**仓库必须工作在工作单元中**

**默认仓储**

在ABP里，一个仓储类实现IRepository<TEntity,TPrimaryKey>接口。你可以直接注入IRepository<TEntity>（或IRepository<TEntity,TPrimaryKey>）。

public class PersonAppService : IPersonAppService

{

**private readonly IRepository<Person> \_personRepository;**

public PersonAppService(**IRepository<Person> personRepository**)

{

\_personRepository = personRepository;

}

public void CreatePerson(CreatePersonInput input)

{

person = new Person { Name = input.Name, EmailAddress = input.EmailAddress };

**\_personRepository.Insert(person);**

}

}

**自定义仓储接口**

如下示例，为一个Person实体定义一个仓储：

public interface IPersonRepository : IRepository<Person>

{

}

如果你的实体键不是int，你可以扩展IRepository<TEntity,TPrimaryKey>接口，如下所示：

public interface IPersonRepository : IRepository<Person, long>

{

}

**基仓储方法**

**自定义返回值**

还有一个另外的方法提供更强大的IQueryable，可以用在工作单元之外。

T Query<T>(Func<IQueryable<TEntity>, T> queryMethod);

Query方法接受一个lambda表达式（或方法），该表达式（或方法）接收IQueryable<T>并返回任何类型的对象。例如：

var people = \_personRepository.Query(q => q.Where(p => p.Name.Contains("H")).OrderBy(p => p.Name).ToList());

**管理数据库连接**

在一个仓储方法里，它不打开或关闭数据库连接，ABP自动管理数据库连接。

当进入一个仓储方法，ABP自动打开一个数据库连接并开始一个事务，当这个方法结束并返回时，所有的变化被保存，事务提交后关闭数据库连接。