

**C** ResponseHandler

- responseBuilder(message: String, status: HttpStatus, data: Object): ResponseEntity<Object>

**C** WeatherController

- weatherService: WeatherService

- WeatherController(weatherService: WeatherService)
- getWeatherDetails(countryId: String): ResponseEntity<Object>
- getAllWeatherDetails(): List<Weather>
- createWeatherDetails(weather: Weather): String
- updateWeatherDetails(weather: Weather): String
- deleteWeatherDetails(countryId: String): String

**I** WeatherService

- createWeather(weather: Weather): String
- updateWeather(weather: Weather): String
- deleteWeather(weatherId: String): String
- getWeather(weatherId: String): Weather
- getAllWeathers(): List<Weather>

**C** WeatherServiceImpl

- weatherRepository: WeatherRepository

- WeatherServiceImpl(weatherRepository: WeatherRepository)
- createWeather(weather: Weather): String
- updateWeather(weather: Weather): String
- deleteWeather(weatherId: String): String
- getWeather(weatherId: String): Weather
- getAllWeathers(): List<Weather>

**I** WeatherRepository

- findByCountryName(countryName: String): List<Weather>

**C** Weather

- countryId: String
- countryName: String
- City: String
- Temperature: String
- WeatherCondition: String

- Weather()
- Weather(countryId: String, countryName: String, City: String, Temperature: String, WeatherCondition: String)
- getCountryId(): String
- setCountryId(countryId: String): void
- getCountryName(): String
- setCountryName(countryName: String): void
- getCity(): String
- setCity(city: String): void
- getTemperature(): String
- setTemperature(temperature: String): void
- getWeatherCondition(): String
- setWeatherCondition(weatherCondition: String): void