**特性**

public abstract class BaseValidationAttribute: Attribute

{

public BaseValidationAttribute(string errorMessage) {

ErrorMessage = errorMessage;

}

public string ErrorMessage { get; set; }

public abstract bool IsValid();

}

public class RoleAuthAttribute : BaseValidationAttribute

{

private List<string> \_Role = new List<string>();

public RoleAuthAttribute(object[] role) : base("授权错误") {

foreach (var item in role) {

\_Role.Add(item.ToString());

}

}

public RoleAuthAttribute(object[] role, string error) : base(error) {

foreach (var item in role)

{

\_Role.Add(item.ToString());

}

}

public override bool IsValid()

{

IPrincipal principal = GenerateAndGetPrincipal.GetPrincipal();

if (principal == null)

{

return false;

}

foreach (var item in \_Role) {

if (principal.IsInRole(item) == true)

return true;

}

return false;

}

}

**验证**

public static bool Validate(Type type,out string errorMessage)

{

errorMessage = null;

foreach (var item in type.CustomAttributes)

{

//获取构造函数信息

ConstructorInfo service = item.Constructor;

if (service.DeclaringType.IsSubclassOf(typeof(BaseValidationAttribute)))

{

List<object> paramList = new List<object>();

//获取构造函数实参信息

foreach(var parameterInfoItem in item.ConstructorArguments) {

List<object> list = new List<object>();

foreach (var parameterItem in (ReadOnlyCollection<CustomAttributeTypedArgument>) parameterInfoItem.Value) {

//添加实参值

list.Add(parameterItem.Value);

}

paramList.Add(list.ToArray());

}

//调用构造方法生成特性实例

BaseValidationAttribute validationAttribute = service.Invoke(paramList.ToArray()) as BaseValidationAttribute;

//调用特性的方法进行验证

bool isSuccess = validationAttribute.IsValid();

if (isSuccess == false)

{

errorMessage = validationAttribute.ErrorMessage;

return false;

}

}

}

return true;

}