﻿// This controller provides the API endpoint to fetch weather data.

// In a real application, this would interact with a database.

using BlazorWeatherApp.Shared;

using Microsoft.AspNetCore.Mvc;

using System;

using System.Collections.Generic;

using System.Linq;

namespace BlazorWeatherApp.Server.Controllers

{

[ApiController]

[Route("[controller]")]

public class WeatherForecastController : ControllerBase

{

// Define the possible weather summaries, including harsh and dangerous ones.

private static readonly string[] Summaries = new[]

{

"Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy", "Hot", "Sweltering", "Scorching", "Cloudy", "Sunny", "Rainy"

};

// Define multiple locations to demonstrate data across different places.

private static readonly string[] Locations = new[]

{

"Pune", "Bengaluru", "Hyderabad", "Indore", "Mumbai", "Delhi", "Chennai", "Kolkata"

};

[HttpGet]

public IEnumerable<WeatherForecast> Get()

{

var rng = new Random();

var forecasts = new List<WeatherForecast>();

// Generate weather data for each location for 5 days

foreach (var location in Locations)

{

for (int i = 0; i < 5; i++)

{

forecasts.Add(new WeatherForecast

{

Date = DateTime.Now.AddDays(i),

TemperatureC = rng.Next(-10, 45), // Wide range to cover various summaries

Summary = Summaries[rng.Next(Summaries.Length)],

Location = location

});

}

}

// Return the forecasts, ordered by location and then by date for better presentation

return forecasts.OrderBy(f => f.Location).ThenBy(f => f.Date);

}

}

}