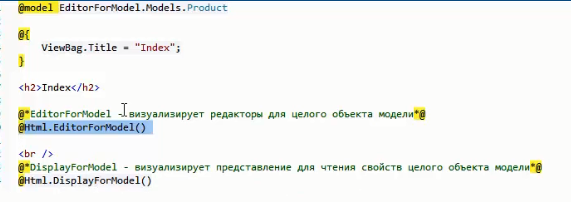
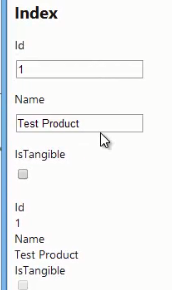


Полностью для всей модели

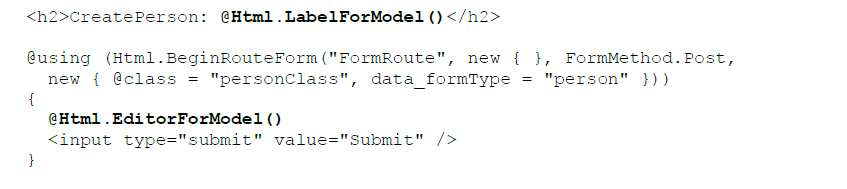




Различия (не берется в теги)

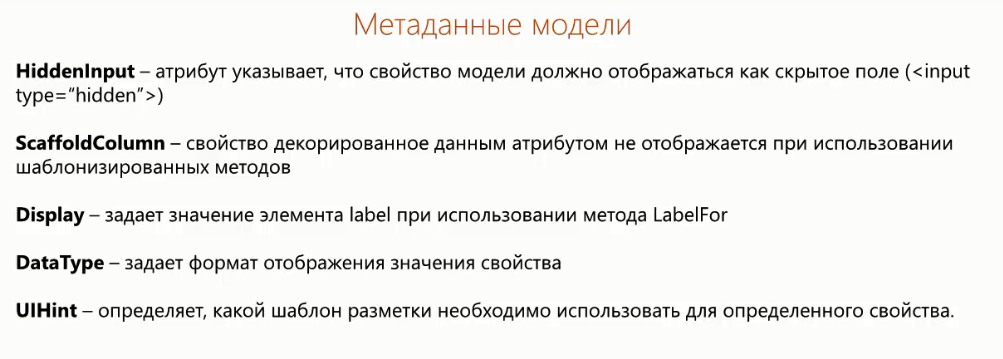


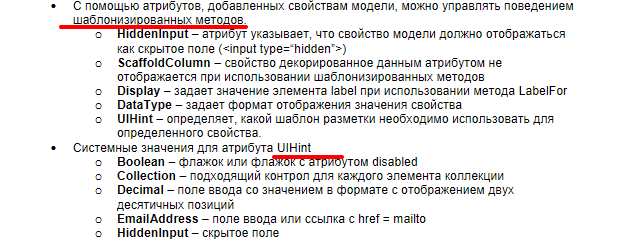
Для автоматического отображения

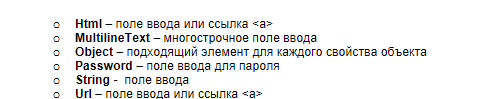


Только @Html.LabelForModel(), DisplayForModel() такую расстановку элементов не предоставит

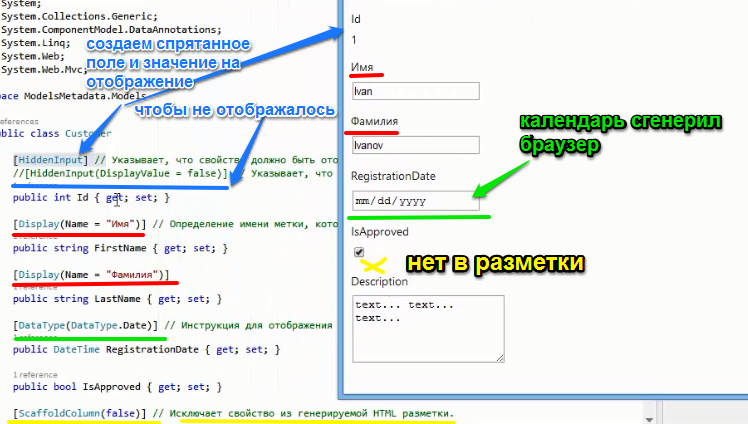


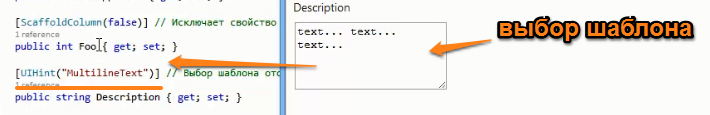






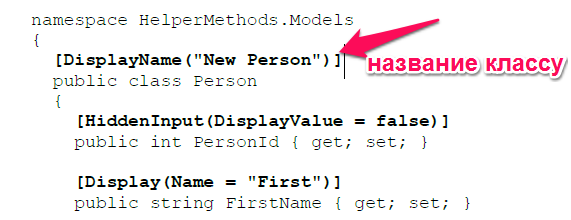
[DisplayFormat(DataFormatString = "{0:dd MMM yyyy}", ApplyFormatInEditMode = true)]





DisplayName,

Этот атрибут можно применять к классам,



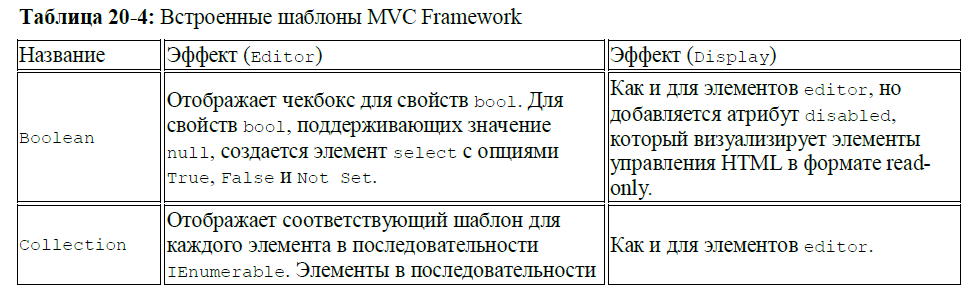
*UIHint*

***Используя атрибут*** *UIHint****, будьте внимательны. Если мы выберем для свойства***

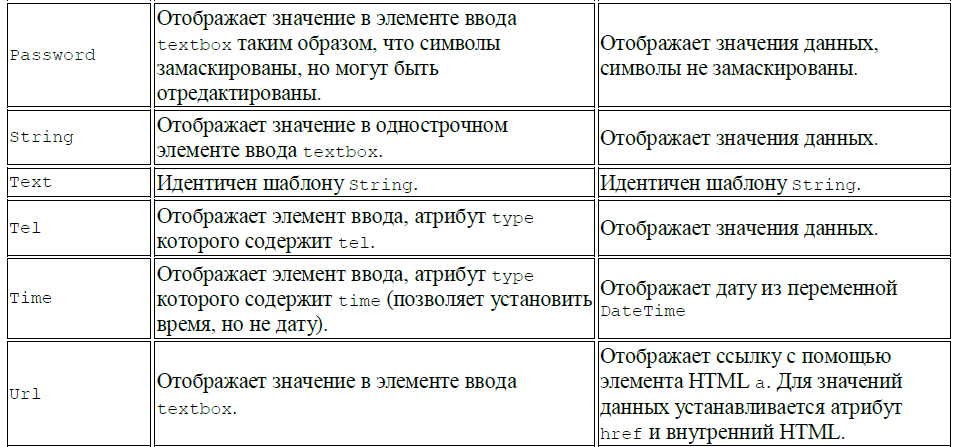
***шаблон, который не может работать с его типом, то получим исключение:***

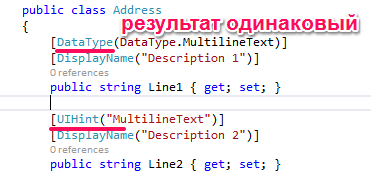
***например, применив шаблон*** *Boolean* ***к свойству*** *string****.***

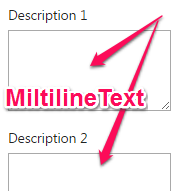
Получив объект для обработки, шаблонный вспомогательный метод сгенерирует HTML только для простых типов свойств и проигнорирует все свойства, которые представляют собой сложные объекты.







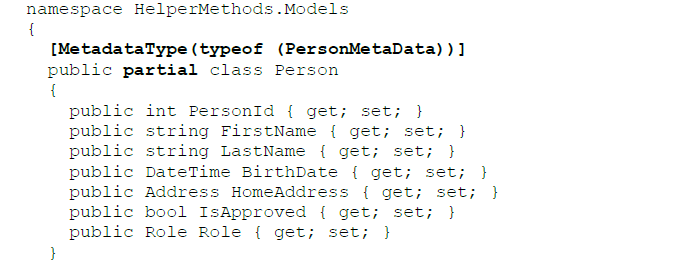




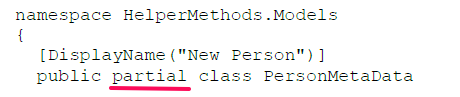
[MetadataType(typeof (PersonMetaData))]

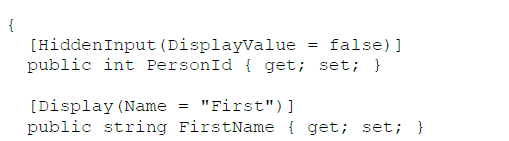
public partial class Person

Не всегда можно применить метаданные к классу сущности модели.(т.е. декорировать свойства атрибутами) Так обычно происходит, когда классы моделей генерируются автоматически, например, с помощью инструментов ORM, таких как Entity Framework



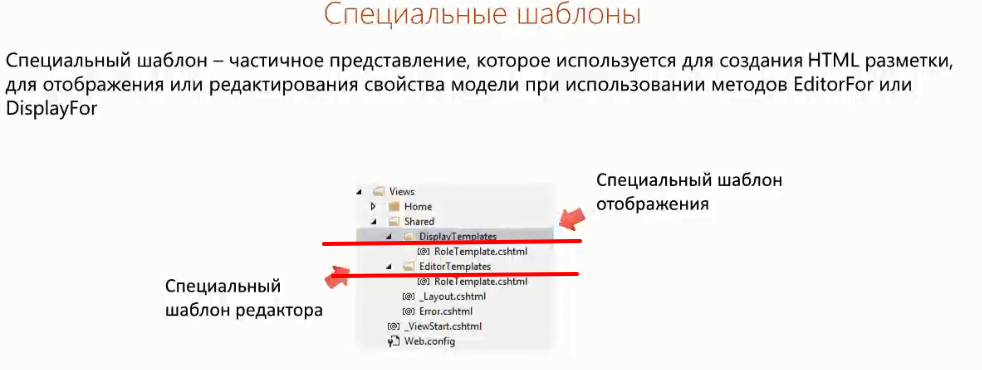
Дополняющий класс должен быть определен в том же пространстве имен и также должен быть частичным.

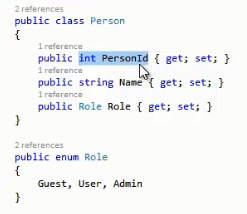


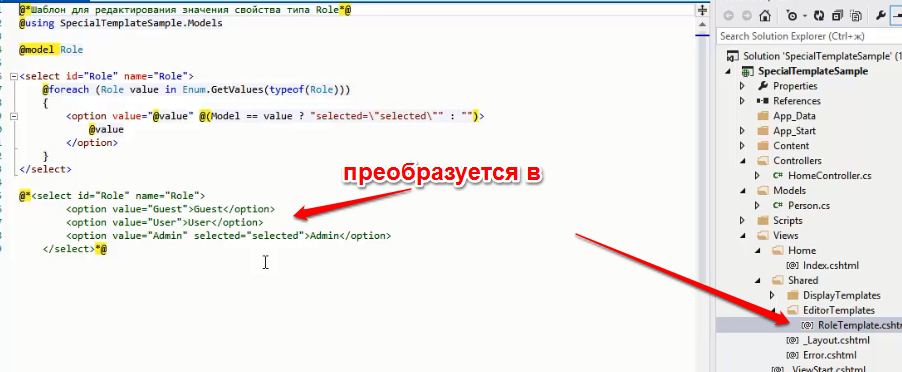


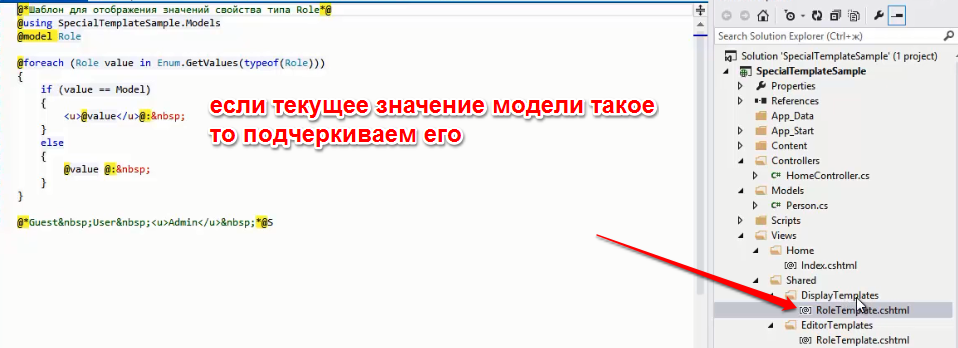
Дополняющий класс должен содержать только те свойства, к которым мы хотим применить

метаданные – нет необходимости воспроизводить все свойства класса

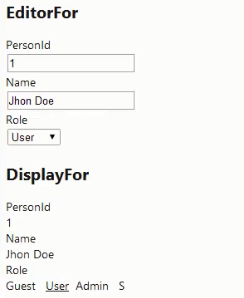


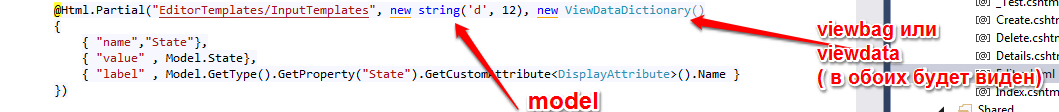


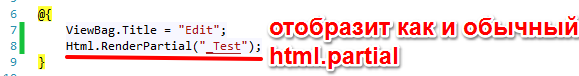


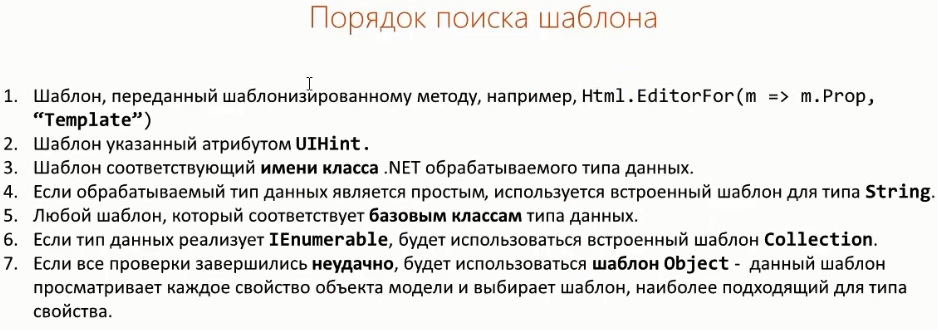


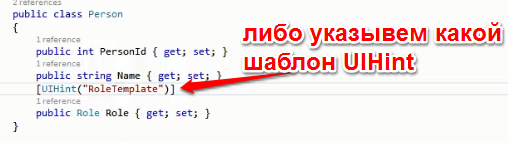


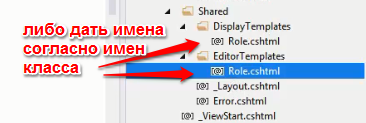




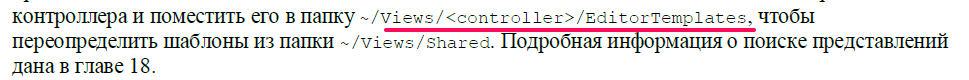


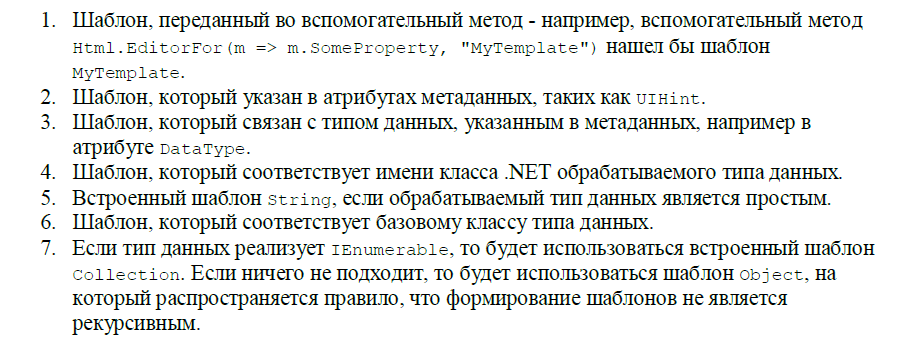




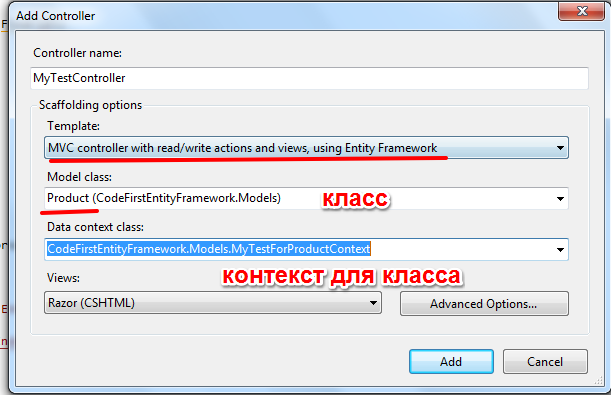


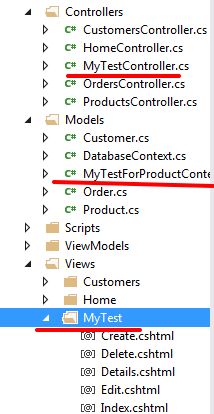
Поиск более подробно

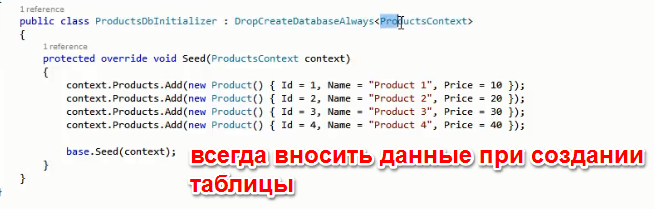




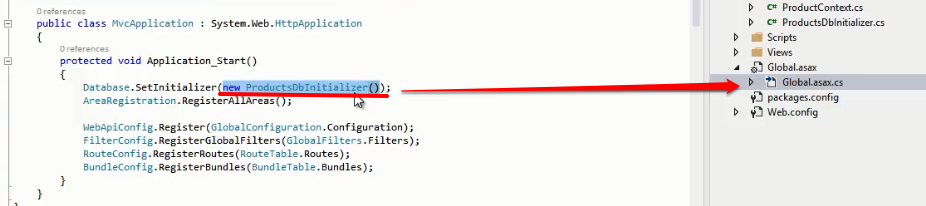
Привязка к БД Entity Framework





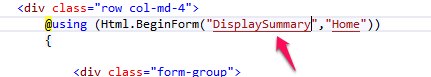


Устанавливаем данное поведение при загрузке приложения



Связывание модели полученных от cookie ()

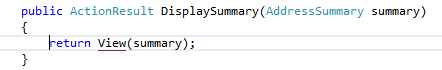
Передаем данные одного класса Person

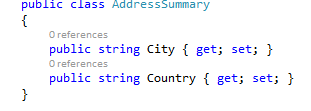




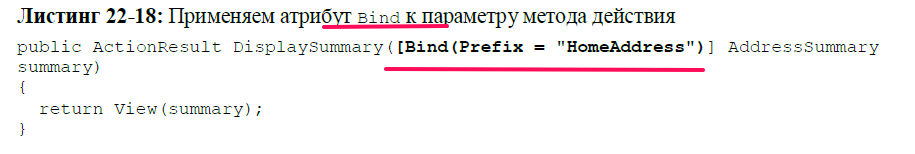
Но обрабатывать будет другой класс, но с такими же имена свойств

(в таком случае связать не сможет)

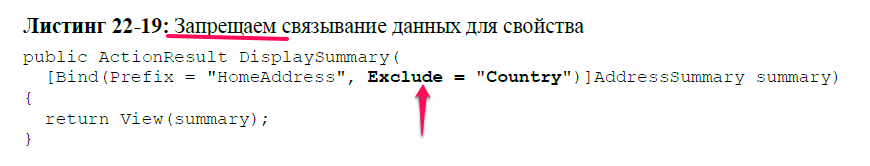




Нужно применить атрибут Bind



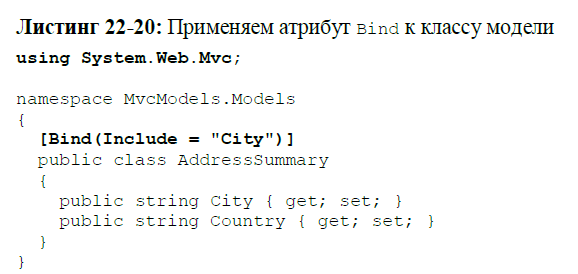
Если запретить связывание Exclude, для какого либо из свойств



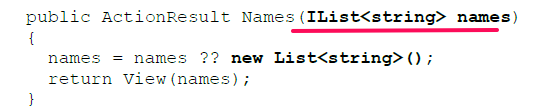
Include

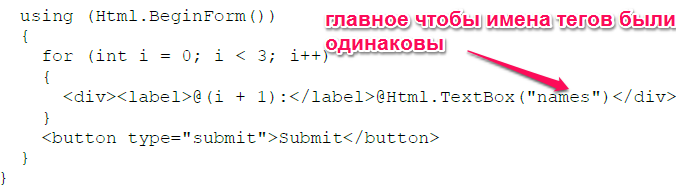
(В качестве альтернативы, вы можете использовать свойство Include, чтобы указать только те свойства, которые должны быть связаны для этой модели; все остальные свойства будут игнорироваться).

Можно применить к модели

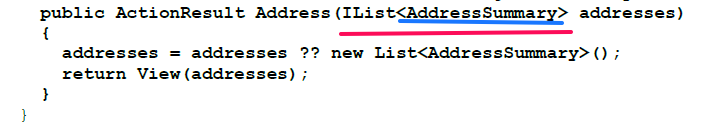


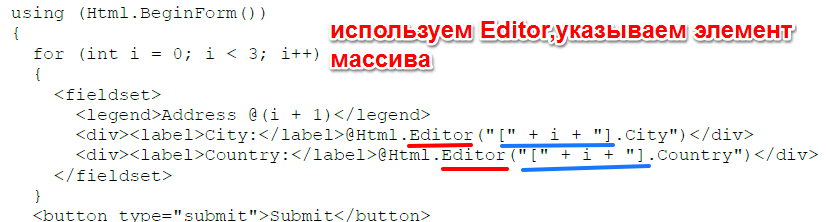
Принимает коллекцию IList (можно массив указать)





Связывание с коллекцией пользовательских типов





Результат

