文档编号 SS-STG-003

Sextant Tour Guide 软件设计报告

**系统组件化**

版本 1.0 （第1稿）

Sextant Studio

2013/04/17

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **日期** | **修订者** | **说明** |
| 1.0 | 2013/04/06 | 郭旭 | 创建 |
| 1.0 | 2013/04/15 | 温斌 | 编写 |
| 1.0 | 2013/04/17 | 郭旭 | 修改 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[Sextant Tour Guide 软件设计报告 1](#_Toc353960652)

[修订记录 2](#_Toc353960653)

[目录 3](#_Toc353960654)

[**一、** **引言** 4](#_Toc353960655)

[1.1 编写目的 4](#_Toc353960656)

[1.2 项目背景 4](#_Toc353960657)

[**二、** **系统架构** 4](#_Toc353960658)

[2.1 架构 4](#_Toc353960659)

[2.2 数据建模 5](#_Toc353960660)

[2.3 引用关系 5](#_Toc353960661)

[**三、** **组件化** 5](#_Toc353960662)

[3.1 组件化说明 5](#_Toc353960663)

[3.2 组件接口设计 5](#_Toc353960664)

[3.3 配置文件 39](#_Toc353960665)

[**四、** **其它** 40](#_Toc353960666)

[4.1 AOP 40](#_Toc353960668)

[4.2 日志 40](#_Toc353960669)

[4.3 数据连接层 40](#_Toc353960670)

1. **引言**
   1. 编写目的

本文档对Sextant Tour Guide旅游软件组件化设计进行详细说明。

* 1. 项目背景

本项目作为《软件工程技术与设计》课程的训练项目提出，由Sextant Studio小组进行实际开发。通过Sextant Tour Guide软件的设计开发，了解软件文档的编写原则，训练掌握软件各类文档写作的技巧，同时提高和增加C#开发的能力和经验。项目开发平台为Visual Studio 2010 (C# .Net 3.5)。

1. **系统架构**
   1. 架构

目前的系统虽然采用C/S架构，由于作业未作要求，因此服务端和客户端暂未分离，程序、服务、数据库共同在一台机器上运行。系统严格按照三层架构设计，包括数据连接层、服务（业务逻辑）层、用户界面层，各层之间通过接口进行访问，以降低耦合性。在访问各层的接口时一般使用简单工厂模式进行类的初始化；工厂类使用.NET的反射性质，使得dll文件在程序运行过程中根据配置文件进行动态加载。



图 1 架构与分工

* 1. 数据建模

数据建模是在项目初期需要解决的一个较麻烦的问题。考虑到小组成员对.NET开发环境比较陌生，对于复杂的面向对象开发也不是很了解，而且本系统的业务逻辑又比较简单，因此没有采用NHibernate等ORM工具，也放弃了最为“面向对象”的领域模型（Domain Model）的建模方式，而是采用比较简洁的活动记录（Active Record）方式，即模型中的实体类与数据库的表一一对应，实体类中的公共属性与表中的字段一一对应；并且使用贫血（失血）模型，即实体类中仅有属性，没有任何方法，以最大程度地降低与数据访问层和业务逻辑的耦合。实体类与数据库的映射采用SQL语句硬编码的方式进行映射。

* 1. 引用关系



图 2 项目(包)间关系

1. **组件化**
   1. 组件化说明

本系统的组件化原则上按照系统的分层架构进行，主要包括：数据库访问组件、服务组件、日志组件，这几个组件均向外部提供接口以进行访问；此外，还包括一个封装了ADO.NET的数据库访问组件，以及在用户界面层实现的可编辑图片列表组件和只读图片列表组件，这几个组件由于不具有通用性，因此没有使用接口，而采用直接向外暴露公共方法以进行访问。在系统中通过读取配置文件来确定实例化组件接口的方式，配置文件将在下文进行说明。

* 1. 组件接口设计

（1）数据连接层接口

|  |  |
| --- | --- |
| 类 | DALFactory |
| 说明 | 数据连接层工厂类 |
| 接口 | /// <summary>  /// 创建一个数据连接层的实例  /// </summary>  /// <typeparam name="T">指定的数据类型</typeparam>  /// <returns>指定数据类型的实例</returns>  T CreateDAL<T>() where T : IBaseDAL |
|  |  |
| 类 | IBlogDAL |
| 说明 | 访问blog |
| 接口 | /// <summary>  /// 获取全部日志  /// </summary>  /// <returns>日志列表</returns>  List<Blog> GetBlogs();  /// <summary>  /// 获取指定用户的日志  /// </summary>  /// <param name="userId">用户ID</param>  /// <returns>日志列表</returns>  List<Blog> GetBlogsByUserId(string userId);  /// <summary>  /// 获取指定景点的日志  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <returns>日志列表</returns>  List<Blog> GetBlogsBySightsId(string sightsId);  /// <summary>  /// 获取指定用户与景点的日志  /// </summary>  /// <param name="userId">用户ID</param>  /// <param name="sightsId">景点ID</param>  /// <returns>日志列表</returns>  List<Blog> GetBlogsByUserIdAndSightsId(string userId, string sightsId);  /// <summary>  /// 获取指定旅行的日志  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <returns>日志列表</returns>  List<Blog> GetBlogsByTourId(string tourId);  /// <summary>  /// 获取指定旅行中一次子旅行的日志  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <param name="subTourId">子旅行ID</param>  /// <returns>日志列表</returns>  List<Blog> GetBlogsByTourIdAndSubTourId(string tourId, string subTourId);  /// <summary>  /// 获取指定日志  /// </summary>  /// <param name="blogId">日志ID</param>  /// <returns>日志</returns>  Blog GetBlogByBlogId(string blogId);  /// <summary>  /// 向数据库插入一篇日志  /// </summary>  /// <param name="blog">日志</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int InsertBlog(Blog blog, DbTransaction trans);  /// <summary>  /// 更新数据库中的一篇日志  /// </summary>  /// <param name="newBlog">新的日志</param>  /// <param name="oldBlog">旧的日志</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int UpdateBlogFromOld(Blog newBlog, Blog oldBlog, DbTransaction trans);  /// <summary>  /// 删除数据库中的一条日志  /// </summary>  /// <param name="blog">日志</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int DeleteBlog(Blog blog, DbTransaction trans); |
|  |  |
| 类 | ICityDAL |
| 说明 | 访问城市 |
| 接口 | /// <summary>  /// 获取所有城市  /// </summary>  /// <returns>城市列表</returns>  List<City> GetCities();  /// <summary>  /// 向数据库中插入一个城市  /// </summary>  /// <param name="city">城市</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int InsertCity(City city, DbTransaction trans);  /// <summary>  /// 更新数据库中的一个城市  /// </summary>  /// <param name="city">新的城市</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int UpdateCity(City city, DbTransaction trans);  /// <summary>  /// 删除数据库中的一个城市  /// </summary>  /// <param name="city">城市</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int DeleteCity(City city, DbTransaction trans); |
|  |  |
| 类 | ICountryDAL |
| 说明 | 访问国家 |
| 接口 | /// <summary>  /// 获取所有的国家  /// </summary>  /// <returns></returns>  List<Country> GetCountries();  /// <summary>  /// 向数据库中插入一个国家  /// </summary>  /// <param name="country">国家</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int InsertCountry(Country country, DbTransaction trans);  /// <summary>  /// 更新数据库中的一个国家  /// </summary>  /// <param name="country">新的国家</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int UpdateCountry(Country country, DbTransaction trans);  /// <summary>  /// 删除数据库中的一个国家  /// </summary>  /// <param name="country">国家</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int DeleteCountry(Country country, DbTransaction trans); |
|  |  |
| 类 | IFavoriteDAL |
| 说明 | 访问收藏夹 |
| 接口 | /// <summary>  /// 获取所有用户收藏  /// </summary>  /// <returns>用户收藏列表</returns>  List<Favorite> GetFavorites();  /// <summary>  /// 获取指定用户的收藏  /// </summary>  /// <param name="userId">用户ID</param>  /// <returns>用户收藏列表</returns>  List<Favorite> GetFavoritesByUserId(string userId);  /// <summary>  /// 获取指定景点的用户收藏  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <returns>用户收藏列表</returns>  List<Favorite> GetFavoritesBySightsId(string sightsId);  /// <summary>  /// 获取指定用户对于指定景点的收藏  /// </summary>  /// <param name="userId">用户ID</param>  /// <param name="sightsId">景点ID</param>  /// <returns>用户收藏</returns>  Favorite GetFavoriteByUserIdAndSightsId(string userId, string sightsId);  /// <summary>  /// 获取指定景点的平均评分  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <returns>平均评分</returns>  float GetAverageStarsBySightsId(string sightsId);  /// <summary>  /// 向数据库中插入一条用户收藏  /// </summary>  /// <param name="favorite">用户收藏</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int InsertFavorite(Favorite favorite, DbTransaction trans);  /// <summary>  /// 更新数据库中的一条用户收藏  /// </summary>  /// <param name="favorite">新的用户收藏</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int UpdateFavorite(Favorite favorite, DbTransaction trans);  /// <summary>  /// 删除数据库中的一条用户收藏  /// </summary>  /// <param name="favorite">用户收藏</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int DeleteFavorite(Favorite favorite, DbTransaction trans); |
|  |  |
| 类 | IPermissionDAL |
| 说明 | 访问用户权限 |
| 接口 | /// <summary>  /// 获取指定用户的权限  /// </summary>  /// <param name="userId">用户ID</param>  /// <returns>权限列表</returns>  List<Permission> GetPermissionsByUserId(string userId);  /// <summary>  /// 向数据库中插入一条用户权限  /// </summary>  /// <param name="permission">用户权限</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int InsertPermission(Permission permission, DbTransaction trans);  /// <summary>  /// 删除数据库中的一条用户权限  /// </summary>  /// <param name="permission">用户权限</param>  /// <param name="trans"></param>  /// <returns>数据库中发生变更的行数</returns>  int DeletePermission(Permission permission, DbTransaction trans); |
|  |  |
| 类 | IPictureCommentDAL |
| 说明 | 访问图片评论 |
| 接口 | /// <summary>  /// 获取指定图片的评论  /// </summary>  /// <param name="pictureId">图片ID</param>  /// <returns>图片评论列表</returns>  List<PictureComment> GetPictureCommentsByPictureId(string pictureId);  /// <summary>  /// 向数据库中插入一条图片评论  /// </summary>  /// <param name="comment">图片评论</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int InsertPictureComment(PictureComment comment, DbTransaction trans);  /// <summary>  /// 更新数据库中的一条图片评论  /// </summary>  /// <param name="comment">新的图片评论</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int UpdatePictureComment(PictureComment comment, DbTransaction trans);  /// <summary>  /// 删除数据库中的一条图片评论  /// </summary>  /// <param name="comment">图片评论</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int DeletePictureComment(PictureComment comment, DbTransaction trans); |
|  |  |
| 类 | IPictureDAL |
| 说明 | 访问图片 |
| 接口 | /// <summary>  /// 获取指定用户上传的关于指定景点的图片  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <param name="uploaderId">用户ID</param>  /// <returns>图片列表</returns>  List<Picture> GetPicturesBySightsIdAndUploaderId(string sightsId, string uploaderId);  /// <summary>  /// 获取指定景点的图片  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <returns>图片列表</returns>  List<Picture> GetPicturesBySightsId(string sightsId);  /// <summary>  /// 获取指定旅行的图片  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <returns>图片列表</returns>  List<Picture> GetPicturesByTourId(string tourId);  /// <summary>  /// 获取指定旅行中一次子旅行的图片  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <param name="subTourId">子旅行ID</param>  /// <returns>图片列表</returns>  List<Picture> GetPicturesByTourIdAndSubTourId(string tourId, string subTourId);  /// <summary>  /// 获取指定图片  /// </summary>  /// <param name="picId">图片ID</param>  /// <returns>图片</returns>  Picture GetPictureByPictureId(string picId);  /// <summary>  /// 向数据库中插入一条图片记录  /// </summary>  /// <param name="pic">图片</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int InsertPicture(Picture pic, DbTransaction trans);  /// <summary>  /// 更新数据库中的一条图片记录  /// </summary>  /// <param name="pic">新的图片</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int UpdatePicture(Picture pic, DbTransaction trans);  /// <summary>  /// 删除数据库中的一条图片记录  /// </summary>  /// <param name="pic">图片</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int DeletePicture(Picture pic, DbTransaction trans) |
|  |  |
| 类 | IProvinceDAL |
| 说明 | 访问省份 |
| 接口 | /// <summary>  /// 获取所有  /// </summary>  /// <returns>列表</returns>  List<Province> GetProvinces();  /// <summary>  /// 向数据库中插入一条记录  /// </summary>  /// <param name="province"></param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int InsertProvince(Province province, DbTransaction trans);  /// <summary>  /// 更新数据库中的一条记录  /// </summary>  /// <param name="province">新的</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int UpdateProvince(Province province, DbTransaction trans);  /// <summary>  /// 删除数据库中的一条记录  /// </summary>  /// <param name="province"></param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int DeleteProvince(Province province, DbTransaction trans); |
|  |  |
| 类 | ISightsCommentDAL |
| 说明 | 访问景点评论 |
| 接口 | /// <summary>  /// 获取指定景点的评论  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <returns>景点评论列表</returns>  List<SightsComment> GetSightsCommentsBySightsId(string sightsId);  /// <summary>  /// 向数据库中插入一条景点评论  /// </summary>  /// <param name="comment">景点评论</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int InsertSightsComment(SightsComment comment, DbTransaction trans);  /// <summary>  /// 更新数据库中的一条景点评论  /// </summary>  /// <param name="comment">景点评论</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int UpdateSightsComment(SightsComment comment, DbTransaction trans);  /// <summary>  /// 删除数据库中的一条景点评论  /// </summary>  /// <param name="comment">景点评论</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int DeleteSightsComment(SightsComment comment, DbTransaction trans); |
|  |  |
| 类 | ISightsDAL |
| 说明 | 访问景点 |
| 接口 | /// <summary>  /// 获取全部景点  /// </summary>  /// <returns>景点列表</returns>  List<Sights> GetSights();  /// <summary>  /// 获取指定城市的景点  /// </summary>  /// <param name="cityId">城市ID</param>  /// <returns>景点列表</returns>  List<Sights> GetSightsByCityId(string cityId);  /// <summary>  /// 获取指定的景点  /// </summary>  /// <param name="provinceId">ID</param>  /// <returns>景点列表</returns>  List<Sights> GetSightsByProvinceId(string provinceId);  /// <summary>  /// 获取指定国家的景点  /// </summary>  /// <param name="countryId">国家ID</param>  /// <returns>景点列表</returns>  List<Sights> GetSightsByCountryId(string countryId);  /// <summary>  /// 获取指定景点  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <returns>景点</returns>  Sights GetSightBySightsId(string sightsId);  /// <summary>  /// 向数据库中插入一条景点信息  /// </summary>  /// <param name="sights">景点</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int InsertSights(Sights sights, DbTransaction trans);  /// <summary>  /// 更新数据库中的一条景点信息  /// </summary>  /// <param name="newItem">新的景点</param>  /// <param name="oldItem">旧的景点</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int UpdateSightsFromOld(Sights newItem, Sights oldItem, DbTransaction trans);  /// <summary>  /// 删除数据库中的一条景点信息  /// </summary>  /// <param name="sights">景点信息</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int DeleteSights(Sights sights, DbTransaction trans); |
|  |  |
| 类 | ISubTourDAL |
| 说明 | 访问子行程 |
| 接口 | /// <summary>  /// 获取指定旅行的所有子旅行  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <returns>子旅行列表</returns>  List<SubTour> GetSubToursByTourId(string tourId);  /// <summary>  /// 获取指定旅行中一次指定的子旅行  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <param name="subTourId">子旅行ID</param>  /// <returns>子旅行</returns>  SubTour GetSubTourByTourIdAndSubTourId(string tourId, string subTourId);  /// <summary>  /// 获取指定用户的子旅行  /// </summary>  /// <param name="userId">用户ID</param>  /// <returns>子旅行列表</returns>  List<SubTour> GetSubToursByUserId(string userId);  /// <summary>  /// 获取指定景点所有的子旅行  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <returns>子旅行列表</returns>  List<SubTour> GetSubToursBySightsId(string sightsId);  /// <summary>  /// 向数据库中插入一条子旅行  /// </summary>  /// <param name="subTour">子旅行</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int InsertSubTour(SubTour subTour, DbTransaction trans);  /// <summary>  /// 更新数据库中的一条子旅行  /// </summary>  /// <param name="newItem">新的子旅行</param>  /// <param name="oldItem">旧的子旅行</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int UpdateSubTourFromOld(SubTour newItem, SubTour oldItem, DbTransaction trans);  /// <summary>  /// 删除数据库中的一条子旅行  /// </summary>  /// <param name="subTour">子旅行</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int DeleteSubTour(SubTour subTour, DbTransaction trans); |
|  |  |
| 类 | ITourCommentDAL |
| 说明 | 访问旅行评论 |
| 接口 | /// <summary>  /// 获取指定旅行的评论  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <returns>旅行评论列表</returns>  List<TourComment> GetTourCommentsByTourId(string tourId);  /// <summary>  /// 向数据库中插入一条旅行评论  /// </summary>  /// <param name="comment">旅行评论</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int InsertTourComment(TourComment comment, DbTransaction trans);  /// <summary>  /// 更新数据库中的一条旅行评论  /// </summary>  /// <param name="comment">旅行评论</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int UpdateTourComment(TourComment comment, DbTransaction trans);  /// <summary>  /// 删除数据库中的一条旅行评论  /// </summary>  /// <param name="comment">旅行评论</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int DeleteTourComment(TourComment comment, DbTransaction trans); |
|  |  |
| 类 | ITourDAL |
| 说明 | 访问旅行 |
| 接口 | /// <summary>  /// 获取指定用户的所有旅行  /// </summary>  /// <param name="userId">用户ID</param>  /// <returns>旅行列表</returns>  List<Tour> GetToursByUserId(string userId);  /// <summary>  /// 获取指定的旅行  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <returns>旅行</returns>  Tour GetTourByTourId(string tourId);  /// <summary>  /// 获取指定景点相关的旅行  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <returns>景点</returns>  List<Tour> GetToursBySightsId(string sightsId);  /// <summary>  /// 获取包含指定日期的旅行  /// </summary>  /// <param name="date">日期</param>  /// <returns>旅行列表</returns>  List<Tour> GetToursByDate(DateTime date);  /// <summary>  /// 获取与指定景点相关，包含指定日期的旅行  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <param name="date">日期</param>  /// <returns>旅行列表</returns>  List<Tour> GetToursBySightsIdAndDate(string sightsId, DateTime date);  /// <summary>  /// 向数据库中插入一条旅行记录  /// </summary>  /// <param name="tour">旅行</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int InsertTour(Tour tour, DbTransaction trans);  /// <summary>  /// 更新数据中的一条旅行记录  /// </summary>  /// <param name="newItem">新的旅行</param>  /// <param name="oldItem">旧的旅行</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int UpdateTourFromOld(Tour newItem, Tour oldItem, DbTransaction trans);  /// <summary>  /// 删除数据库中的一条旅行记录  /// </summary>  /// <param name="tour">旅行</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int DeleteTour(Tour tour, DbTransaction trans); |
|  |  |
| 类 | IUserCommentDAL |
| 说明 | 访问用户评论 |
| 接口 | /// <summary>  /// 获取针对指定用户的评论  /// </summary>  /// <param name="userId">用户ID</param>  /// <returns>用户评论列表</returns>  List<UserComment> GetUserCommentsByUserId(string userId);  /// <summary>  /// 向数据库中插入一条用户评论  /// </summary>  /// <param name="comment">用户评论</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int InsertUserComment(UserComment comment, DbTransaction trans);  /// <summary>  /// 更新数据库中的一条用户评论  /// </summary>  /// <param name="comment">用户评论</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int UpdateUserComment(UserComment comment, DbTransaction trans);  /// <summary>  /// 删除数据库中的一条用户评论  /// </summary>  /// <param name="comment">用户评论</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int DeleteUserComment(UserComment comment, DbTransaction trans); |
|  |  |
| 类 | IUserDAL |
| 说明 | 访问用户 |
| 接口 | /// <summary>  /// 获取所有的用户  /// </summary>  /// <returns>用户列表</returns>  List<User> GetUsers();  /// <summary>  /// 获取指定Email的用户  /// </summary>  /// <param name="email">Email</param>  /// <returns>用户</returns>  User GetUserByEmail(string email);  /// <summary>  /// 获取指定登录名的用户  /// </summary>  /// <param name="loginName">登录名</param>  /// <returns>用户</returns>  User GetUserByLoginName(string loginName);  /// <summary>  /// 获取指定登录名与口令的用户  /// </summary>  /// <param name="loginName">登录名</param>  /// <param name="password">口令</param>  /// <returns>用户</returns>  User GetUserByLoginNameAndPassword(string loginName, string password);  /// <summary>  /// 获取指定用户  /// </summary>  /// <param name="userId">用户ID</param>  /// <returns>用户</returns>  User GetUserByUserId(string userId);  /// <summary>  /// 向数据库中插入一条用户记录  /// </summary>  /// <param name="user">用户</param>  /// <param name="password">登录口令</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int InsertUser(User user, string password, DbTransaction trans);  /// <summary>  /// 更新数据库中的一条用户记录  /// </summary>  /// <param name="user">用户</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int UpdateUser(User user, DbTransaction trans);  /// <summary>  /// 更新数据库中的一个用户口令  /// </summary>  /// <param name="user">用户</param>  /// <param name="newPassword">新口令</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int UpdateUserPassword(User user, string newPassword, DbTransaction trans);  /// <summary>  /// 删除数据库中的一条用户记录  /// </summary>  /// <param name="user">用户</param>  /// <param name="trans">数据库事务</param>  /// <returns>数据库中发生变更的行数</returns>  int DeleteUser(User user, DbTransaction trans); |

（2）服务层接口

|  |  |
| --- | --- |
| 类 | ServiceFactory |
| 说明 | 服务工程类 |
| 接口 | /// <summary>  /// 创建一个服务（业务逻辑）层的实例  /// </summary>  /// <typeparam name="T"></typeparam>  /// <returns></returns>  public static T CreateService<T>() where T : IBaseService |
|  |  |
| 类 | IBlogService |
| 说明 | 提供blog服务接口 |
| 接口 | /// <summary>  /// 获取指定用户的所有日志  /// </summary>  /// <param name="userId">用户ID</param>  /// <returns>日志列表</returns>  List<Blog> GetBlogsByUserId(string userId);  /// <summary>  /// 获取指定景点的所有日志  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <returns>日志列表</returns>  List<Blog> GetBlogsBySightsId(string sightsId);  /// <summary>  /// 获取指定用户与指定景点的所有日志  /// </summary>  /// <param name="userId">用户ID</param>  /// <param name="sightsId">景点ID</param>  /// <returns>日志列表</returns>  List<Blog> GetBlogsByUserIdAndSightsId(string userId, string sightsId);  /// <summary>  /// 获取指定旅行的所有相关日志  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <param name="subTourId">子旅行ID</param>  /// <returns>日志列表</returns>  List<Blog> GetBlogsByTourIdAndSubTourId(string tourId, string subTourId);  /// <summary>  /// 获取指定日志  /// </summary>  /// <param name="blogId">日志ID</param>  /// <returns>日志</returns>  Blog GetBlogByBlogId(string blogId);  /// <summary>  /// 获取指定旅行的所有相关日志  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <returns>日志列表</returns>  List<Blog> GetBlogsByTourId(string tourId);  /// <summary>  /// 删除指定日志  /// </summary>  /// <param name="blog">日志</param>  /// <param name="deletePictures">是否删除与该日志相关的图片</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool DeleteBlog(Blog blog, bool deletePictures, out string message);  /// <summary>  /// 保存日志  /// </summary>  /// <param name="blog">日志</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool SaveBlob(Blog blog, out string message); |
|  |  |
| 类 | ICommentService |
| 说明 | 提供评论接口 |
| 接口 | /// <summary>  /// 获取指定图片的评论  /// </summary>  /// <param name="pictureId">图片ID</param>  /// <returns>图片评论列表</returns>  List<PictureComment> GetPictureCommentsByPictureId(string pictureId);  /// <summary>  /// 获取指定景点的评论  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <returns>景点评论列表</returns>  List<SightsComment> GetSightsCommentsBySightsId(string sightsId);  /// <summary>  /// 获取指定旅行的评论  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <returns>旅行评论列表</returns>  List<TourComment> GetTourCommentsByTourId(string tourId);  /// <summary>  /// 获取关于指定用户的评论  /// </summary>  /// <param name="userId">用户ID</param>  /// <returns>用户评论列表</returns>  List<UserComment> GetUserCommentsByUserId(string userId);  /// <summary>  /// 新增一条图片评论  /// </summary>  /// <param name="comment">图片评论</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool InsertPictureComment(PictureComment comment, out string message);  /// <summary>  /// 更新一条图片评论  /// </summary>  /// <param name="comment">图片评论</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool UpdatePictureComment(PictureComment comment, out string message);  /// <summary>  /// 删除一条图片评论  /// </summary>  /// <param name="comment">图片评论</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool DeletePictureComment(PictureComment comment, out string message);  /// <summary>  /// 新增一条景点评论  /// </summary>  /// <param name="comment">景点评论</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool InsertSightsComment(SightsComment comment, out string message);  /// <summary>  /// 更新一条景点评论  /// </summary>  /// <param name="comment">景点评论</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool UpdateSightsComment(SightsComment comment, out string message);  /// <summary>  /// 删除一条景点评论  /// </summary>  /// <param name="comment">景点评论</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool DeleteSightsComment(SightsComment comment, out string message);  /// <summary>  /// 新增一条旅行评论  /// </summary>  /// <param name="comment">旅行评论</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool InsertTourComment(TourComment comment, out string message);  /// <summary>  /// 更新一条旅行评论  /// </summary>  /// <param name="comment">旅行评论</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool UpdateTourComment(TourComment comment, out string message);  /// <summary>  /// 删除一条旅行评论  /// </summary>  /// <param name="comment">旅行评论</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool DeleteTourComment(TourComment comment, out string message);  /// <summary>  /// 新增一条用户评论  /// </summary>  /// <param name="comment">用户评论</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool InsertUserComment(UserComment comment, out string message);  /// <summary>  /// 更新一条用户评论  /// </summary>  /// <param name="comment">用户评论</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool UpdateUserComment(UserComment comment, out string message);  /// <summary>  /// 删除一条用户评论  /// </summary>  /// <param name="comment">用户评论</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool DeleteUserComment(UserComment comment, out string message); |
|  |  |
| 类 | IDictService |
| 说明 | 提供字典服务接口 |
| 接口 | /// <summary>  /// 获取所有国家  /// </summary>  /// <returns>国家列表</returns>  List<Country> GetCountries();  /// <summary>  /// 获取所有  /// </summary>  /// <returns>列表</returns>  List<Province> GetProvinces();  /// <summary>  /// 获取所有城市  /// </summary>  /// <returns>城市列表</returns>  List<City> GetCities();  /// <summary>  /// 获取指定国家的  /// </summary>  /// <param name="countryId">国家ID</param>  /// <returns>列表</returns>  List<Province> GetProvincesByCountryId(string countryId);  /// <summary>  /// 获取指定的城市  /// </summary>  /// <param name="provinceId">ID</param>  /// <returns>城市列表</returns>  List<City> GetCitiesByProvinceId(string provinceId);  /// <summary>  /// 获取指定国家的城市  /// </summary>  /// <param name="countryId">国家ID</param>  /// <returns>城市列表</returns>  List<City> GetCitiesByCountryId(string countryId);  /// <summary>  /// 获取指定的国家  /// </summary>  /// <param name="provinceId">ID</param>  /// <returns>国家</returns>  Country GetCountryByProvinceId(string provinceId);  /// <summary>  /// 获取指定城市的  /// </summary>  /// <param name="cityId">城市ID</param>  /// <returns></returns>  Province GetProvinceByCityId(string cityId);  /// <summary>  /// 获取指定国家  /// </summary>  /// <param name="countryId">国家ID</param>  /// <returns>国家</returns>  Country GetCountryByCountryId(string countryId);  /// <summary>  /// 获取指定  /// </summary>  /// <param name="provinceId">ID</param>  /// <returns></returns>  Province GetProvinceByProvinceId(string provinceId);  /// <summary>  /// 获取指定城市  /// </summary>  /// <param name="cityId">城市ID</param>  /// <returns>城市</returns>  City GetCityByCityId(string cityId);  /// <summary>  /// 获取权限类型字典  /// </summary>  /// <returns>权限类型字典</returns>  Dictionary<int, string> GetPermissionsDict();  /// <summary>  /// 获取旅行状态字典  /// </summary>  /// <returns>旅行状态字典</returns>  Dictionary<int, string> GetTourStatusDict();  /// <summary>  /// 新增国家  /// </summary>  /// <param name="country">国家</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool InsertCountry(Country country, out string message);  /// <summary>  /// 更新国家  /// </summary>  /// <param name="country">国家</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool UpdateCountry(Country country, out string message);  /// <summary>  /// 删除国家  /// </summary>  /// <param name="country">国家</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool DeleteCountry(Country country, out string message);  /// <summary>  /// 新增  /// </summary>  /// <param name="province"></param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool InsertProvince(Province province, out string message);  /// <summary>  /// 更新  /// </summary>  /// <param name="province"></param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool UpdateProvince(Province province, out string message);  /// <summary>  /// 删除  /// </summary>  /// <param name="province"></param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool DeleteProvince(Province province, out string message);  /// <summary>  /// 新增城市  /// </summary>  /// <param name="city">城市</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool InsertCity(City city, out string message);  /// <summary>  /// 更新城市  /// </summary>  /// <param name="city">城市</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool UpdateCity(City city, out string message);  /// <summary>  /// 删除城市  /// </summary>  /// <param name="city">城市</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool DeleteCity(City city, out string message); |
|  |  |
| 类 | IPictureService |
| 说明 |  |
| 接口 | /// <summary>  /// 保存图片  /// </summary>  /// <param name="pictures">待保存的图片列表</param>  /// <param name="removedPictures">待删除的图片列表</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool SavePictures(List<Picture> pictures, List<Picture> removedPictures, out string message);  /// <summary>  /// 获取图片  /// </summary>  /// <param name="pictureId">图片ID</param>  /// <returns>图片</returns>  Picture GetPictureByPictureId(string pictureId); |
|  |  |
| 类 | ISightsService |
| 说明 | 提供景点服务接口 |
| 接口 | /// <summary>  /// 获取所有景点  /// </summary>  /// <returns>景点列表</returns>  List<Sights> GetSights();  /// <summary>  /// 获取指定国家的景点  /// </summary>  /// <param name="countryId">国家ID</param>  /// <returns>景点ID</returns>  List<Sights> GetSightsByCountryId(string countryId);  /// <summary>  /// 获取指定的景点  /// </summary>  /// <param name="provinceId">ID</param>  /// <returns>景点列表</returns>  List<Sights> GetSightsByProvinceId(string provinceId);  /// <summary>  /// 获取指定城市的景点  /// </summary>  /// <param name="cityId">城市ID</param>  /// <returns>景点列表</returns>  List<Sights> GetSightsByCityId(string cityId);  /// <summary>  /// 获取指定景点  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <returns>景点</returns>  Sights GetSightsBySightsId(string sightsId);  /// <summary>  /// 获取指定景点的平均评分  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <returns>分数</returns>  float GetAverageStarsBySightsId(string sightsId);  /// <summary>  /// 获取指定用户上传的指定景点的图片  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <param name="uploaderId">用户ID</param>  /// <returns>图片列表</returns>  List<Picture> GetPicturesBySightsIdAndUploaderId(string sightsId, string uploaderId);  /// <summary>  /// 获取用户游览过的景点  /// </summary>  /// <param name="userId">用户ID</param>  /// <returns>景点列表</returns>  List<Sights> GetVisitedSights(string userId);  /// <summary>  /// 获取用户游览过的指定国家的景点  /// </summary>  /// <param name="countryId">国家ID</param>  /// <param name="userId">用户ID</param>  /// <returns>景点列表</returns>  List<Sights> GetVisitedSightsByCountryId(string countryId, string userId);  /// <summary>  /// 获取用户游览过的指定的景点  /// </summary>  /// <param name="provinceId">ID</param>  /// <param name="userId">用户ID</param>  /// <returns>景点列表</returns>  List<Sights> GetVisitedSightsByProvinceId(string provinceId, string userId);  /// <summary>  /// 获取用户浏览过的指定城市的景点  /// </summary>  /// <param name="cityId">城市ID</param>  /// <param name="userId">用户ID</param>  /// <returns>景点列表</returns>  List<Sights> GetVisitedSightsByCityId(string cityId, string userId);  /// <summary>  /// 新增景点  /// </summary>  /// <param name="sights">景点</param>  /// <param name="pictures">相关图片</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool InsertSights(Sights sights, List<Picture> pictures, out string message);  /// <summary>  /// 更新景点  /// </summary>  /// <param name="sights">景点</param>  /// <param name="pictures">待更新图片</param>  /// <param name="removedPictures">待删除图片</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool UpdateSights(Sights sights, List<Picture> pictures, List<Picture> removedPictures, out string message);  /// <summary>  /// 删除景点  /// </summary>  /// <param name="sights">景点</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool DeleteSights(Sights sights, out string message); |
|  |  |
| 类 | ITourService |
| 说明 | 提供旅行服务接口 |
| 接口 | /// <summary>  /// 获取指定用户的所有旅行  /// </summary>  /// <param name="userId">用户ID</param>  /// <returns>旅行列表</returns>  List<Tour> GetToursByUserId(string userId);  /// <summary>  /// 获取指定旅行  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <returns>旅行</returns>  Tour GetTourByTourId(string tourId);  /// <summary>  /// 获取指定景点相关的旅行  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <returns>旅行ID</returns>  List<Tour> GetToursBySightsId(string sightsId);  /// <summary>  /// 获取包括指定日期的旅行  /// </summary>  /// <param name="date">日期</param>  /// <returns>旅行列表</returns>  List<Tour> GetToursByDate(DateTime date);  /// <summary>  /// 获取指定景点相关，并包括指定日期的旅行  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <param name="date">日期</param>  /// <returns>旅行列表</returns>  List<Tour> GetToursBySightsIdAndDate(string sightsId, DateTime date);  /// <summary>  /// 删除旅行  /// </summary>  /// <param name="tour">旅行</param>  /// <param name="deletePictures">是否删除相关图片</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool DeleteTour(Tour tour, bool deletePictures, out string message);  /// <summary>  /// 保存旅行  /// </summary>  /// <param name="tour">旅行</param>  /// <param name="subTours">待保存的子旅行列表</param>  /// <param name="removedSubTours">待删除的子旅行列表</param>  /// <param name="msg">操作信息</param>  /// <returns>操作是否成功</returns>  bool SaveTour(Tour tour, List<SubTour> subTours, List<SubTour> removedSubTours, out string msg);  /// <summary>  /// 获取指定旅行的子旅行  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <returns>子旅行列表</returns>  List<SubTour> GetSubToursByTourId(string tourId);  /// <summary>  /// 获取指定子旅行  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <param name="subTourId">子旅行ID</param>  /// <returns>子旅行</returns>  SubTour GetSubTourByTourIdAndSubTourId(string tourId, string subTourId);  /// <summary>  /// 获取指定用户的子旅行  /// </summary>  /// <param name="userId">用户ID</param>  /// <returns>子旅行列表</returns>  List<SubTour> GetSubToursByUserId(string userId);  /// <summary>  /// 获取指定景点相关的所有子旅行  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <returns>子旅行列表</returns>  List<SubTour> GetSubToursBySightsId(string sightsId);  /// <summary>  /// 获取指定旅行的图片  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <returns>图片列表</returns>  List<Picture> GetPicturesByTourId(string tourId);  /// <summary>  /// 获取指定子旅行的图片  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <param name="subTourId">子旅行ID</param>  /// <returns>图片列表</returns>  List<Picture> GetPicturesByTourIdAndSubTourId(string tourId, string subTourId); |
|  |  |
| 类 | IUserService |
| 说明 | 提供用户服务接口 |
| 接口 | /// <summary>  /// 获取所有用户  /// </summary>  /// <returns>用户列表</returns>  List<User> GetUsers();  /// <summary>  /// 用户登录  /// </summary>  /// <param name="loginName">登录名</param>  /// <param name="password">口令</param>  /// <returns>若登录成功则返回用户信息，否则返回null</returns>  User Login(string loginName, string password);  /// <summary>  /// 获取指定用户  /// </summary>  /// <param name="loginName">登录名</param>  /// <returns>用户</returns>  User GetUserByLoginName(string loginName);  /// <summary>  /// 获取指定用户  /// </summary>  /// <param name="email">Email</param>  /// <returns>用户</returns>  User GetUserByEmail(string email);  /// <summary>  /// 获取指定用户  /// </summary>  /// <param name="userId">用户ID</param>  /// <returns>用户</returns>  User GetUserByUserId(string userId);  /// <summary>  /// 获取用户权限列表  /// </summary>  /// <param name="userId">用户ID</param>  /// <returns>权限列表</returns>  List<Permission> GetPermissionsByUserId(string userId);  /// <summary>  /// 新增用户  /// </summary>  /// <param name="user">用户</param>  /// <param name="password">口令</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool InsertUser(User user, string password, out string message);  /// <summary>  /// 更新用户  /// </summary>  /// <param name="user">用户</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool UpdateUser(User user, out string message);  /// <summary>  /// 获取用户收藏景点  /// </summary>  /// <param name="userId">用户ID</param>  /// <returns>收藏景点列表</returns>  List<Favorite> GetFavoritesByUserId(string userId);  /// <summary>  /// 获取指定的收藏景点  /// </summary>  /// <param name="userId">用户ID</param>  /// <param name="sightsId">景点ID</param>  /// <returns>收藏景点</returns>  Favorite GetFavoriteByUserIdAndSightsId(string userId, string sightsId);  /// <summary>  /// 更新用户权限  /// </summary>  /// <param name="userId">用户ID</param>  /// <param name="permissions">权限列表</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool UpdatePermissionsByUserId(string userId, List<Permission> permissions, out string message);  /// <summary>  /// 保存用户收藏景点  /// </summary>  /// <param name="favorite">收藏</param>  /// <param name="message">操作信息</param>  /// <returns>操作是否成功</returns>  bool SaveFavorite(Favorite favorite, out string message); |

（3）日志接口

|  |  |
| --- | --- |
| 类 | ILogHelper |
| 说明 | 日志接口，提供了日志的常用操作 |
| 接口 | void Debug(object obj, Exception ex);  void Debug(object obj);  void Error(object obj, Exception ex);  void Error(object obj);  void Fatal(object obj, Exception ex);  void Fatal(object obj);  void Info(object obj, Exception ex);  void Info(object obj);  void Warn(object obj, Exception ex);  void Warn(object obj);  string GetLogTimeString(); |

（4）UI层User Control组件接口

|  |  |
| --- | --- |
| 类 | STGReadonlyPictures |
| 说明 | 只读图片列表组件 |
| 接口 | /// <summary>  /// 设置或获取控件中的所有图片  /// </summary>  List<Picture> Pictures { get; private set; }  /// <summary>  /// 设置控件，显示指定用户上传的关于指定景点所有的图片  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <param name="uploaderId">用户ID</param>  /// <param name="allowComment">是否允许添加图片评论</param>  void SetPicturesForSights(string sightsId, string uploaderId, bool allowComment);  /// <summary>  /// 设置控件，显示指定用户上传的关于指定旅行的所有图片  /// </summary>  /// <param name="tourId">旅行ID</param>  /// <param name="allowComment">是否允许添加图片评论</param>  void SetPicturesForTour(string tourId, bool allowComment);  /// <summary>  /// 重置控件，清除所有图片  /// </summary>  void ResetList(); |
|  |  |
| 类 | STGPictures |
| 说明 | 图片列表组件 |
| 接口 | /// <summary>  /// 设置或获取控件中的所有图片  /// </summary>  List<Picture> Pictures { get; private set; }  /// <summary>  /// 设置或获取控件中所有已删除的图片  /// </summary>  List<Picture> RemovedPictures { get; private set; }  /// <summary>  /// 设置控件，显示指定用户上传的关于指定景点所有的图片  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <param name="uploaderId">用户ID</param>  void SetPicturesForSights(string sightsId, string uploaderId);  /// <summary>  /// 设置控件，显示指定用户上传的关于指定景点和指定旅行的所有图片  /// </summary>  /// <param name="sightsId">景点ID</param>  /// <param name="tourId">旅行ID</param>  /// <param name="subTourId">子旅行ID</param>  /// <param name="uploaderId">用户ID</param>  void SetPicturesForTour(string sightsId, string tourId, string subTourId, string uploaderId);  /// <summary>  /// 重置控件，清除所有图片  /// </summary>  void ResetList(); |

* 1. 配置文件

配置文件通过.NET提供的CongfigurationManager类进行访问，按照作用分为以下几部分：数据库Provider配置，数据库连接字符串配置，数据连接层接口配置，服务层接口配置，日志接口配置以及其它相关配置。

数据库Provider配置，是对.NET默认提供的几种连接数据库的Provider的补充，主要是使用其中的SQLite配置；数据库连接字符串配置记录了连接数据库的字符串；数据连接层接口配置，记录了程序在运行中使用的数据连接层，通常针对不同的数据库开发不同的数据连接层；服务层接口和日志接口配置同上所属，记录了程序运行过程中使用的服务层和日志。

1. **其它**
2. 1. AOP

之前引入Spring.NET作为实现AOP的框架,尽管实现了全部的需求，但是考虑到C#语言的一些特点，Spring.NET框架没有体现出AOP“织入”的理念，并不是最佳的实现方案。因此本次尝试以PostSharp作为框架，实现AOP，以减少对源代码的修改。不同于传统的代理模式，PostSharp是使用了.NET语言的特性(Attribute)来实现AOP的。

* 1. 日志

日志模块提供了ILog接口，上一阶段使用Log4Net动态连接库，对其进行封装并实现了ILog接口；现增加了通过NLog动态连接库项目实现该功能。可以通过配置文件对Log4Net和NLog进行选择。

* 1. 数据连接层

在之前的基础上新增了OracleDAL，实现IDAL中的接口，使程序支持对Oracle数据库的访问。可以通过配置文件对SQLite和Oracle进行选择。