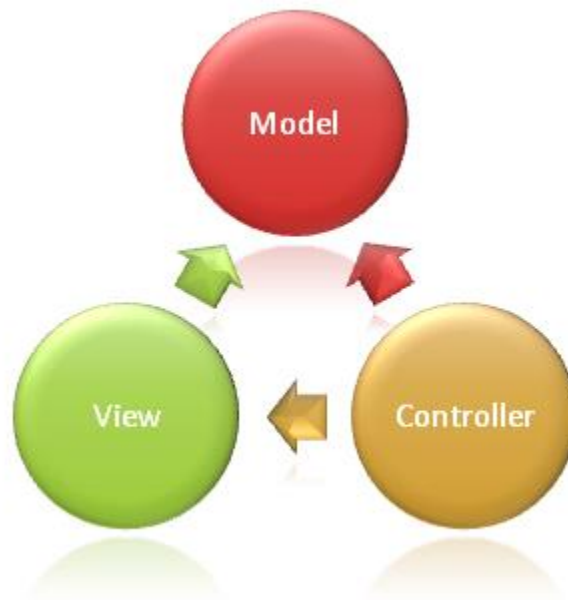


MVC

*Model-View-Controller*¹(MVC) архитектура раздваја апликацију на три групе компоненти: *Models*, *Views* и *Controllers* (Модел, прикази и контролери). На тај начин се постиже раздвајање проблема (акција) по надлежности. Помоћу овог обрасца, захтеви корисника се преусмеравају (рутирају) на контролер који је одговоран за рад са моделом ради извршавања корисничких радњи и / или добијања резултата упита. Контролер одабере приказ који ће се приказати кориснику и пружа му све податке модела који су му потребни.

На слици испод можемо видети све три компоненте апликације и њихову међусобну повезаност.



Ово раздвајање одговорности помаже нам да поједноставимо апликацију, јер је лакше кодирати, исправљати грешке и тестирати модел, приказ или контролер који има само један посао него читаву апликацију.

Напомена: Компоненте *View* и *Controller* су зависни од компоненте *Model*. С друге стране, компонента *Model* је потпуно независна од друге две компоненте. Ово је једна од кључних предности раздвајања апликације на компоненте по надлежности. Другим речима, омогућено нам је да градим, тестирамо и ажурирамо модел независно од визуелне презентације.

¹ <https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-3.0>

Одговорности модела

Модел у MVC апликацији представља пословну логику (логику решења проблема) или операције које би ова апликација требала обављати.

Одговорност компоненте View

Views је компонента која је одговорна за приказ садржаја у корисничком интерфејсу. У *ASP.NET* технологији се користи *Razor view engine* за ембедовање *.NET* кода у *HTML* елементе. У овој компоненти не би требало бити много логике, а ако је има, она треба да буде везана за презентацију резултата.

Одговорности Контролера

Контролери су компоненте који обрађују (хендлују) интеракције са корисником, раде са моделом, и на крају бирају приказ који ће се извршити. У *MVC* апликацијама, компонента *View* само приказује информације, док контролер обрађује кориснички унос и бира излазне податке. Контролери би требали да буду релативно једноставни: Не треба да садрже бизнис логику (логику решавања проблема) – ово треба да буде садржано у моделу.