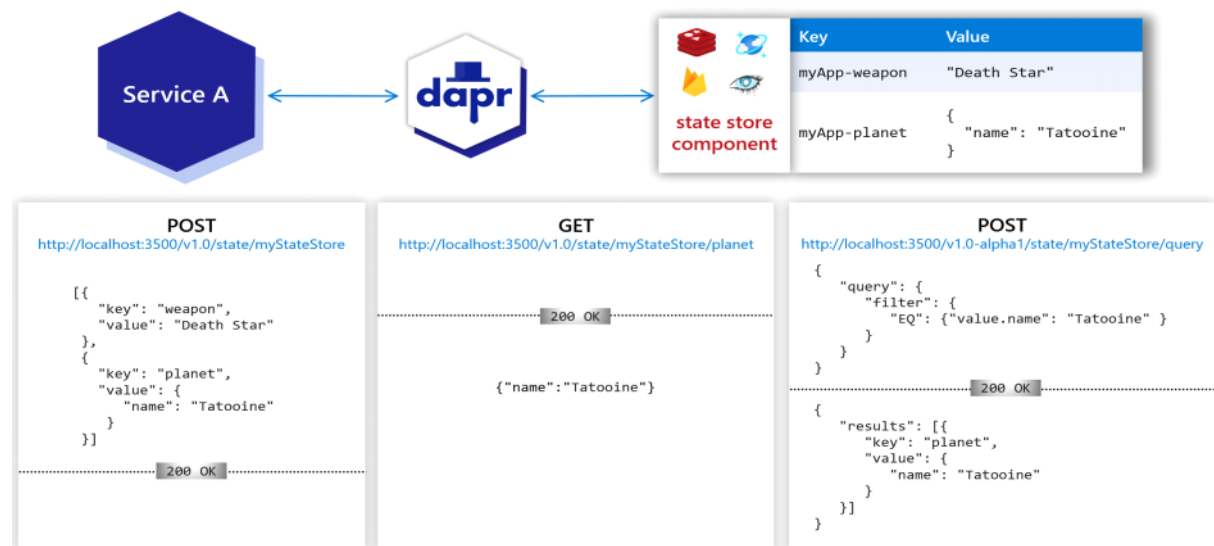


第07期-状态管理构建块

2022年6月29日 10:01

状态管理概述

应用程序可以使用 Dapr 的状态管理 API，使用状态存储组件保存和读取键值对。



Redis on Windows

直接在 Windows 上使用 Redis 缓存

<https://github.com/microsoftarchive/redis>

使用 WSL 在 Windows 上安装 Linux

<https://redis.io/docs/getting-started/installation/install-redis-on-windows>

Redis 设置密码

```
# requirepass foobared
```

配置状态存储组件

<https://github.com/dapr/components-contrib/tree/master/state>

配置状态存储组件

<https://docs.dapr.io/zh-hans/reference/components-reference/supported-state-stores/setup-redis>

stateStore.yaml

```
apiVersion: dapr.io/v1alpha1
kind: Component
metadata:
  name: mystore
  namespace: default
spec:
  type: state.redis
  version: v1
  metadata:
    - name: redisHost
      value: localhost:6379
    - name: redisPassword
      value: ""
```

在 .NET 平台上使用 SDK 管理状态存储

```
<PackageReference Include="Dapr.Client" Version="1.8.0" />
```

```
private async Task UseDaprClientSaveState()
{
    WeatherForecast chongqing = new() { Temperature = 40, Summary = "Hot" };
    WeatherForecast shanghai = new() { Temperature = 26, Summary = "Cool" };

    await _daprClient.SaveStateAsync("mystore", nameof(chongqing),
    chongqing);
    await _daprClient.SaveStateAsync("mystore", nameof(shanghai), shanghai);

    WeatherForecast weather = await
    _daprClient.GetStateAsync<WeatherForecast>("mystore", nameof(shanghai));
}
```

ASP.NET Core 中集成 Dapr 操作

```
<PackageReference Include="Dapr.AspNetCore" Version="1.8.0" />
```

```
builder.Services.AddDaprClient(configureClient =>
{ configureClient.UseGrpcEndpoint("http://localhost:50001"); });
```

```
builder.Services.AddControllers().AddDapr();
```

```
public WeatherForecastsController(DaprClient daprClient)
{
    _daprClient = daprClient;
}

[HttpGet("{city=chongqing}")]
public ActionResult<WeatherForecast> Get([FromState("mystore", "city")]
StateEntry<WeatherForecast> forecast)
{
    if (forecast.Value == null)
    {
        return NotFound();
    }

    return forecast.Value;
}

[HttpPut("{city=shanghai}")]
public async Task<IActionResult> Put(WeatherForecast updatedForecast,
[FromState("mystore", "city")] StateEntry<WeatherForecast> currentForecast)
{
    // update cached current forecast with updated forecast passed into
    service endpoint
    currentForecast.Value = updatedForecast;

    // update state store

    if (await currentForecast.TrySaveAsync())
    {
        return Ok();
    }

    return BadRequest();
}
```

状态存储 API 参考

https://docs.dapr.io/zh-hans/reference/api/state_api

<https://docs.microsoft.com/zh-cn/dotnet/architecture/dapr-for-net-developers/state-management>

状态查询如何使用 SQL 语句过滤器

<https://github.com/dapr/dapr/issues/925>

<https://github.com/dapr/dapr/issues/1339>