在线考试系统系统

开发文档

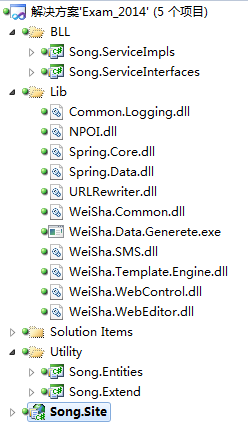
# 概述

系统基于.Net 4.0开发，采用C#语言，开发工具为Microsoft Visual Studio 2010；数据库采用了Sqlserver2005。

整个系统基于B/S架构，采用了传统的三层开发模型，数据层采用了ORM，使用的是Mysoft.Data的组件；业务层与表现层之间用了spring.net的IOC组件；表现层采用了Vtemplate组件进行模型与控制、视图的分离，可以理解为一种MVC的解决方案吧。后面会对每部分进行详述。

# 解决方案描述

在解决方案中，共分为五个项目。如下图



## 业务层

BLL解决方案文件夹下，是业务层的两个项目；

Song.ServiceInterfaces是所有业务层的接口

Song.ServiceImpls 是业务层（Song.ServiceInterfaces）的接口实现。

之所以分成两个项目，是为了基于面向接口编程，例如当业务层代码需要变动时，只需要改动实现，不用改动接口。实口与实现类的关系是在Song.Site项目中的web.config中配置的。如下图



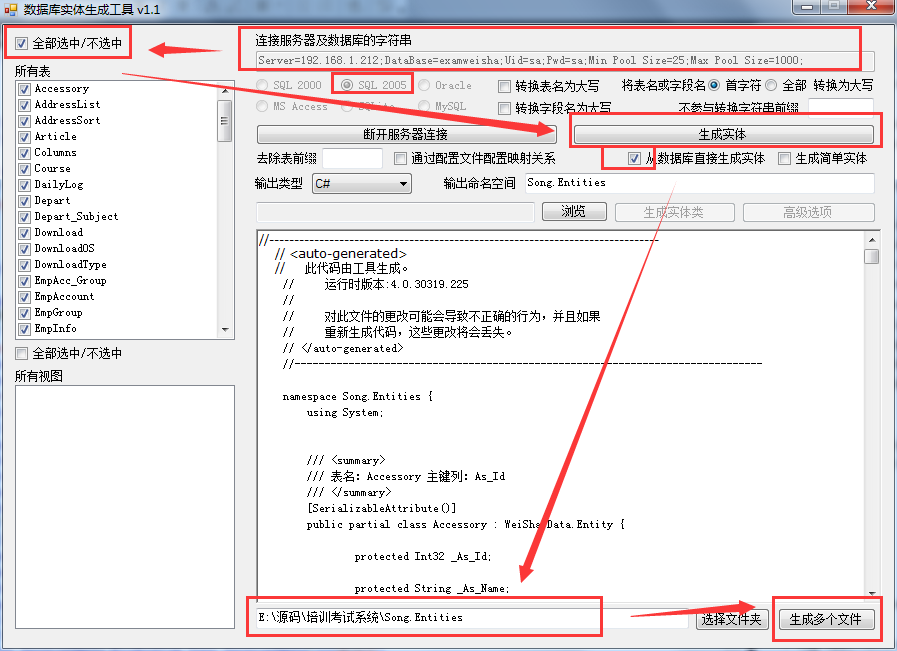
上图定义了接口名称，以及其实现类。这里是用了Spring.net的IOC（控制反转）组件。

## 数据层

Song.Entities是数据实体。

数据层的处理采用了ORM方式处理，这里是用了Mysoft.Data开源组件，网上可以找到下载。使用方法如下：

在Lib解决方案文件夹下，有WeiSha.Data.Generete.exe，双击打开。如下图步骤操作。



生成实体后，在Song.ServiceImpls项目中引用Song.Entities

在编写时，例如增加一个记录

/// <summary>

/// 添加

/// </summary>

/// <param name="entity">业务实体</param>

public void Add(Accessory entity)

{

entity.As\_CrtTime = DateTime.Now;

Song.Entities.Organization org = Business.Do<IOrganization>().OrganCurrent();

if (org != null)

{

entity.Org\_ID = org.Org\_ID;

entity.Org\_Name = org.Org\_Name;

}

Gateway.Default.Save<Accessory>(entity);

}

其中关键性语句是 Gateway.Default.Save<Accessory>(entity);

## 表现层

Song.Site是表现层，主要是aspx网页，Js，css等文件构成。

当在表现层调用业务层代码时，采用如下示例代码

Song.Entities.Subject[] subj = Business.Do<ISubject>().SubjectCount(Organ.Org\_ID, true, 0);

上述代码是调用专业的接品，取当前机构下所有专业。

**Vtemplate组件的使用**

Vtemplate组件实现了模型与视图的分离，可以通过它来实现更换模板风格等操作。

使用方法是，在Song.Site项目中的.ashx文件继承于Basepage.cs类，Basepage实现了一个模型与视图的匹配控制。例如About.ashx的视图是指向Templates/(当前选中的模板库)/ About.htm

## 其它项目

Song.Extend是一些扩展类，其中包括用户登录等。