**Nuget 包**

在ABP里使用EF的Nuget包是Abp.EntityFramework，你应该把它加入到你的应用里，最好在你项目里单独建立一个EF程序集（dll），然后依赖该于这个包。

**默认仓储**

Abp.EntityFramework为所有定义在你的DbContext里的实体，实现了默认的仓储，你不用创建仓储类，就可以直接使用预定义的仓储方法，例如：

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);**

}

}

**特定的仓储基类**

ABP提供了一个基类EfRepositoryBase，可以很容易地实现仓储。一个SimpleTaskSystem应用的所有仓储的基类示例：

//Base class for all repositories in my application

public class SimpleTaskSystemRepositoryBase<TEntity, TPrimaryKey> : **EfRepositoryBase<SimpleTaskSystemDbContext, TEntity, TPrimaryKey>**

where TEntity : class, IEntity<TPrimaryKey>

{

Public SimpleTaskSystemRepositoryBase(**IDbContextProvider<SimpleTaskSystemDbContext> dbContextProvider**): base(dbContextProvider)

{

}

}

public class SimpleTaskSystemRepositoryBase<TEntity> :

**SimpleTaskSystemRepositoryBase<TEntity, int>**

where TEntity : class, IEntity<int>

{

public SimpleTaskSystemRepositoryBase(**IDbContextProvider<SimpleTaskSystemDbContext> dbContextProvider**)

: base(dbContextProvider)

{

}

}

**自定义仓储示例**

为实现一个自定义仓储，可以直接继承你应用里特定的仓储基类（如我们上面创建的）。

假设我们有一个Task实体，它可以分配给一个Person（实体），并且一个Task有一个State（new，assigned，completed...），我们需要编写一个自定义方法，通过一些条件和预先获取的AssisgnedPerson属性使用一条数据库查询，获取Tasks列表，示例代码如下：

public interface ITaskRepository : **IRepository<Task, long>**

{

List<Task> GetAllWithPeople(int? assignedPersonId, TaskState? state);

}

public class TaskRepository : **SimpleTaskSystemRepositoryBase<Task, long>, ITaskRepository**

{

public TaskRepository(IDbContextProvider<SimpleTaskSystemDbContext> dbContextProvider)

: base(dbContextProvider)

{

}

public List<Task> GetAllWithPeople(int? assignedPersonId, TaskState? state)

{

}

}