第一种方法是继承DefaultModelBinder类，然后重写它的CreateModel方法，并且在Application\_Start方法中设置ModelBinders.Binders.DefaultBinder为这个模型绑定器（表明现在的默认模型绑定器用的是我们自己定义的）：

public class DIModelBinder : DefaultModelBinder {

protected override object CreateModel(ControllerContext controllerContext, ModelBindingContext bindingContext, Type modelType) {

return DependencyResolver.Current.GetService(modelType) ?? base.CreateModel(controllerContext, bindingContext, modelType);

}

}

然后在global.asax的Application\_Start方法中设置默认模型绑定器为DIModelBinder：

protected void Application\_Start() {

// ...

ModelBinders.Binders.DefaultBinder = new DIModelBinder();

// ...

}

第二种方法继承IModelBinder接口，然后实现它的接口方法：

public class PersonModelBinder : IModelBinder

{

public object BindModel(ControllerContext controllerContext, ModelBindingContext bindingContext) {

// 检查是否有现成的model对象，如果没有创建一个（如果使用手动绑定则bindingContext.Model就不会为null）

Person model = (Person) bindingContext.Model ?? (Person)DependencyResolver.Current.GetService(typeof(Person));

// find out if the value provider has the required prefix

bool hasPrefix = bindingContext.ValueProvider.ContainsPrefix(bindingContext.ModelName); // bindingContext.ModelName返回当前模型的名称

string searchPrefix = (hasPrefix) ? bindingContext.ModelName + "." : "";

// 填充model对象的字段

model.PersonId = int.Parse(GetValue(bindingContext, searchPrefix, "PersonId"));

model.FirstName = GetValue(bindingContext, searchPrefix, "FirstName");

model.LastName = GetValue(bindingContext, searchPrefix, "LastName");

model.BirthDate = DateTime.Parse(GetValue(bindingContext, searchPrefix, "BirthDate"));

model.IsApproved = GetCheckedValue(bindingContext, searchPrefix, "IsApproved");

model.Role = (Role)Enum.Parse(typeof(Role), GetValue(bindingContext, searchPrefix, "Role"));

return model;

}

private string GetValue(ModelBindingContext context, string prefix, string key) {

ValueProviderResult vpr = context.ValueProvider.GetValue(prefix + key);

return vpr == null ? null : vpr.AttemptedValue;

}

private bool GetCheckedValue(ModelBindingContext context, string prefix, string key) {

bool result = false;

ValueProviderResult vpr = context.ValueProvider.GetValue(prefix + key);

if (vpr != null) {

result = (bool)vpr.ConvertTo(typeof(bool));

}

return result;

}

}

然后同样地我们需要注册模型绑定器，可以全局注册，方法是在Application\_Start方法中添加下面代码：

protected void Application\_Start() {

// ...

ModelBinders.Binders.Add(typeof(Person), new PersonModelBinder());

// ...

}

下面进行模型绑定器与类型的绑定：

[ModelBinder(typeof(PersonModelBinder))]public class Person {

[HiddenInput(DisplayValue=false)]

public int PersonId { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

}