MVC Nedir? MVC Yaşam Döngüsü (Life Cycle)

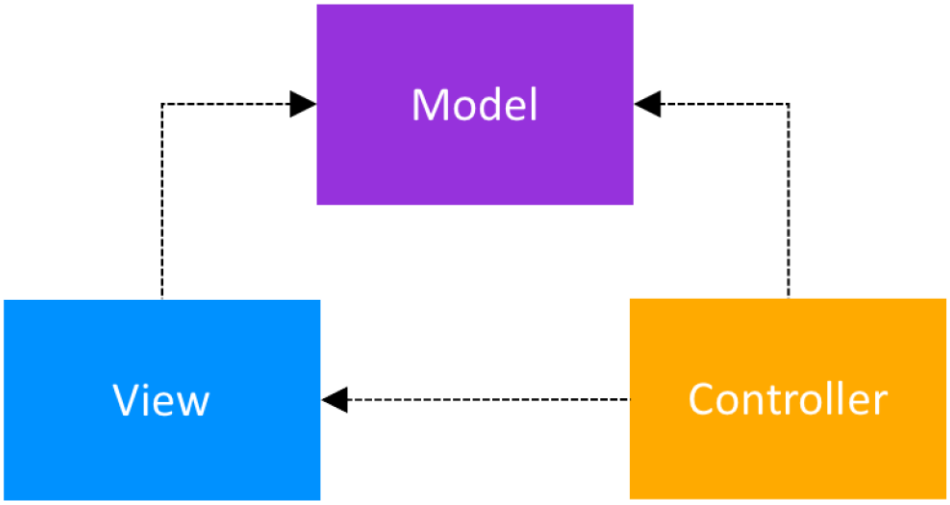
MVC Nedir?

MVC, Yazılım Mühendisliği'nde önemli bir yere sahip architectural patterns (yazılım mimari desenleri)'ın bir parçasıdır. Model, View ve Controller kelimelerinin baş harflerinden oluşan MVC (Model-View-Controller), 1979 yılında Tygve Reeskaug tarafından oluşturulmuş ve yazılım gelişmede bir çok projede kullanılmıştır. Son dönemlerde Microsoft'un MVC desenini Asp.Net teknolojisi ile birleştirmesi ile popülaritesi daha da artmıştır.

MVC ile ilgili en yanlış bilgi, MVC'nin Microsoft tarafından çıkartıldığı düşüncesidir. Yukarıdaki paragrafta da bahsedildiği üzere, MVC'nin Asp.Net'e entegre edilmesinden önce bu deseni bir çok (.Net) yazılım

geliştiricisi bilmemekteydi, bilse de kullanmıyorlardı. Asp.Net MVC'nin gelişiminin ardından MVC'ye ilgi oldukça artmış görünüyor.

MVC deseni, 3 katmandan oluşmaktadır ve katmanları birbirinden bağımsız (birbirini etkilemeden) olarak çalışmaktadır. Bu sebeple çoğunlukla büyük çaplı projelerde projelerin yönetiminin ve kontrolünün daha rahat sağlanabilmesi için tercih edilmektedir. MVC ile geliştirilen projelerde projenin detaylarına göre bir çok kişi eş zamanlı olarak kolaylıkla çalışabilmektedir.



Model Nedir?

Model, MVC'de projenin iş mantığının (business logic) oluşturulduğu bölümdür. İş mantığıyla beraber doğrulama (validation) ve veri erişim (data access) işlemleri de bu bölümde gerçekleştirilmektedir.

Model tek katmandan oluşabileceği gibi kendi içinde birden fazla katmandan da oluşabilir. İç yapılandırma projenin büyüklüğü ile yazılım geliştiricinin planlamasına kalmış bir durumdur. Eğer proje büyük çaplı ise modeli birden çok katmana ayırmak projenin yönetimi açısından faydalı olacaktır.

View Nedir?

View, MVC'de projenin arayüzlerinin oluşturulduğu bölümdür. Bu bölümde projenin kullanıcılara sunulacak olan HTML dosyaları yer almaktadır. Projenin geliştirildiği yazılım dillerine göre dosya uzantıları da değişebilmektedir. Projelerin büyüklüğüne göre dikkat edilmesi gereken bir nokta ise, klasörlemedir.

Eğer bir web projesi geliştiriyorsanız, projenin View'larının yer aldığı klasörlerinin hiyerarşisi, ilerleyen dönemlerde karmaşıklığa sebep olmaması için dikkatli yapılmalıdır. Kimi yazılım geliştiriciler web projelerinde HTML dosyaları ile Javascript, CSS ve resim dosyalarını aynı klasör içinde barındırmaktadır. Ufak bir ayrıntı gibi görünse de projenin ilerleyen dönemlerinde ciddi problemler oluşturmaktadır.

View'ın bir görevi de, kullanıcılardan alınan istekleri controller'a iletmektir.

Controller Nedir?

Controller, MVC'de projenin iç süreçlerini kontrol eden bölümdür. Bu bölümde View ile Model arasındaki bağlantı kurulur. Kullanıcılardan gelen istekler (request) Controller'larda değerlendirilir, isteğin detayına göre hangi işlemlerin yapılacağı ve kullanıcıya hangi View'ın döneceği (response) belirtilir.

MVC'nin Yaşam Döngüsü(Life Cycle)

MVC'nin parçaları olan Model, View ve Controller'ın ne olduğu yukarıdaki bölümde anlatıldı. Şimdi bu bilgileri toparlayıp MVC'nin yaşam döngüsünü (çalışma prensibini) detaylıca inceleyelim.

1) HTTP Request: Sizin her ASP.NET MVC uygulamasını görüntülemek istemeniz bir request(istek) tir.

Bu istediğinizi HTTP üzerinden IIS tarafından alınır. Her yaptığınız istek Server tarafından bir yanıtla
son bulması gerekir.
2) Routing: ASP.NET MVC uygulamasını her istek yaptığınızda, yaptığınız yanıt UrlRoutingModule
HTTP Module tarafından durdurulur. UrlRoutingModule bir isteği durdurduğu zaman, gelen istek
RouteTable'dan hangi Controller tarafından üstleneceğine karar verilir.
3) Controller: RouteTable'dan gelen route bilgisine göre Controller hangi Action'ı çalıştıracaksa o
View çalıştırılır. View, Controller tarafından render edilmez. Controller tarafından geriye ViewResult
döndürülür.
4) ViewResult: ViewResult, View'i render etmek için aktif View Engine'i çağırır.
5) ViewEngine : Bir CSHTML dosyayı oluşturduğunuzda içerisindeki script ve markuplar, Razor View
Engin tarafından bazı ASP.NET API'lerini sayfalarınızı HTML'e çevirmek için kullanır.
6) View: View Engine tarafından HTML'e çevirilen kodlar kullanıcıya sunulur.
7) Response: HTTP üzerinden View kullanıcıya gösterilir.