领域服务（或服务）用来执行领域操作和业务规则。Eric Evans描述一个好的服务需要三个特点（在他的DDD书里）：

操作与领域概念（不是一个实体或值对象天生的一部分）相关。

接口要按照领域模型的其它元素来定义。

操作是无状态的。

与[应用服务](https://www.cnblogs.com/kid1412/p/5995248.html" \t "https://www.cnblogs.com/kid1412/p/_blank)获取/返回[DTO](https://www.cnblogs.com/kid1412/p/5995248.html" \t "https://www.cnblogs.com/kid1412/p/_blank)（数据传输对象）不同，领域服务获取/返回领域对象（如[实体](http://www.cnblogs.com/kid1412/p/5992443.html" \t "https://www.cnblogs.com/kid1412/p/_blank)或值类型）。

领域服务可被应用服务或其它领域服务调用，但不直接被展现层（应用服务是针对它的）使用。

**实现领域服务需继承如下**

IDomainService 接口 或 DomainService 服务

**例子**

假设我们有一个任务管理系统，并有分配任务给人的业务规则。

**创建一个接口**

首先，我们为服务定义一个接口（不是必需，但是一个好的实践）：

public interface ITaskManager : **IDomainService**

{

void AssignTaskToPerson(Task task, Person person);

}

**实现服务**

让我们看一下实现：

public class TaskManager : **DomainService**, **ITaskManager**

{

public void AssignTaskToPerson(Task task, Person person)

{

}

}

**使用应用服务**

现在，让我们看一下一个应用服务如何使用TaskManager服务：

public class TaskAppService : ApplicationService, ITaskAppService

{

**private readonly ITaskManager \_taskManager;**

public TaskAppService(**ITaskManager taskManager**)

{

\_taskManager = taskManager;

}

public void AssignTaskToPerson(AssignTaskToPersonInput input)

{

**\_taskManager.AssignTaskToPerson(task, person);**

}

}