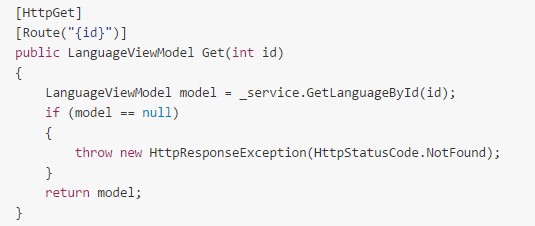
Köstum villu á service layer.. fer í Web API.. nema handled differently..

**HttpResponseException**

WEB API throws uncaught exception => most translated to status code 500.. Server error..

**HttpResponseException** is a special case.. returns any HTTP status code that you specify in the exception constructor...



Can also construct the entire response message..



**Exception Filters**

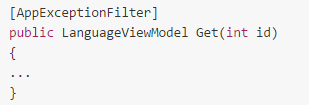
Can customize how Web API handles exceptions by **writing an EXCEPTION FILTER**..

An **exception filter is executed when a controller method throws ANY UNHANDLED EXCEPTION** that is **not an HttpResponseException**..

Can register a WEB API Exception filter:

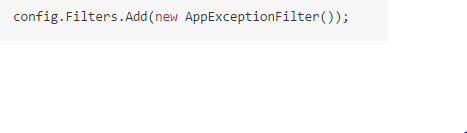
* By action
* By controller
* Globally

Action:



Same for controller...

And to all:



**EXCEPTION FILTER ALLOWS METHODS / OTHER LAYERS TO CONTAIN NO EXCEPTION LOGIC, ALL IN THE EXCEPTION FILTER ATTRIBUTE CLASS**

**Logging Unhandled Exceptions**

No easy way in Web API to log or handle errors globally... NUMBER OF CASES EXCEPTION FILTERS CAN‘T HANDLE:

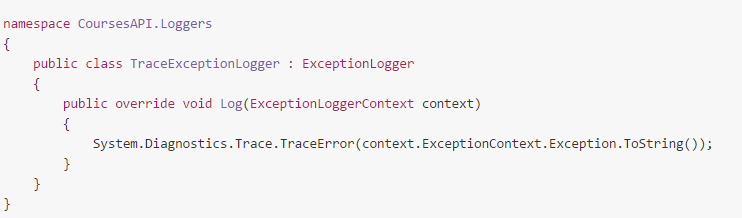
* Exceptions thrown from controller constructors
* Thrown from message handlers
* Thrown during routing
* Thrown during response content serialization

The **ExceptionLogger** class which implements the IexceptionLogger interface can be used to log all unhandled exceptions. If an unhandled exception occurs, the **Log** method will be called DIRECTLY after the exception and before all ExceptionFilter attributes defined for the controller..

Exception loggeres provide you a **single point** into which you can plug in a service that would **log information aby any exceptions occurring in the Web API pipeline** – regardless where they originated from...

**EXCEPTION LOGGERS ALWAYS CALLED**..

Can register multiple ExceptionLoggers in WebAPI..



Can also use RayGun.. IexceptionLogger implementations are added to the config.Services in the WebAPIConfig class..

**Global IExceptionHandler**

The IexceptionHandler handles all UNHANDLED EXCEPTIONS from all controllers... If an exception occurs.... **IexceptionLogger CALLED FIRST**, then the **CONTROLLER EXCEPTIONFILTERS**, and if still unhandled.. **IexceptionHandler**..



In the Web API config.. IexceptionHandler has to be replaced unlike the IexceptionLogger... Only 1ExceptionHandler can be used for the service..

* **Exception loggers** are the SOLUTION to **SEEING ALL UNHANDLED EXCEPTIONS CAUGHT BY WEB API**
* **Exception handlers** are the solution for customizing all possible **RESPONSES TO UNHANDLED EXCEPTIONS** caught by Web API
* **Exception filters** are the easisst solution for **PROCESSING THE SUBSET OF UNHANDLED EXCEPTIONS** related to a **SPECIFIC ACTION OR CONTROLLER**