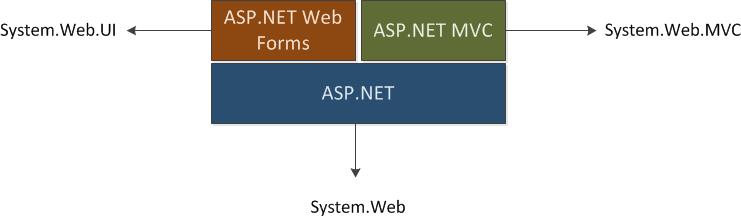
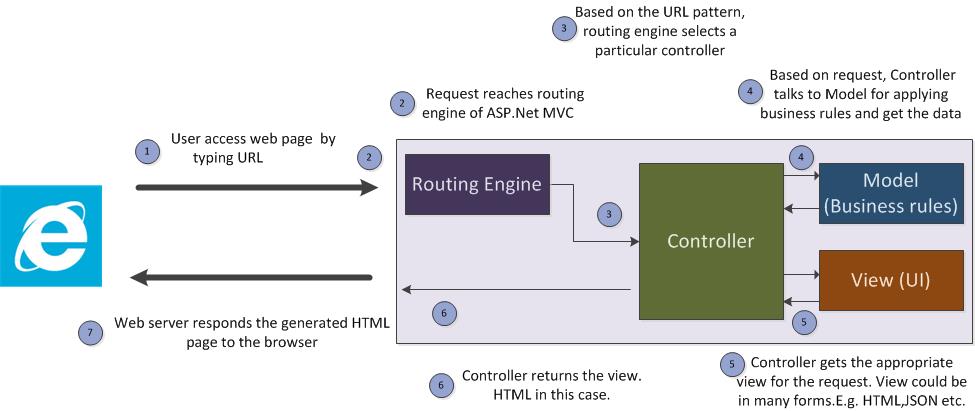
**ASP.NET Layers**

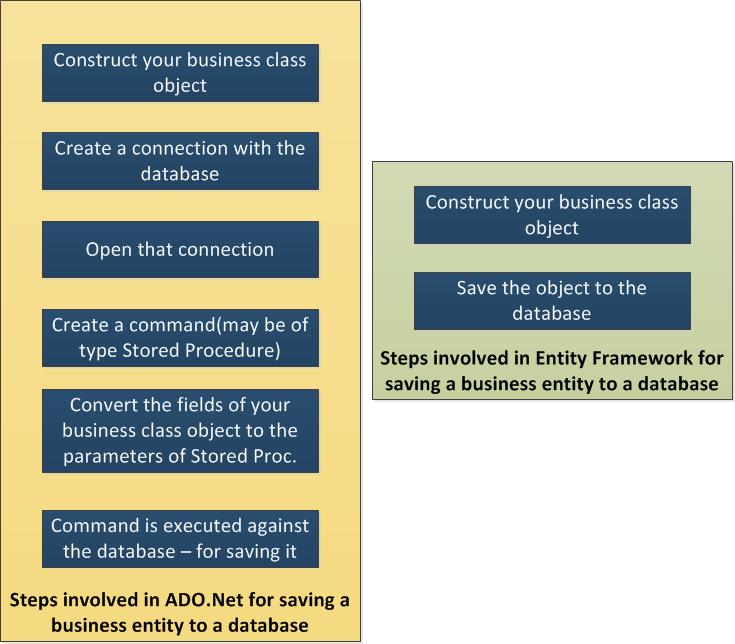




Following are the sequence of steps that happens when user tries to access a web page in ASP.Net MVC application

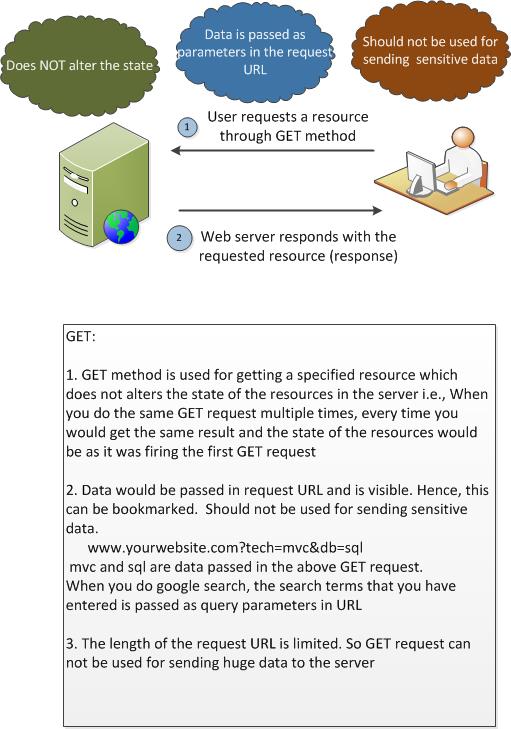
1. User types the URL in the browser([google.com/web/](http://www.google.com/web/) or [www.google.com/images/](http://www.google.com/images/) ) and types the query
2. Request reaches the web server and it’s forwarded to the routing engine of ASP.Net MVC
3. Based on URL pattern, routing engine selects appropriate controller. If user has typed [google.com/web/](http://www.google.com/web/) , then routing engine would controller by name ‘web’. Or if the user tried to access [www.google.com/images/](http://www.google.com/images/), routing engine would select ‘images’ controller. Each controller is responsible for receiving the request and orchestrating with model and view to deliver the appropriate response. Let’s assume that user has accessed [www.google.com/images/](http://www.google.com/images/) and images controller is selected.
4. Controller talks to the database using the model layer to get the appropriate images for the entered query. If the user ‘s query is ‘FIFA’, model gives the information of all images related to FIFA
5. Once the controller receives the required data, it gets the relevant View to show it to the user. Instead of 10 blue links (that we get when we search in web), Google shows the thumbnail images with infinite scroll.
6. Images controller returns this view with thumbnails.
7. The requested resource is sent back to the requested browser.

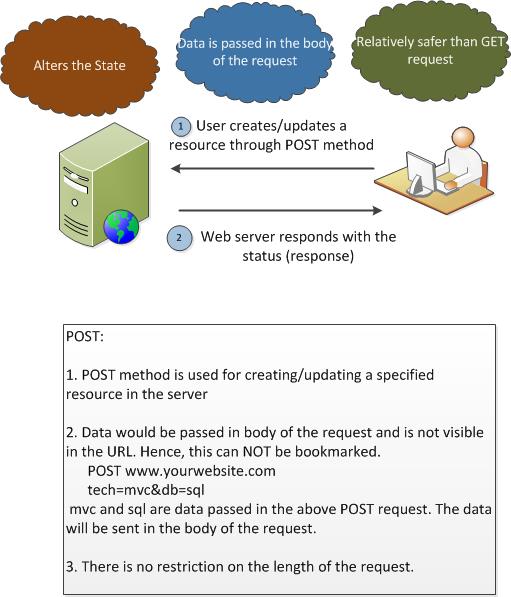


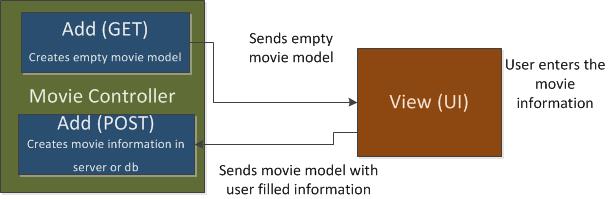


In order to use DbContext API, you need to do the following steps.

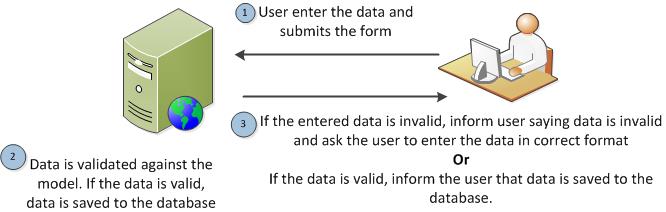
1. DbContext is available in System.Data.Entity namespace. You need to include this namespace if you use DbContext.
2. Create a class (MovieDBContext in below code) which inherits from DbContext so that we can use the methods of DbContext API.
3. Have all the classes(only Movie class in our case) which you want to persist/query the database using DbSet.



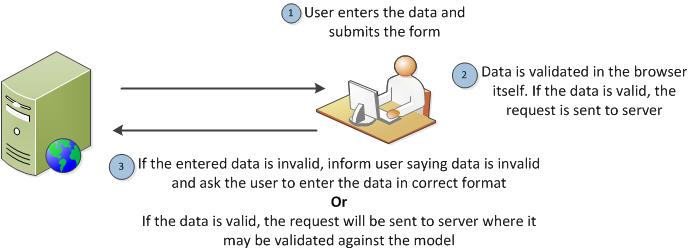




**Server side validation:**



#### Client Side Validation:



#### Client & Server Validation:

