### 把老师的PPT看一遍(熟悉概念)

### 自己写一个增删改查的程序(熟悉代码编写)

### 什么是MVC设计模式?

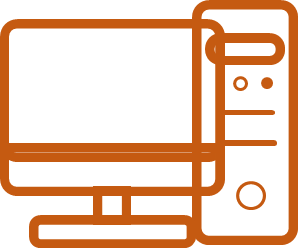
MVC即Model-View-Controller（模型-视图-控制器）是一种软件设计模式，最早出现在Smalltalk语言中，后被Sun公司推荐为Java EE平台的设计模式。

可以用下图来表示MVC模式中三者之间的关系：

Model

View

Controller



传递数据参数

Response

返回查询结果

返回数据模型

Request

* jsp
* html
* xml
* json
* ect
* 实体
* 业务逻辑
* 工具类
* 数据访问
* 用户交互
* 与模型协作
* 选择一个视图展示数据
* xml

**Figure 1 MVC设计模式**

弄清MVC架构的原理,整个数据流向,

如何实现添加数据,添加数据过程的数据流通

### 使用Ajax提交请求

$(function(){

$('#send').click(function(){

$.ajax({

type: "GET",

url: "test.do ",

data: {username:$("#username").val(),

content:$("#content").val()},

dataType: "json",

success: function(data){

//do something

}

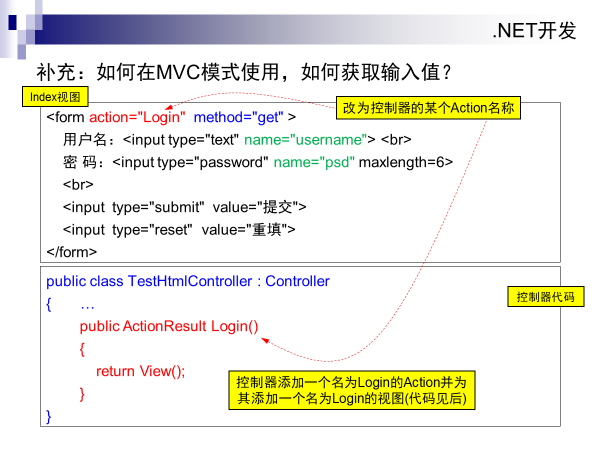
});

});

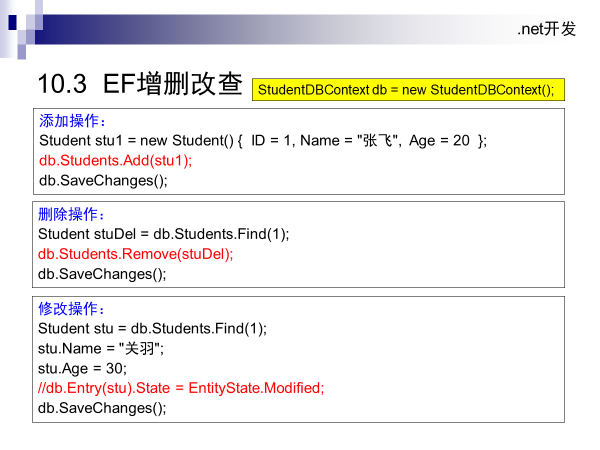
});

### 表单书写+提交+数据获取



### 增删改查



查找操作:

1.查找所有数据

db.Students.ToList();

2.条件查询

db.Students.Where<Student>(u => u.Age < 22 && u.Name.Contains("张"));

db.Students.Where<Student>(u=>u.ID == 1)

### 传递数据的几种方式



#### 使用ViewBag传递数据

public ViewResult Welcome()

{

ViewBag.Message = "Hello";

ViewBag.Date = DateTime.Now;

return View();

}

视图:

@{

ViewBag.Title = "Welcome";

}

Today is: @ViewBag.Date.DayOfWeek

The message is: @ViewBag.Message

#### 使用ViewData传递数据

ViewData使用Key/Value形式传递数据，数据需要在视图中进行对象转换。

public ViewResult Welcome()

{

ViewData["Message"] = "Hello";

ViewData["Date"] = DateTime.Now;

return View();

}

视图:

The day is: @(((DateTime)ViewData["Date"]).DayOfWeek)

The message is: @ViewData["Message"]

#### 使用视图模型传递数据

把一个对象作为View方法的参数传递给视图。视图中使用Model关键字来访问这个对象。

public ViewResult Welcome()

{

DateTime date = DateTime.Now;

return View( date );

}

视图:

@model DateTime//先声明Model的类型

The day is: @Model.DayOfWeek

RedirectResult主要用于执行指向其它地址的重定向。

public ActionResult RedirectToUrl()

{

return Redirect("/Home/Index");

}

### 灵活使用增删改查(分析题40分)

具体代码(可能是写一个学生系统)

注册(添加操作)

修改学号操作(修改操作)

查询成绩(查询操作)

转专业(修改操作)

namespace r3.Models

{

public class User

{

[Key]

[Required(ErrorMessage = "x")]

[Display(Name = "学号")]

public string username { get; set; }

[Display(Name = "密码")]

[Required(ErrorMessage = "x")]

[DataType(DataType.Password)]

public string password { get; set; }

[Display(Name = "姓名")]

public string name { get; set; }

}

public class userDBContext : DbContext

{

public DbSet<User> Users { get; set; }

}

}

**public UserDbContext db=new UserDbContext();**