



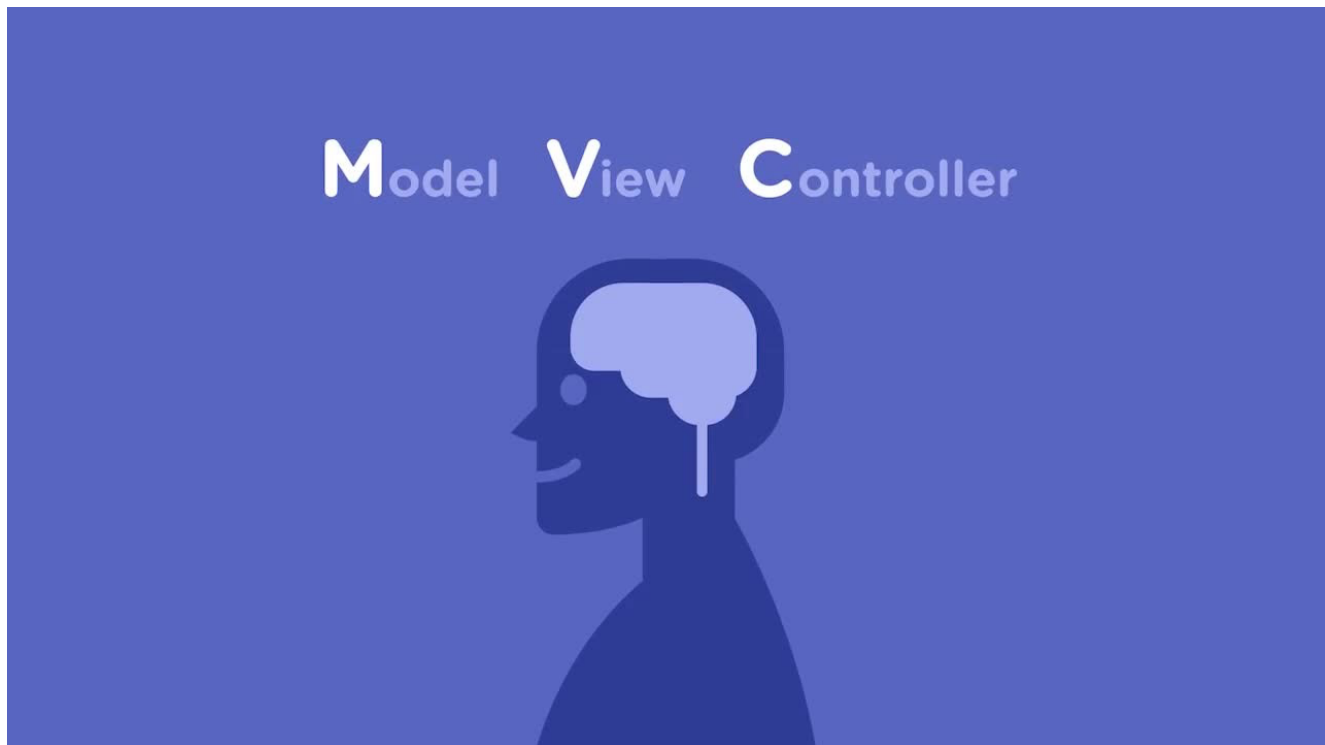
عنوان مقاله: هرآنچه باید در مورد الگوی MVC بدانید

نویسنده مقاله: تیم فنی نیک‌آموز

تاریخ انتشار: تیرماه ۱۴۰۱

منبع: <https://nikamooz.com/mvc-pattern/>

شاید شما هم جزء افرادی باشید که به برنامه‌نویسی و توسعه نرم‌افزارهای مختلف تحت وب علاقه‌مند هستید. از این‌رو، نیاز است تا با الگوها و معماری‌های مختلف آشنا شده و نحوه کار با آن‌ها را یاد بگیرید. یکی از این الگوها که محبوبیت بسیار زیادی در بین توسعه‌دهندگان پیدا کرده است، الگوی MVC است. یکی از الگوهای طراحی و توسعه برنامه‌های کاربردی وب، وب‌سایت‌ها و صفحات مختلف وب است. این الگو دارای مزایای زیادی بوده که استفاده از آن را در بین برنامه‌نویسان رونق داده است.

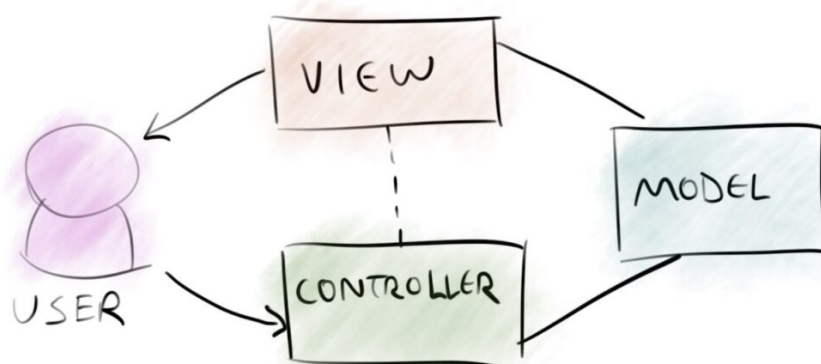


MVC یک الگوی طراحی و توسعه برنامه‌های کاربردی سمت وب است

## الگوی MVC چیست؟

Model View Controller یا MVC به عنوان یک الگو یا روش طراحی شناخته می‌شود. توسعه‌دهندگان یا طراحان از این الگو برای توسعه مؤثر برنامه‌های کاربردی تحت وب استفاده می‌کنند. باید بدانید که این الگو در مقایسه با الگوهای سنتی‌تری مانند ASP.NET ساده‌تر است و همین امر موجب می‌شود که یادگیری آن نیز راحت‌تر باشد.

همچنین اجزای معماری الگوی MVC برای رسیدگی به جنبه‌های مختلف یک برنامه کاربردی در حال توسعه، طراحی شده‌اند. الگوی طراحی MVC در خدمت جداسازی لایه ارائه یا نمایش از منطق تجاری خواهد بود. شاید برای شما هم این سؤال پیش آمده باشد که چرا توسعه‌دهندگان تا این اندازه به معماری MVC اهمیت می‌دهند؟ در پاسخ باید گفت که استفاده از MVC در توسعه برنامه و وب می‌تواند باعث ساده‌تر شدن فرایندها و مراحل مختلف آن شود.



معماری MVC به سه بخش مختلف تقسیم می‌شود

## آشنایی با معماری MVC

تا اینجای مقاله متوجه شدیم که الگوی محبوب MVC یک الگو توسعه اپلیکیشن است که می‌تواند مزایای زیادی را با خود به همراه داشته باشد؛ اما سؤال این است که این مزایا چگونه به دست می‌آیند؟ باید بدانید که MVC می‌تواند الگوی طراحی نرم‌افزار را به سه روش زیر تقسیم‌بندی کند:

مدل در MVC شکل داده‌ها را نشان می‌دهد. همچنین می‌تواند داده‌ها و منطق مربوط به آن‌ها را ذخیره‌سازی کند. از سوی دیگر، مدل مسئول پاسخگویی به درخواست کنترلر است و این قابلیت را دارد تا خودش را به‌روزرسانی کند.

### چشم‌انداز (View)

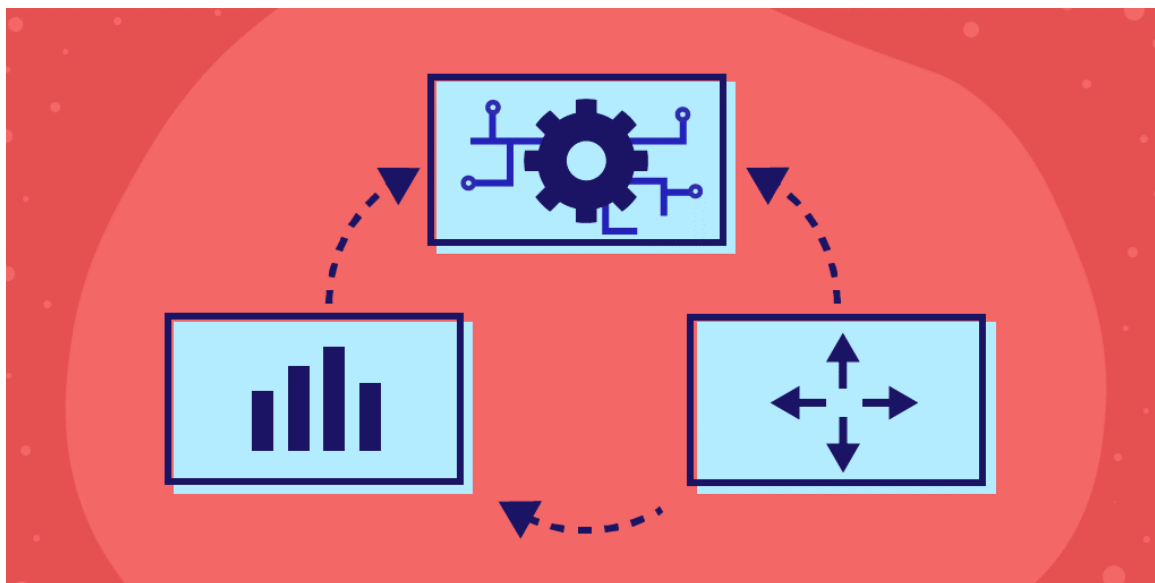
چشم‌انداز یا View در الگوی MVC یک رابط کاربری است که داده‌های مدل را به کاربر نشان خواهد داد. همچنین چشم‌انداز قادر خواهد بود تا به کاربر اجازه دهد داده‌های نمایش داده شده را تغییر دهد. چشم‌انداز می‌تواند داده‌ها و اطلاعات موجود را در قالب نمودارها و جداول به کاربران نشان دهد.

### کنترل‌کننده (Controller)

کنترل‌کننده یا کنترل بخشی از برنامه است که به درخواست کاربر و تعامل او رسیدگی خواهد کرد. کنترلر دستوری را به مدل ارسال می‌کند تا خودش را به‌روزرسانی کند. همچنین کنترل‌کننده در معماری MVC درخواست‌های HTTP مطرح شده توسط کاربر و فرایند مشاهده را مدیریت می‌کند.

تقسیم‌بندی منطق معماری طراحی MVC می‌تواند مزایای زیادی داشته باشد. از مهم‌ترین مزایای آن می‌توان به این نکته اشاره کرد که دیگر مدل، چشم‌انداز و کنترل‌کننده به یکدیگر وابسته نیستند. به همین دلیل، زمانی که یک تیم متشکل از طراح، معمار پایگاه داده یا مهندس برای توسعه نرم‌افزار دور یکدیگر جمع می‌شوند، هرکدام از تیم‌ها می‌توانند به صورت هم‌زمان روی قطعه مورد نظر خود کار کنند.

از سوی دیگر، این تقسیم‌بندی برای تعمیر و نگهداری نرم‌افزار نیز بسیار سودمند خواهد بود. هرکدام از قطعات مدل، چشم‌انداز، کنترل‌کننده قابلیت این را دارند تا کاملاً مستقل از یکدیگر کار کنند. در این حالت، توسعه‌دهندگان می‌توانند یک اشکال یا عیب را در یک قطعه کد و بدون نیاز به بررسی سایر کدها به صورت کامل بررسی کنند.

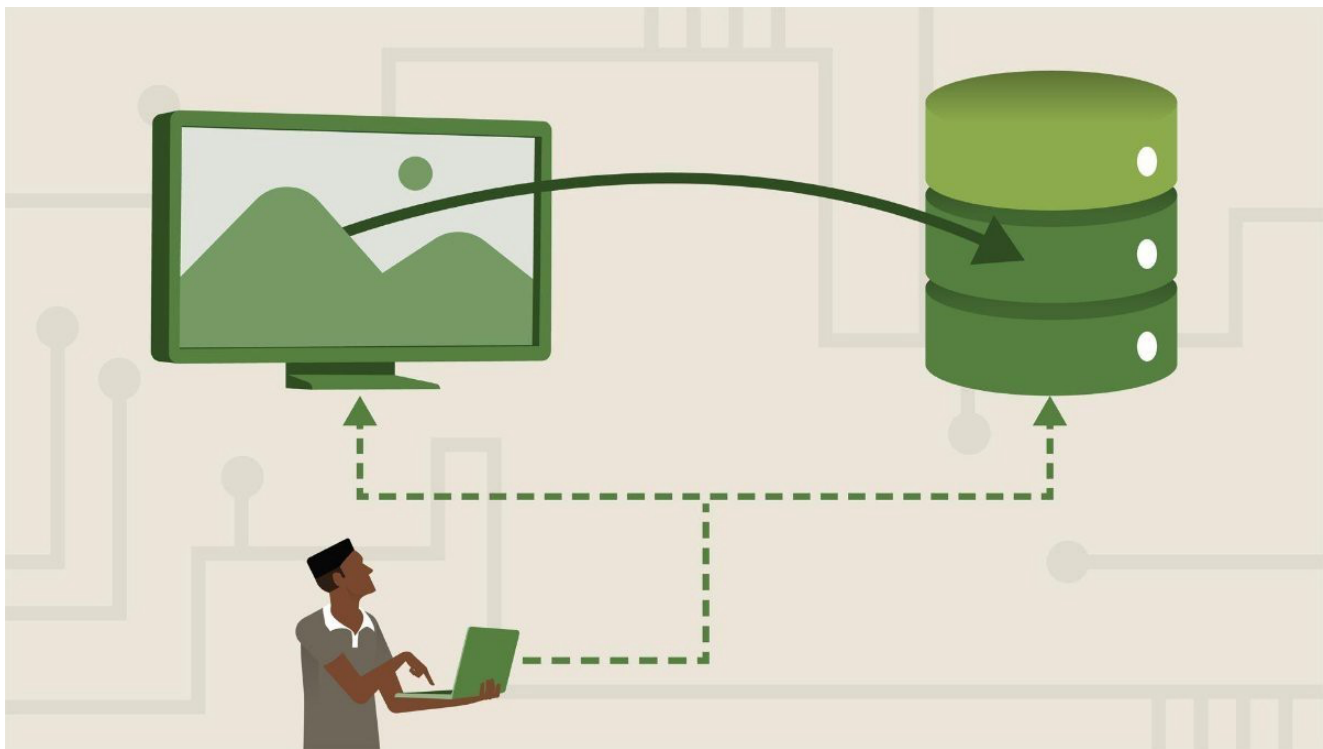


عملکرد معماری MVC بسیار ساده و پرسرعت است

## نحوه عملکرد فرایند معماری MVC

در قسمت قبل اشاره کردیم که معماری الگوی MVC اپلیکیشن را به سه قسمت مدل، چشم‌انداز و کنترل‌کننده تقسیم می‌کند. در این قسمت قصد داریم ببینیم که این سه قسمت چگونه با یکدیگر فعالیت دارند و اصولاً نحوه عملکرد معماری طراحی MVC چگونه خواهد بود.

- ابتدا کاربر از طریق چشم‌انداز درخواست خود را انجام می‌دهد.
- درخواست از چشم‌انداز به سمت کنترلر ارسال می‌شود و کنترلر پیکربندی مسیر را بررسی خواهد کرد.
- پس از بررسی پیکربندی، درخواست به مدل منتقل می‌شود و مدل آن را مورد بررسی قرار می‌دهد.
- در این بخش، مدل دو مرحله را پیش روی خود خواهد داشت و طی آن مقداردهی اولیه صفحه شروع می‌شود و نتایج ایجاد خواهند شد.
- زمانی که نتیجه مورد نظر تولید شد، برای مشاهده از طریق موتور نمایش بارگیری می‌شود.



استفاده از الگوی MVC مزایای زیادی خواهد داشت

## مزایای الگوی MVC

مزایای بی‌شمار MVC موجب شده است تا استفاده از این الگو و طراحی در بین برنامه‌نویسان، طراحان و توسعه‌دهندگان محبوبیت زیادی داشته باشد و به این اندازه رونق پیدا کند. در این قسمت به برخی از مهم‌ترین مزایای MVC اشاره خواهیم کرد:

- MVC به طور سنتی برای رابط‌های کاربردی گرافیکی (GUI) استفاده می‌شود.
- این الگو در بین برنامه‌های کاربردی محبوبیت زیادی دارد.
- مسئولیت‌های MVC بین مشتری و سرور تقسیم شده است که با معماری برنامه‌های وب سازگار خواهد بود.
- MVC یک الگوی طراحی مفید در هنگام برنامه‌ریزی توسعه است.
- تقسیم عملکرد به مدل، چشم‌انداز و کنترلر کننده باعث از بین رفتن نگرانی‌های اجرایی خواهد شد.
- MVC می‌تواند به خوبی با Ruby on Rails کار کند.
- MVC قادر است تا وابستگی‌های غیرضروری را حذف کند.
- MVC کلاس‌های مدل را بدون تغییر، قابل استفاده مجدد می‌کند.
- امکان استفاده از کد قابل توسعه در این الگو وجود دارد.
- انسجام این معماری بسیار مناسب است.
- نگهداری یا تغییر MVC بسیار ساده است.
- امکان پشتیبانی از نمایش‌های متعدد وجود دارد.
- MVC می‌تواند هر یک از قسمت‌های مدل، چشم‌انداز و کنترلر را به صورت مستقل مورد آزمایش قرار دهد.

## معایب الگوی MVC

در کنار تمامی مزایایی که برای MVC برشمردیم، باید بدانید که این الگو خالی از عیب نیست و ممکن است به همراه خود معایبی را به وجود بیاورد. برای بررسی بی‌طرفانه و شناخت دقیق MVC نیاز است تا با معایب این الگو نیز آشنایی پیدا کنید. از بارزترین این معایب می‌توان به موارد زیر اشاره کرد:

- MVC پشتیبانی اعتبارسنجی واضحی را برای برنامه‌ها ارائه نمی‌کند.
  - MVC برای همه برنامه‌ها مناسب نیست و نمی‌تواند از برنامه‌های کوچک پشتیبانی کند.
  - پیچیدگی برای توسعه برنامه‌ها با استفاده از MVC زیاد است.
  - MVC به توسعه‌دهندگان بیشتری نیاز دارد تا برنامه‌نویسی موازی را برای برنامه‌ها انجام دهند.
  - با استفاده از رابط‌های کاربری، توسعه برنامه‌های کاربردی برای MVC دشوار است.
- با وجود چنین معایبی در طراحی MVC، همچنان می‌توان این الگو را به‌عنوان یکی از برترین و محبوب‌ترین الگوهای توسعه نرم‌افزارهای مختلف برشمرد.

## نتیجه‌گیری

الگوی MVC را می‌توان یک الگو بسیار کاربردی برای توسعه و طراحی نرم‌افزارها و صفحات تحت وب دانست. MVC می‌تواند از وبسایت‌های واکنش‌گرا با کمک بوت‌استرپ پشتیبانی کند و مزایای زیادی را با خود به همراه داشته باشد. از آنجایی که معماری MVC به صورت بخش‌بندی شده به سه قسمت مدل، چشم‌انداز و کنترل‌کننده تقسیم شده است، نگهداری، تعمیر یا رفع نواقص برنامه بسیار ساده‌تر می‌شود و هزینه‌های تعمیر و توسعه نیز کاهش پیدا می‌کند.