

بسم الله الرحمن الرحيم



دانشگاه علامه طباطبائی

## الگوی طراحی MVC

درس: تاثیر فناوری اطلاعات در کسب و کار

استاد: دکتر مسعود حجاریان

شادی مقدادی

نیمسال اول سال 91-92

## مقدمه

نرم افزارهای اولیه نماهای یکپارچه و خود مختاری را به کاربر ارائه می کردند که در نتیجه تعامل کاربر و فراهم شدن ورودی ها ، پردازش بر روی داده ها توسط بخش داخلی نما ها انجام میشد که در نهایت یا نمای جدیدی به کاربر ارائه می شد و یا نمای قبلی بروز می شد.

در واقع نرم افزارها با هر میزان پیچیدگی بصورت بلوک یکپارچه ای از کد نوشته می شد که برنامه نویسی ساختارمند سعی در شکستن آن به اجزاء کارآمد و قابل استفاده مجدد می کرد. از طرفی واسط کاربر همچنان مجموعه ای از نماهای خود مختار باقی می ماند که هرکدام مدیریت موقعیت ، ضبط اعمال کاربر و تصمیم گیری مراحل بعدی کار را بعهده داشتند.

بنابراین کد ها شامل فراخوانی هایی از واسط کاربر به هسته مرکزی خود می شدند و لذا وابستگی میان اجزاء مختلف ، انعطاف پذیری ، آزمایش پذیری و در نتیجه نگهداری و مدیریت نرم افزار را عملاً ناممکن می کرد. جستجو برای یافتن مدل بهتری منجر به معرفی MVC شد. هدف الگوی معماری MVC صرفاً یکپارچگی در معماری نرم افزار است و به کمک آن بدست گیری نرم افزار در راستای مدیریت و توسعه به سهولت انجام می گیرد.

## الگوی MVC

با MVC سه بازیگر با نقش های مشخصی وارد میدان شدند. بعبارتی نرم افزارها حداقل به سه بخش مجزا تقسیم می شوند:

**Model:** که بیانگر آیتم ها ، اعمال و قوانینی است که در محدوده موضوع نرم افزار با معنی هستند و همچنین مسوولیت نگهداری موقعیت کلی نرم افزار را نیز بعهده دارد و از هر مفهومی وابسته به واسط کاربر جدا است.

**View:** که چگونگی ارائه دادن بخشی از مدل را در قالب عناصر قابل نمایش واسط کاربر توصیف می کند و خالی از هرگونه منطقی است.

**Controller:** درخواست های رسیده را بررسی کرده ، اعمالی را روی مدل انجام می دهد و **View** مناسبی را برای ارسال به کاربر ارایه می کند.

**MVC** در سال ۱۹۷۹ میلادی توسط **Trygve Reenskaug** و سپس کار بر روی اسمال تاک در **XEROX PARC** معرفی شد و گونه های متفاوتی برای آن بوجود آمده و بررسی شده است که **Stephen Walther** در مقاله ای سیر تکاملی **MVC** را شرح داده است. آنچه امروزه مقبولیت دارد ، در واقع گونه تحت وب **MVC** می باشد که آن را **Model2** می نامند و برای استفاده در **JSP** پیشنهاد شده است . این مدل بیشتر محبوبیت خود را مدیون چهارچوب **Struts** است.

اما به هر حال تمام این گونه ها یک هدف اصلی را دنبال می کنند : جدایی اهمیت ها (**SoC**). **SoC** یکی از اصول اولیه و اساسی طراحی نرم افزار است که به کمک آن می توان به مفاهیمی چون **high cohesion** و **low coupling** دسترسی پیدا کرد و در سال ۱۹۷۴ توسط **Edger W. Dijkstra** در مقاله ای با نام **On the Role of Scientific Thought** معرفی شده است. و آنچه در **MVC** بیشتر مورد توجه است جدایی اهمیت ها در لایه نمایش می باشد. **MVC** را می توان الگویی برای طراحی کل نرم افزار در نظر گرفت ولی در واقع بیشترین فعالیت **MVC** بر روی **View** و **Controller** است و تنها به **Model** ای برای کار بر روی داده های **View** نیاز دارد. بنابر این **Model** می تواند لایه منطقی موجود و یا مجموعه ای از کلاسهای تعریف شده برای دسترسی به الگوی **MVC** باشد .

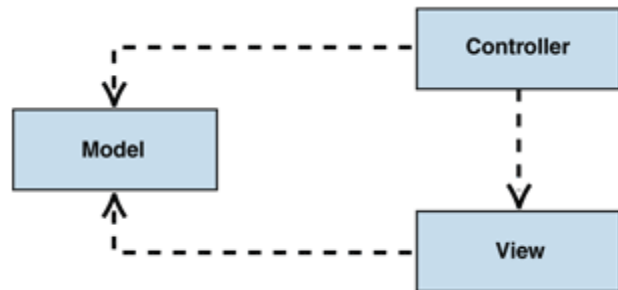
برای یک نرم افزار ساده، پیاده سازی **MVC** به مفهوم ایجاد **View** و **Controller** های مورد نیاز به همراه یک لایه منطقی تحت عنوان **Model** می باشد. اما در حالت پیشرفته از **MVC** برای جداسازی اهمیت ها و رسیدن به آزمایش پذیری در لایه نمایش استفاده می شود.

**Model2:** در **Model2** جریان درخواست ها از پویشگر مستقیماً به یک ناظر اولیه که بعنوان جداکننده **HTTP** پیاده سازی می شود و در **ASP.NET** به آن **HTTP Module** می گوئیم میرسد. عبارت دیگر واسط کاربر **Model2** ورودی های **HTML** را پیشنهاد می کند که بواسطه آنها لینکی فراهم می شود و در نهایت یک ارسال **HTTP** رخ می دهد. ناظر اولیه در وب سرور درخواست ها را

ضبط کرده و مطابق ساختار URL آنها را بررسی می کند و تصمیم می گیرد که کدام کنترلر MVC باید در برابر درخواست نمونه سازی شود. بعد از شناسایی کنترلر، فرمانی که روی مدل تاثیر می گذارد فراخوانی می شود، سپس کنترلر به نما فرمان می دهد تا HTML مناسب را تولید کند و نما اطلاعات دقیق را بطور مستقیم از کنترلر می گیرد و خروجی به ناظر اولیه فرستاده می شود تا به پویشگر ارسال شود.

گرچه MVC انواع مختلفی دارد، کنترل جریان عموماً به صورت زیر است:

- کاربر به نوعی با واسط کاربری در تعامل است. (برای مثال با فشردن دکمه ماوس)
- کنترلر گر رویداد وارده از واسط کاربری را معمولاً از طریق یک کنترلر گر رویداد ثبت شده و یا callback کنترلر می کند و رویداد را به یک عمل مناسب کاربری قابل فهم برای مدل تبدیل می کند.
- کنترلر گر، مدل عمل کاربری را اعلام می کند که احتمال دارد منجر به تغییری در وضعیت مدل شود. (برای مثال کنترلر گر، سبد خرید کاربر را به روز می رساند).
- یک نما، از مدل به منظور تولید یک واسط کاربری مناسب پرس و جو می کند. نما داده خودش را از مدل می گیرد. در برخی پیاده سازی ها کنترلر گر ممکن است دستورالعملی عمومی به نما بدهد تا خودش را بارگذاری کند. در سایر پیاده سازی ها نما به طور خودکار توسط مدل از تغییرات در ناظر وضعیت مطلع می شود که نیازمند به روز رسانی صفحه است.
- واسط کاربری منتظر تعاملات کاربری بیشتری می ماند که چرخه کنترل جریان را از نو آغاز می کند.



بعضی از پیاده سازی ها مانند فرم های کنسرسیوم وب جهان شمول از مفهوم نمودار وابستگی نیز برای خودکار کردن نماها زمان تغییر مدل استفاده می کنند.

هدف MVC با جداسازی مدل و نما کاهش پیچیدگی طراحی الگوریتم و افزایش انعطاف پذیری و نگهداشت پذیری کد مبدأ است. همچنین اموی سی برای ساده سازی طراحی سیستم های خودمختار و خودمدیریتی استفاده می شود.

## :ASP .NET MVC

Asp.net MVC بخشی از چهار چوب Asp .NET است که توسط Microsoft برای پیاده سازی نرم افزارهای مبتنی بر وب با الگوی MVC طراحی شده است. Asp.net MVC اولین و تنها انتخاب برای پیاده سازی الگوی MVC تحت چهارچوب .net نیست ، به عنوان مثال می توان به Ruby on Rails و monorail اشاره کرد. با استفاده از Asp .net MVC تمام مزایای الگوی MVC2 را خواهید داشت و از آنجایی که ASP .Net MVC بخشی از ASP .NET است لذا تمام قابلیت های Asp.net مانند caching ، security ، session state و دیگر امکانات وابسته را نیز در اختیار خواهیم داشت.

پایه های:ASP .net MVC

Routing:

موتور Routing یکی از قسمت های ASP .NET MVC نیست و در واقع بخشی از چهارچوب . net 3.5 sp1 می باشد و اولین قسمتی است که توسط درخواست های ارسالی از جانب پویشگر هدف قرار می گیرد و هدف آن استخراج اطلاعات مورد نیاز از URL و هدایت تمام درخواستهای واصله به

handler صحیح (که در اینحالت MvcHandler است) می باشد. برای برآورده کردن این مهم موتور Routing را باید بگونه ای تنظیم کرد که بداند چگونه باید اطلاعات را از URL استخراج کند. این تنظیمات در متد RegisterRoutes در فایل Global.asax انجام می شود.

#### Model:

ASP.NET MVC قرارداد خاصی برای Model وضع نمی کند. در نگاه اول Model تنها پوشه ای است که اصولاً تمام کلاس های مورد نیاز برای برقراری ارتباط با لایه منطقی و یا لایه خدمات را در آن قرار می دهیم. این مفهوم می تواند مطابق میل شما شکل گیرد ، می تواند مدل تولید شده توسط Nhibernate ، Entity Framework ، LINQ 2 SQL و یا هرچیز دیگری که اطلاعات مورد نیاز برای ارایه View را در اختیار قرار دهد باشد.

#### Controller:

یک Controller درواقع کلاسی است که از کلاس پایه Controllers در ASP.NET MVC ارث بری شده است و نام آن همواره باید به واژه Controller ختم شود و اصولاً در پوشه ای که به همین نام منظور شده است قرار گیرد. فعالیت اصلی در واقع توسط متدهای درورنی این کلاس انجام می شود که آنها را Action Method می خوانیم و بکمک موتور Routing هر URL به یکی از این متد ها اشاره دارد.

#### : ActionMethod

همانطور که گفته شد یک Action Method چیزی نیست مگر متدی Public درون یک کلاس Controller که معمولاً خروجی ای از نوع ActionResult دارد و تعدادی پارامتر دلخواه که در درخواست HTTP ذکر شده اند را دریافت می کند. علاوه بر این جدول ViewData نیز که بمنظور ذخیره متغیرهای مورد نیاز View می باشد نیز از طریق شیء ActionResult به View ارسال می شود.

#### View:

در MVC این آیتم (View) باید تا حد امکان گنگ ، محقر و منفعل باشد. View تنها باید به ترسیم

واسطی برای نمایش بکاربر اهمیت دهد. بطور ایده آل هرچیزی فرای ترسیم گرافیکی باید از View خارج شده و در Model یا Controller قرار گیرد. در ASP.NET MVC در حالت پیش فرض View یک فایل aspx می باشد بدون (code-behind) که شامل HTML و دستورات server side درون خط مانند asp کلاسیک می باشد که اطلاعاتی را که از طریق ViewData به آن ارسال شده است را پردازش می کند.

### :HTMLHelper

شما ممکن است علاقه ای به نوشتن کدهای HTML نداشته باشید. برای حل این مشکل و همچنین بمنظور اتصال اطلاعات ارسالی از Controller ، چهارچوب ASP.NET MVC شامل متدهایی است که تحت عنوان HtmlHelpers شناخته می شوند.

### منابع:

- *A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System (Glenn E. Krasner and Stephen T. Pope ParcPlace Systems, Inc.)*
- [msdn.microsoft.com](http://msdn.microsoft.com)