

ModelState Vs ModelBinder

لو اشتغلت على ASP.NET MVC قبل كده، أكيد سمعت عن ModelBinder و ModelState. هما اثنين من الأدوات اللي بتساعدك تربط البيانات اللي جاية من اليوزر مع الكود بتاعك، وتتأكد إنها صحيحة. بس الناس كتير بتتلخبط بينهم. فهنشوف ازاي

إيه هو ModelBinder؟

ال ModelBinder هو اللي مسؤؤل عن "ربط" البيانات اللي جاية من الطلب (Request) زي Form Data، Query String، أو JSON Body، مع البارامترات أو ال Models في الكونترولر بتاعك.

يعني لما اليوزر بيعت فورم، ال ModelBinder بياخد البيانات دي ويحولها لكائن من نوع ال Model اللي انت عايزه.

ده بيحصل تلقائيًا في ASP.NET MVC، بس ممكن تخصصه لو عايز.

مثال بسيط:

تخيل عندك Model زي ده:

```
public class UserModel {  
    public string Name { get; set; }  
    public int Age { get; set; }  
}
```

وفي الكونترولر:

```
public ActionResult Register(UserModel user) {  
    // هيكون مربوط بالبيانات اللي جات من الفورم user هنا ال  
}
```

ال ModelBinder بيحول ال "Name=Ahmed&Age=25" إلى كائن UserModel جاهز.

أنواع ModelBinders:

- DefaultModelBinder: الافتراضي، يشتغل مع معظم الحالات.
- Custom ModelBinder: لو عايز تكتب واحد خاص بيبك، زي لو البيانات جاية بشكل غريب.

إيه هو ModelState؟

ال ModelState هو قاموس (Dictionary) بيحتوي على حالة ال Model بعد الربط. هو اللي بيتحقق من "صحة" البيانات بناءً على ال Validation Attributes زي [Required] أو [Range].

يعني بعد ما ال ModelBinder يربط البيانات، ال ModelState بيتشيك لو في أخطاء، زي لو حقل مطلوب مش موجود، أو قيمة غلط.

مثال:

في نفس الموديل:

```
public class UserModel {  
    [Required(ErrorMessage = "الاسم مطلوب")]  
    public string Name { get; set; }  
    [Range(18, 100, ErrorMessage = "العمر بين 18 و 100")]  
    public int Age { get; set; }  
}
```

في الكونترولر:

```
if (!ModelState.IsValid) {  
    // في أخطاء، رجع اليوزر للفورم مع الرسائل  
    return View(user);  
}
```

ال ModelState هنا يحتوي على ModelStateDictionary اللي فيه Keys لكل حقل، و Errors لو في مشاكل.

1. الفرق بين ModelState و ModelBinder

الاثنين دول مش نفس الحاجة، بس بيشتغلوا مع بعض زي الفريق:

◆ ModelBinder

- يركز على ال Binding: تحويل البيانات الخام إلى Model.
- يحصل قبل ال Validation.
- لو في مشكلة في الربط (زي نوع غلط، مثلاً string بدل int)، هضيف خطأ تلقائياً إلى ModelState.
- سريع وتلقائي، بس مش بيتحقق من القواعد اللي انت حددتها.

◆ ModelState

- يركز على ال Validation: التحقق من الصحة بعد الربط.
- يعتمد على ال Data Annotations أو Custom Validators.
- بيسمحك تضيف أخطاء يدوياً، زي:

```
ModelState.AddModelError("Name", "الاسم غير صالح");
```

- لو IsValid = false، معناها في أخطاء، وبتقدر تعرضها في ال View بـ @Html.ValidationMessageFor tag helper او باستخدام ال tag helper.

الفرق الرئيسي: ModelBinder هو "المدخل" اللي بيحيب البيانات، و ModelState هو "الحارس" اللي بيتأكد إنها سليمة.

2. إزاي يشتغلوا مع بعض؟

العملية بتبدأ كده:

1. الطلب بيجي (GET أو POST).
2. ال ModelBinder يربط البيانات بال Model.
3. لو في مشاكل في الربط، يضيف أخطاء إلى ModelState.
4. بعد كده، ال Validation يشتغل بناءً على ال Attributes، ويحدث ModelState.
5. في الكونترولر، تشيك ModelState.IsValid عشان تقرر إيه تسوي.

مثال كامل:

تخيل يوزر بيبعت فورم بدون اسم:

- ال ModelBinder بيربط Age بس، و Name هيكون null.
- Validation بيكتشف إن Name مطلوب، فيضيف خطأ إلى ModelState.
- ModelState.IsValid = false، فيرجع ال View مع رسائل الأخطاء.

3. أخطاء شائعة في استخدامهم

- ♦ **التعامل مع Complex Models:** لو الموديل معقد (زي nested objects)، ال DefaultBinder مش دائماً بيشتغل كويس، فاكتر Custom Binder.
- ♦ **تجاهل أو نسيان ModelState:** لو مش بتشيك ModelState، ال Model هيمر حتى لو غلط.
- ♦ **Overposting:** اليوزر ممكن بيعت بيانات إضافية، وال Binder هيربطها لو مش محمي، فاستخدم [Bind(Include="Name, Age")]

4. نصائح مهمة للتعامل مع ModelState و ModelBinder

- ✓ **استخدم Data Annotations دائماً:** زي [StringLength]، [Required] عشان ال Validation يكون سهل.
- ✓ **استخدم ال [Bind Attribute]:** لو عايز تستثني حقل معين من الربط، استخدم [Bind(Exclude="SecretField")]

✓ اكتب Custom ModelBinder لو البيانات جاية من مصدر غريب، زي XML أو Custom Format.

✓ في ال Views، استخدم @Html.ValidationSummary عشان تعرض كل الأخطاء في مكان واحد. او باستخدام ال tage helper وببقي ده شكلها

```
<div class="validation-summary-errors" asp-validation-summary="All"></div>
```

✓ لل APIs، رجع BadRequest(ModelState) لو IsValid = false.

✓ اختبر ال Binding مع أنواع مختلفة: primitives, collections, complex types.

✓ انتبه لل Security: استخدم AntiForgeryToken مع ModelState عشان تمنع CSRF Attacks.