

文章编号: 2096-1472(2016)-12-40-03

# 基于Entity Framework和easyUI的毕业论文管理系统

喻 健, 朱三元

(湖北工程学院, 湖北 孝感 432000)

**摘要:** 毕业论文是高等教育不可或缺的环节, 为了解决传统论文管理方式存在的材料冗余、数据更新缓慢、师生交流不畅等问题, 本文应用Entity Framework框架和easyUI工具集构建了一个全新的论文管理系统, 实现了毕业论文整个流程的信息化和模块化。文章重点阐述了以EF框架为基础创建数据模型、构建后台逻辑的关键技术, 以及使用easyUI编写用户视图的方法, 能对基于EF技术框架和使用easyUI的WEB系统开发者提供一定参考。

**关键词:** Entity Framework; easyUI; MVC; Web应用

**中图分类号:** TP311 **文献标识码:** A

## The Graduation Thesis Management System Based on Entity Framework and EasyUI

YU Jian, ZHU Sanyuan

(Hubei Engineering University, Xiaogan 432000, China)

**Abstract:** The graduation thesis is an indispensable part of higher education. In order to solve the problems in traditional paper management methods, such as redundant materials, slow data updating, and the miscommunication between teachers and students, the study constructs a new graduation thesis management system with the application of Entity Framework and EasyUI, achieving the informatization and modularization of the whole graduation thesis management process. This paper focuses on the key techniques of constructing data models and background logic based on Entity Framework, and the methods to write user view based on EasyUI. This study offers some reference to the WEB developers applying the EF technical framework and EasyUI.

**Keywords:** entity framework; EasyUI; MVC; web application

### 1 引言(Introduction)

随着我国高等教育的普及, 高校在校人数的激增, 学生毕业论文的工作量也与日俱增。传统的毕业论文流程以人工操作为主、信息化程度低、学生选题和师生间交流探讨都是线下完成, 诸如论文题目的审核、往年题目的查询、相似题目的筛选、选题学生数量的控制等问题都难以解决。而随着论文开题、阶段总结、中期答辩、预答辩、最终答辩等环节的推进, 论文材料和修改版本增多, 更会造成数据冗余、查询困难、资料容易丢失等问题, 导致论文工作效率低下。

本文应用微软的ADO.NET Entity Framework框架和JqueryEasyUI用户界面工具集, 借助Visual Studio 2013开发的论文管理系统具有良好的扩展性和维护性。在满足性能要求的前提下, 给用户提供了便捷的操作界面, 一定程度上解决了传统论文处理方式的问题<sup>[1]</sup>。

### 2 平台简介(Platform profile)

#### 2.1 Entity Framework

Entity Framework是微软官方提供的以ADO.NET为基础的ORM(O/R Mapping)工具, 早期被称为ObjectSpace, ORM让开发人员能专注于业务逻辑层代码, 节省了数据库访问的代码时间。开发人员用Entity Framework把数据库对象转换成应用程序对象(entity), 数据字段转换为属性(property), 关系则转换为关联属性(association), 让数据库

操作变得和操作Object对象一样方便。

Entity Framework支持三种开发模式Database First、Model First和Code First, 各模式的开发流程完全不一样, 但是三种开发模式各有优缺点, 适用于不同的开发场景。Database First用于从数据库生成实体类Class, Model First用于通过Visual Studio的可视化设计器设计数据模型然后生成实体类, Code First则用于从实体类生成数据库结构<sup>[2]</sup>。三种开发模式的流程如图1所示。

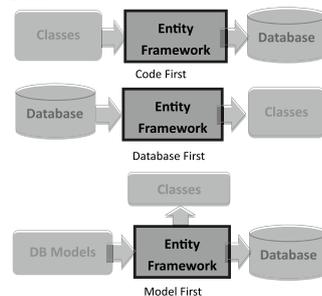


图1 Entity framework三种典型开发模式

Fig.1 Three typical development modes of entity framework

#### 2.2 JQuery EasyUI

jQuery是一个高效简洁的JavaScript框架, 是一个优秀的JavaScript代码库。jQuery设计的宗旨是“Write Less, Do More”, 即倡导写更少的代码, 做更多的事情。它封装了

JavaScript常用的功能代码，提供一种简便的JavaScript设计模式，优化了HTML文档操作、事件处理、动画设计和Ajax交互等。

JqueryEasyUI是基于jQuery的一组用户界面工具集，它比jQuery提供更多的功能和更好的用户体验，同时支持HTML5的完整框架。开发者只需要通过编写一些简单HTML标记，就可以定义用户界面，为网页的前端开发提高了效率<sup>[3]</sup>。

### 3 系统设计(System design)

#### 3.1 功能模块设计

在本论文介绍的管理系统中涉及的用户角色有以下几类，它们对系统的需求大有不同。因此需要和各种角色的用户充分沟通交流，发掘用户对目标软件系统在功能、行为、性能、设计约束等方面的期望。需求分析是系统功能模块设计的基础。

##### (1)学生

学生角色是本系统的主体，涉及个人信息的维护、论文的选题、开题报告的提交、论文答辩等各环节与系统的交互，以及各个阶段资料的提交。

##### (2)教师

在本毕业论文系统中，教师角色又被分为三种：普通教师、答辩组长和推优专家。

普通教师的角色任务包括维护教师个人信息、论文出题、教师和被指导学生的双向选择、维护指导学生的信息、发布任务书、检查学生论文完成的环节、接收学生各阶段提交的材料、参加答辩、评定论文成绩等。

答辩组长角色除了能完成普通教师的角色任务外，还能安排答辩的时间、选定答辩小组的成员教师、对本小组答辩学生推优等。

推优专家角色除了具备上面两种教师角色的功能外，最主要的功能是对各小组推荐的优秀论文进行评阅并量化打分，但是不能对自己所指导小组的论文评分。

##### (3)院系管理员

院系管理员可以发布并维护通知公告、维护教师和学生用户的信息、设置毕业论文相关的参数、对教师所出论文题目的审核和修改、维护学生选题的结果、维护论文设计各个阶段的文档资料、设置教师的角色权限、维护论文评分方案、统计论文数据等。

##### (4)系统管理员

系统管理员是本系统里权限最高的用户角色，可以完成系统的初始化和参数设置、公告管理、用户管理、角色管理、资料管理、数据统计等。

根据以上需求分析提出的角色和功能划分，系统被分为学生模块、教师模块、院系管理员模块和系统管理员模块四大功能模块及其子模块。系统在用户登录的时候，根据用户的ID判断用户的角色并跳转到相应的操作界面。系统的功能模块结构如图2所示。

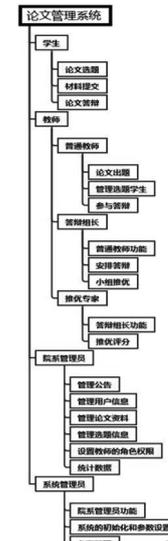


图2 系统功能模块图

Fig.2 System function module

#### 3.2 业务流程设计

本系统采用B/S模式，并结合先进且成熟的Entity Framework技术，采用统一的用户认证方式，对不同用户的操作和数据安全进行有效控制，图3反映了毕业论文管理系统的基本工作流程。

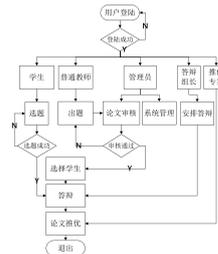


图3 系统流程图

Fig.3 System flow chart

### 4 系统关键技术(System key technology)

#### 4.1 Code First 开发模式

##### (1)创建数据实体模型

本文的系统是基于Entity Framework的Code First开发模式，并使用了MVC架构，创建系统的数据实体模型是系统设计的首要任务，这里我们以Student实体为例说明创建模型的基本方法<sup>[4]</sup>。

在MVC的Models文件夹里创建类Student.cs，代码结构如图4所示。

```

1 namespace PaperManage.Models.DataEntity
2 {
3     [Table("Students")]
4     public class Student
5     {
6         [Key]
7         [Display(Name = "学号")]
8         public string StudentNo { get; set; }
9         [Display(Name = "姓名")]
10        public string StudentName { get; set; }
11        [Display(Name = "性别")]
12        public bool Sex { get; set; }
13        [Display(Name = "班级")]
14        [ForeignKey("Class")]
15        public int ClassId { get; set; }
16        public class Class { get; set; }
17        [ForeignKey("Paper")]
18        public int? PaperId { get; set; }
19        public Paper Paper { get; set; }
20    }
21 }

```

图4 数据模型定义

Fig.4 Data model definition

在完成数据模型的创建或者修改后，需在程序包管理控制器中先运行Enable-Migrations命令，再运行update-database命令，把模型的改变更新到数据库，实现Code First对数据库的操作。

(2)创建上下文

创建完数据实体后需要定义数据库上下文，上下文的作用是协调数据模型和Entity Framework之间的工作，它表示一个数据库会话。我们自定义的上下文类必须直接或者间接的继承自System.Data.Entity.DbContext类，并在此类中为每个数据实体类定义一个公开的DbSet<TEntity>，如上面的Student类在上下文类中可定义为public DbSet<Student> Students{ get;set; }。

(3)数据的读写

实体数据的读写主要由对应的控制器完成，以Student实体为例，在控制器文件夹Controllers里创建控制器StudentController，控制器类必须继承自System.Web.Mvc.Controller类。在执行数据操作前，还要先实例化上面定义的数据库上下文类<sup>[5]</sup>。

系统在前台使用easyUI插件来构建视图界面完成用户与系统的交互，把用户输入的表单数据序列化以后，以json格式通过AJAX传回给控制器处理，一段典型的JS交互函数代码如图5所示。

```

1 function Do() {
2   if ($("#ff").form("validate")) {
3     $.ajax({
4       url: $("#").content("~/Student/Save"),
5       data: $("#ff").serialize(),
6       type: "post",
7       dataType: "json",
8       success: function (data) {
9         if (data.result == true) {
10          //保存成功
11          $("#msgidit").window("close");
12          $("#list").datagrid("reload");
13          //刷新列表
14          } else { //保存失败
15          }
16          error:
17          function (XMLHttpRequest, textStatus, errorThrown) {
18            //异常处理
19          }
20        });
21      } else { //表单数据无效
22      }
23    }

```

图5 JS交互函数

Fig.5 JS interactive function

控制器方法接收到前台的json数据后，利用LINQ和EF框架对数据库进行相应操作，然后返回JsonResult类型的数据结果，控制器方法如图6所示。

```

1 public JsonResult Save(Student student)
2 {
3   if (student.StudentID == 0)
4   {
5     db.Students.Add(student);
6     string UserId =
7     StudentOperation.CreateUser(student.StudentID, student.StudentName);
8     if (UserId.Equals(string.Empty))
9     {
10      return JsonResult("创建新用户失败。");
11    }
12    else
13    {
14      student.UserId = UserId;
15    }
16    else
17    {
18      Student s_new =
19      db.Students.FirstOrDefault(o => o.StudentID == student.StudentID);
20      DbEntityEntry<Student> entry = db.Entry<Student>(s_new);
21      entry.State = EntityState.Unchanged;
22      //添加新的属性
23      if (s_new.StudentName != student.StudentName)
24      {
25        s_new.StudentName = student.StudentName;
26        entry.Property(e => e.StudentName).IsModified = true;
27      }
28      //修改其他属性
29      {
30      }
31      db.SaveChanges();
32      return JsonResult("result = true");
33    }
34  }

```

图6 控制器方法

Fig.6 Controller function

4.2 easyUI的调用

(1)直接在HTML的样式表中调用

在编写视图页面的HTML代码时，把HTML组件的class属性设置为easyUI内置的属性即可，如<div id="menu" class="easyui-menu"></div>可把当前DIV组件设置为easyUI的menu控件。

(2)使用JavaScript动态的调用

这种方式能更灵活的改变easyUI控件的样式，如上文已经定义了一个HTML组件<input id="test" style="width:100px"/>，下文可以用JS代码\$("#test").combobox({url: ...,valueField:'id',textField:'text'});来把上面的input组件设置为easyUI的combobox控件。

5 结论(Conclusion)

Entity Framework提供了自动化CRUD操作，使开发者从数据库API和SQL中解放出来，从而专注于业务逻辑层代码。easyUI基于jQuery，让开发者通过一些简单HTML标记就可以定义美观实用的用户界面。本文介绍的论文管理系统基于这两种成熟的技术，让业务逻辑和界面代码完全分开，提高了程序的开发效率和可维护性，让论文管理流程实现了信息化，优化了论文工作流程，对其他WEB应用的开发也有参考价值。

参考文献(References)

- [1] XieZhongming,et al.Design and Application of Lighting Energy Consumption Monitoring Platform Based on MVC and Entity Framework[C].ICIA 2014:IEEE International Conference on Information and Automation,2014:473-477.
- [2] LiuZhongwei,WuShu-Guang.Design of .Net Courseware on Demand System[C].ICMTMA 2016:2016 8th International Conference on Measuring Technology and Mechatronics Automation,2016:206-209.
- [3] LuoTing.Research and Application of Easy UI Asynchronous Tree[C].ICMST 2013:Advanced Materials Research,2013: 1116-1120.
- [4] 盖洛德,温兹,拉丝托格.ASP.NET 4.5高级编程第8版[M].北京:清华大学出版社,2014:367-370.
- [5] TomDykstra.Getting Started with Entity Framework 6 Code First Using MVC 5[J]/OL.

作者简介:

喻 健(1989-), 男, 硕士, 助理实验师.研究领域: 计算机应用.

朱三元(1970-), 男, 硕士, 副教授.研究领域: 计算机应用.