在了解该篇文章时，你应该对CQRS有了解

**事件总线**

ABP提供的默认的事件总线

EventBus.Default

获取事件总线推荐使用属性注入

public class TaskAppService : ApplicationService

{

    public IEventBus EventBus { get; set; }

    public TaskAppService()

    {

        EventBus = NullEventBus.Instance;

    }

}

**事件**

**定义事件**

事件必须继承EventData类

public class TaskCompletedEventData : EventData

{

    // 事件ID

    public int TaskId { get; set; }

}

**发送事件**

如下，我们向事件总线发送事件

// 事件类型 TaskCompletedEventData，事件ID 42

EventBus.Trigger(new TaskCompletedEventData {TaskId = 42});

**事件处理器**

向事件总线发送事件后，事件处理器会接收到事件，并处理事件

**定义事件处理器**

事件处理器必须继承IEventHandler

public class ActivityWriter :

    // 处理的事件类型 TaskCompletedEventData

    IEventHandler<TaskCompletedEventData>,

    // 自动注册到 Ioc 容器

    ITransientDependency

{

    // 处理事件

    public void HandleEvent(TaskCompletedEventData eventData)

    {

        WriteActivity("A task is completed by id = " + eventData.TaskId);

    }

}

**处理多事件**

public class ActivityWriter :

    IEventHandler<TaskCompletedEventData>,

    IEventHandler<TaskCreatedEventData>,

    ITransientDependency

{

    // TaskCompletedEventData 类型事件

    public void HandleEvent(TaskCompletedEventData eventData)

    {

    }

    // TaskCreatedEventData 类型事件

    public void HandleEvent(TaskCreatedEventData eventData)

    {

    }

}

**注册事件处理器**

事件处理器需要注册到事件总线上

**自动注册**

自动注册只需要继承IEventHandler和ITransientDependency即可

public class ActivityWriter :

    IEventHandler<TaskCompletedEventData>,

    ITransientDependency

{

    ...

}

**手动注册**

注册：

// 类方式

EventBus.Register<TaskCompletedEventData, ActivityWriter>();

// lambda方式

EventBus.Register<TaskCompletedEventData>(eventData =>

{

    WriteActivity("A task is completed by id = " + eventData.TaskId);

});

**注销事件处理器**

**注销某一处理器**

对应手动注册，可以注册该处理器

//注册到一个事件...

var registration = EventBus.Register<TaskCompletedEventData>(...);

//取消注册事件

registration.Dispose();

**注销某一事件类型的处理器**

使用UnregisterAll注销某一事件类型的处理器