public class PointModelBinder : IModelBinder

{

public object BindModel(ControllerContext controllerContext,

ModelBindingContext bindingContext)

{

if (!String.IsNullOrEmpty(bindingContext.ModelName) &&

!bindingContext.ValueProvider.ContainsPrefix(bindingContext.ModelName))

//模型名称不为空且提供的值不包含以模型名称为前缀的值

{

if (!bindingContext.FallbackToEmptyPrefix) //文本要求绑定器需要使用模型名称为前缀

return null;

bindingContext = new ModelBindingContext

{

ModelMetadata = bindingContext.ModelMetadata,

ModelState = bindingContext.ModelState,

PropertyFilter = bindingContext.PropertyFilter,

ValueProvider = bindingContext.ValueProvider

};

}

bindingContext.ModelMetadata.Model = new Point();

//返回模型实例

return new Point(

Get<int>(controllerContext, bindingContext, "X"),

Get<int>(controllerContext, bindingContext, "Y")

);

}

private TModel Get<TModel>(ControllerContext controllerContext,

ModelBindingContext bindingContext,

string name)

{

string fullName = name;

if (!String.IsNullOrWhiteSpace(bindingContext.ModelName)) //如果模型名称不为空

fullName = bindingContext.ModelName + "." + name;

//根据模型名称用提供器查找值对应的值

ValueProviderResult valueProviderResult = bindingContext.ValueProvider.GetValue(fullName);

ModelState modelState = new ModelState { Value = valueProviderResult };

//添加模型状态到模型状态列表中

bindingContext.ModelState.Add(fullName, modelState);

//获取属性的元数据

ModelMetadata metadata = bindingContext.PropertyMetadata[name];

//取出的字符串形式的结果

string attemptedValue = valueProviderResult.AttemptedValue;

if (metadata.ConvertEmptyStringToNull && String.IsNullOrWhiteSpace(attemptedValue)) //元数据指定将空字符串转为NULL且attemptedValue是空的

attemptedValue = null;

TModel model;

bool invalidValue = false;

try

{

//将结果转为指定类型

model = (TModel)valueProviderResult.ConvertTo(typeof(TModel));

metadata.Model = model;

}

catch (Exception)

{

model = default(TModel);

metadata.Model = attemptedValue;

invalidValue = true;

}

//返回模型的验证程序列表

IEnumerable<ModelValidator> validators = ModelValidatorProviders.Providers.GetValidators(metadata, controllerContext);

//枚举验证器

foreach (var validator in validators)

//运行验证，并枚举验证结果

foreach (var validatorResult in validator.Validate(bindingContext.Model))

//将验证结果的错误消息添加到当前的模型状态中

modelState.Errors.Add(validatorResult.Message);

if (invalidValue && modelState.Errors.Count == 0)

modelState.Errors.Add(

String.Format(

"The value '{0}' is not a valid value for {1}.",

attemptedValue,

metadata.GetDisplayName()

)

);

return model;

}

}