

1. Struktura MVC składa się z 3 warstw. Model, View, Controller.
 - a. Model odpowiada za określenie struktury, na której pracujemy. Dodatkowo może zawierać metody bezpośrednio z nim związane,
 - b. View odpowiada za prezentowanie danych,
 - c. Controller do komunikacji pomiędzy tymi elementami i wpływaniu na obecny stan modelu.
2. Konwencja nazw:
 - a. Model -> PascalCase w liczbie pojedynczej + „Model”. W przypadku modelu mającego znajdować się bezpośrednio w widoku można skorzystać z końcówki ViewModel.
 - b. View -> PascalCase w liczbie pojedynczej. W przypadku Paritali: „_” + PascalCase + „Partial”.
 - c. Controller -> PascalCase w liczbie mnogiej + Controller
 - d. Dla każdego Controllera tworzony jest folder w modelach o tej samej nazwie z zakończeniem „Models”/”ViewModels”. Następnie jeśli controller nie korzysta tylko z jednego modelu to dla metod tworzone są modele o tej samej nazwie. Analogicznie z widokami.
3. Dane można przekazywać w formie modelu, bądź za pośrednictwem ViewBag. W wersji Core występują również TempData i ViewData.
4. Mapowaniem możemy zarządzać w RouteConfig.cs. Domyślnie jest to {adres_witryny}/{controller}/{action}/{id}", gdzie: controller = "Home", action = "Index", id = UrlParameter.Optional
5. Zarządzać żądaniami http można przy pomocy atrybutów. [HttpPost], [HttpGet] itd.
6. Aby kontroler odbierał tylko żądania od nas należy skorzystać z [ValidateAntiForgeryToken]. W widoku należy umieścić @Html.AntiForgeryToken(). W nowszych wersjach nie jest to wymagane.
7. Aby sprawdzić, w kontrolerze czy dany model zawiera wszystkie wymagane pola wystarczy skorzystać z ModelState.IsValid. Aby zdefiniować wymagane pola w modelu należy skorzystać z [Required].