ERROR

1. Syntax Error
2. Logical Error
3. Exception (We can handle them)

We can handle exceptions in MVC in different ways

1. At ActionMethod Level

public ActionResult Create([Bind(Include = "userKey,FirstName,LastName,Age,Password,ReTypePassword,DateOfjoin")] User user)

{

try {

db.Users.Add(user);

db.SaveChanges();

return RedirectToAction("Index");

}

catch (Exception ex)

{

return View("Error", new HandleErrorInfo(ex, "users", "Create"));

}

return View(user);

}

public ActionResult Contact()

{

try

{

string msg = null;

ViewBag.Message = msg.Length;

return View();

}

catch(Exception ex)

{

return View("Error", ex);

}

}

ERROR VIEW

@model System.Web.Mvc.HandleErrorInfo

<h1 class="text-danger">Error.@Model.Exception.Message</h1>

<h2 class="text-danger">An error occurred while processing your request.</h2>

@Html.ActionLink("Back", @Model.ActionName, @Model.ControllerName)

1. Handling Exceptions at Controller level:

**: We use OnException()**

protected override void OnException(ExceptionContext filterContext)

{

Exception exception = filterContext.Exception;

//Logging the Exception

filterContext.ExceptionHandled = true;

var Result = this.View("Error", new HandleErrorInfo(exception,

filterContext.RouteData.Values["controller"].ToString(),

filterContext.RouteData.Values["action"].ToString()));

filterContext.Result = Result;

}

--------------------------------------------------------

FOR ALL ACTION METHODS

public ActionResult Create([Bind(Include = "userKey,FirstName,LastName,Age,Password,ReTypePassword,DateOfjoin")] User user)

{

db.Users.Add(user);

db.SaveChanges();

return RedirectToAction("Index");

}

1. Handle Exception at Application level

We use Application\_Error() event in global.asax file

protected void Application\_Error(object sender, EventArgs e)

{

Exception exception = Server.GetLastError();

System.Diagnostics.Debug.WriteLine(exception);

}

To practice this, we added an action Method in other controller

public string w()

{

throw new Exception("ex1111");

return "A";

}

We have to do some more settings for Exception Handlig

1st Thing:

In Web.config, add customErrors

<system.web>

<customErrors mode="On" defaultRedirect="\users\error"></customErrors>

2nd thing: Add FilterConfig.cs file in App\_Start() folder

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace WebApplication61

{

public class FilterConfig

{

public static void RegisterGlobalFilters(GlobalFilterCollection filters)

{

filters.Add(new HandleErrorAttribute());

}

}

}

3rd Thing: Register it in Application\_Start() Event

protected void Application\_Start()

{

AreaRegistration.RegisterAllAreas();

**FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);**

RouteConfig.RegisterRoutes(RouteTable.Routes);

}