Blazor+Dapr+K8s微服务之状态管理

# 状态管理服务器端接口

## 添加Dapr.AspNetCore包

在DaprTest1.Server项目中添加Dapr.AspNetCore包，该包实现了ASP.NET Core与Dapr的集成，例如依赖注入DaprClient对象，将状态管理功能直接集成到 ASP.NET Core 模型绑定 功能中等。

Graphical user interface

Description automatically generated

修改DaprTest1.Server项目的Startup.cs文件，将Dapr对象依赖注入到ASP.NET Core。

services.AddControllersWithViews().AddDapr();

## 状态保存接口

在DaprTest1.Server项目的WeatherForecastController.cs文件中增加状态保存接口。

## 状态删除接口

在DaprTest1.Server项目的WeatherForecastController.cs文件中增加状态删除接口。

## 状态获取接口

在DaprTest1.Server项目的WeatherForecastController.cs文件中增加两个状态获取接口。

注意，第二个状态获取接口是通过ASP.NET Core 模型绑定实现的。

# 状态管理菜单和页面

在DaprTest1.Server项目的NavMenu.razor文件新增状态管理菜单。

在DaprTest1.Server项目新增状态管理页面

# 状态管理测试

和上一节一样，我们先开启每个微服务的SideCar，然后启动两个微服务：

Graphical user interface

Description automatically generated with medium confidence

我们可以查看缓存，看到状态数据，其中缓存Key是由保存状态的微服务ID+“||”+自定义的缓存Key组成，缓存内容Dapr自动增加了版本号。

Graphical user interface, text, application, chat or text message

Description automatically generated