端C#/Java API）。
2. **无状态**：每次请求独立，服务端不存储客户端状态（依赖Token、Session等机制保持会话）。
3. **数据格式**：常用JSON（轻量、易解析）或XML（结构化、适合复杂数据）。

**典型应用场景**

1. **移动应用后端**：APP通过API获取数据（如社交软件的消息流、电商商品列表）。
2. **第三方集成**：例如微信支付API、地图服务API（高德/百度地图）。
3. **前后端分离**：前端（React/Vue）通过API调用后端数据，实现开发解耦。
4. **微服务通信**：分布式系统中，各微服务通过API互相调用（如订单服务调用用户服务）

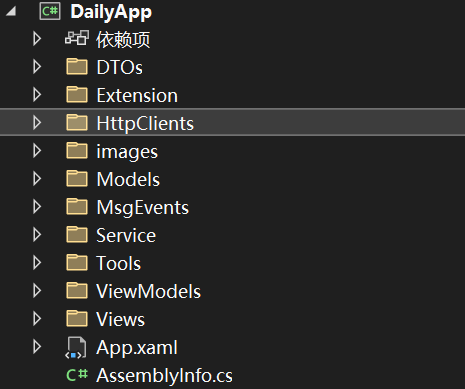
# 项目演示

## 1、项目创建及配置



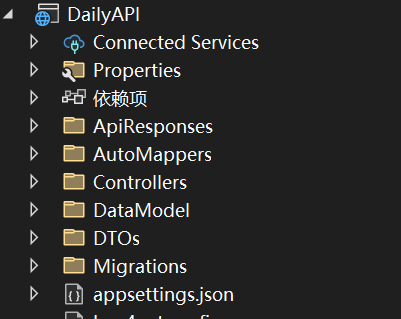
整个工程包含两个项目：DailyAPP(WPF)，DailyAPI(WebApi)

### 1.1、DailyAPP：



* 1. DTOs：封装接口所需字段，减少数据实体字段暴露范围。
  2. HttpClients：存放接口请求模型（路由/请求方式/请求参数/发送数据类型）、接口返回信息模型、接口调用工具类。
  3. Images:存储图片。
  4. Model：存放数据模型。
  5. MsgEvents：存放自定义消息传递类（继承自Prism.Events.PubSubEvent<T>）。
  6. Service：存放自定义服务（例如集成自Prism. Dialogs.DialogService的对话服务）。
  7. Tools：存放工具类（列入文本加密）。
  8. ViewModels：存放视图模型，作为View与Model的中间层，暴露数据属性（供View绑定）和命令（供View触发操作），并处理UI逻辑。
  9. Views：存放视图，通过数据绑定与ViewModel交互，自身不包含业务逻辑。

### 1.2、DailyApi:



1、ApiResponses：存放返回数据模型。

2、AutoMappers：将数据库实体转换为Api响应DTO。

3、Controllers：APi控制器。

4、DataModels：数据模型，生成数据表。

5、DTOs：隔离数据库实体与API接口，优化传输效率并控制字段暴露范围。

## 2、项目演示