**Data Annotations**



Required when not null.. StringLength.. 5 is max..

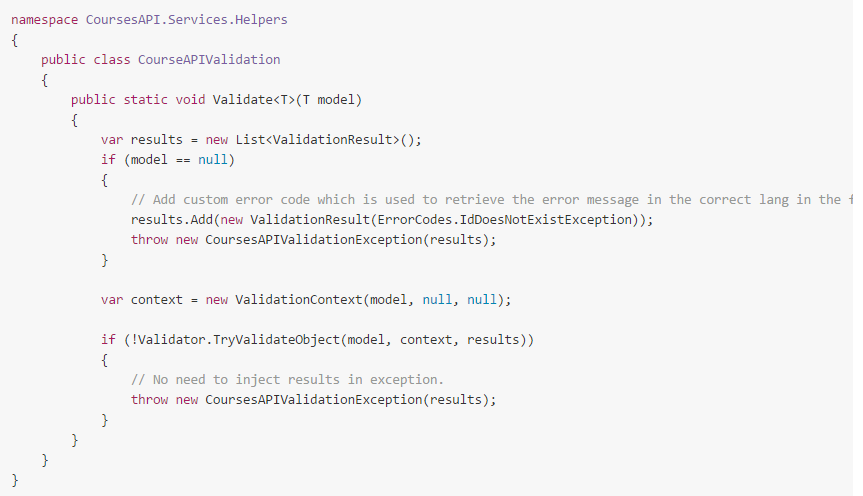
In **controller**.. can see if.. valid..



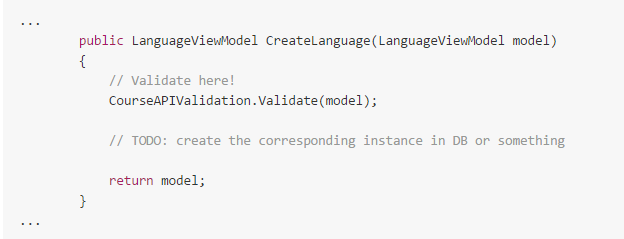
**MOVING VALIDATION TO THE SERVICE LAYER**

If we don‘t want it in the controller..

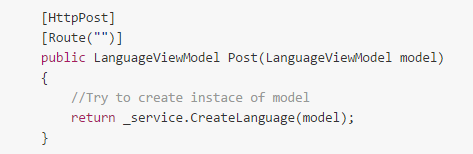
Create a class with a function that validates a model of any type:



Would allow us to have a method in our service for creating a LanguageViewModel instance like this...



Then our controller action will look like this.. a lot cleaner:



**Handling Validation Errors**

Controller needs to check the model state and respond.. can also create an **action filter** to check the model state before the controller action is invoked. Or you could also create a **filter which handles ALL exceptions**, throw a validation specific exception and grab it on the filter as the code above does.

I.e. you create CourseAPI.Filters..

AppExceptionFilter : ExceptionFilterAttribute.. override OnException... if CourseApiValidationException.. etc. Do something..

To filter all Web API controllers.. we add an instance of the filter to the HttpConfiguration.Filters collection during configuration..

Public static void Register(HttpConfiguration config) { config.Filters.Add(new AppExceptionFilter());