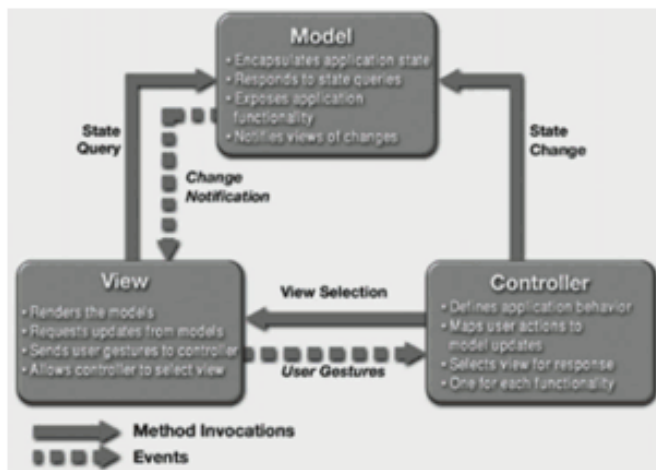
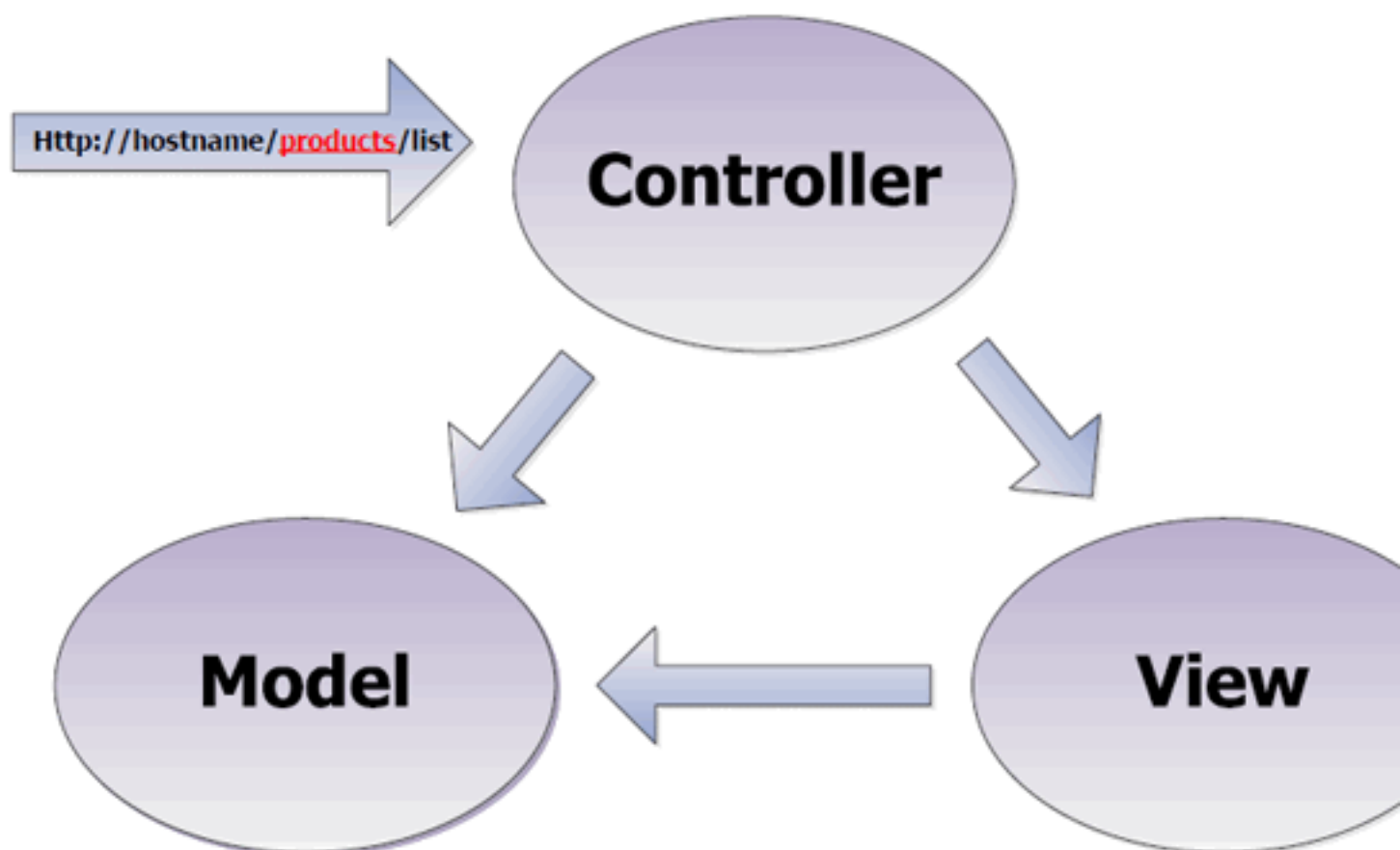


MVC چیست و اساس کار آن چگونه است؟

الگوی MVC در سال‌های اول دهه 70 میلادی در شرکت زیراکس توسط خالقین زبان اسمال‌تاک که جزو اولین زبان‌های شیء‌گرا محسوب می‌شود، ارائه گردید. نام MVC از الگوی Model-View-Controller گرفته شده و چندین دهه است که در صنعت تولید نرم افزار مورد استفاده می‌باشد. هدف اصلی آن جدا سازی مسئولیت‌های اجزای تشکیل دهنده «لایه نمایشی» برنامه است. این الگو در سال 2004 برای اولین بار در سکویی به نام Rails به کمک زبان روبی جهت ساخت یک فریم ورک وب MVC مورد استفاده قرار گرفت و پس از آن به سایر سکوها مانند جاوا، دات نت (در سال 2007)، PHP و غیره راه یافت. تصاویری را از این تاریخچه در ادامه ملاحظه می‌کنید؛ از دکتر Trygve Reenskaug تا شرکت زیراکس و معرفی آن در Rails به عنوان اولین فریم ورک وب MVC.



C در MVC معادل Controller است. کنترلر قسمتی است که کار دریافت ورودی‌های دنیای خارج را به عهده دارد؛ مانند پردازش یک درخواست HTTP ورودی.

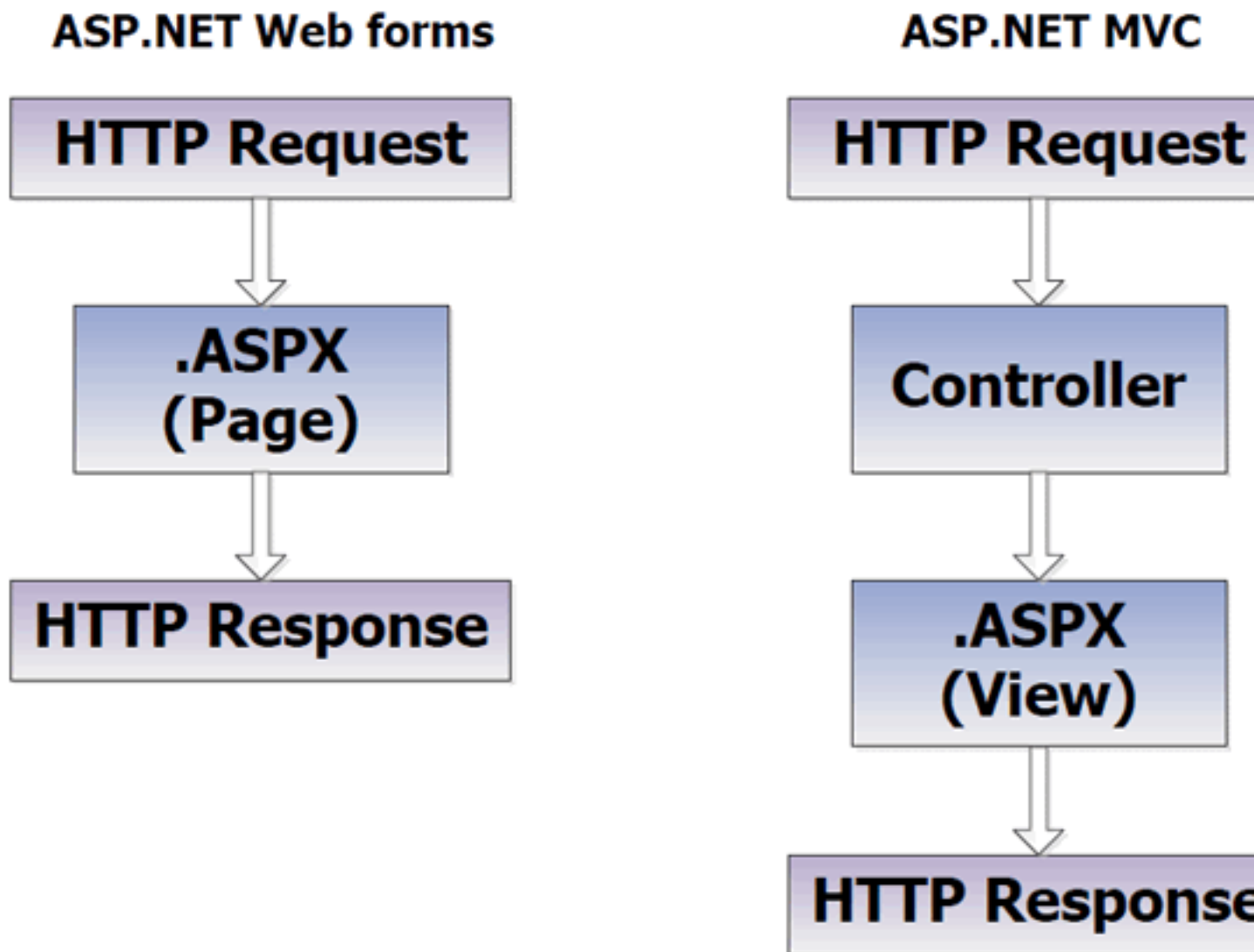


زمانیکه کنترلر این درخواست را دریافت می‌کند، کار وهله سازی Model را عهده دار خواهد شد و حاوی اطلاعاتی است که نهایتاً در اختیار کاربر قرار خواهد گرفت تا فرآیند پردازش درخواست رسیده را تکمیل نماید. برای مثال اگر کاربری جهت دریافت آخرین اخبار به سایت شما مراجعه کرده است، در اینجا کار تهیه لیست اخبار بر اساس مدل مرتبط به آن صورت خواهد گرفت. بنابراین کنترلرها، پایه اصلی و مدیر ارکستر الگوی MVC محسوب می‌شوند.

در ادامه، کنترلر یک View را جهت نمایش Model انتخاب خواهد کرد. View در الگوی MVC یک شیء ساده است. به آن می‌توان به شکل یک قالب که اطلاعاتی را از Model دریافت نموده و سپس آن‌ها را در مکان‌های مناسبی در صفحه قرار می‌دهد، نگاه کرد. نتیجه استفاده از این الگو، ایزوله سازی سه جزء یاد شده از یکدیگر است. برای مثال View نمی‌داند و نیازی ندارد که بداند چگونه باید از لایه دسترسی به اطلاعات کوئری بگیرد. یا برای مثال کنترلر نیازی ندارد بداند که چگونه و در کجا باید خطایی را با رنگی مشخص نمایش دهد. به این ترتیب انجام تغییرات در لایه رابط کاربری برنامه در طول توسعه کلی سیستم، ساده‌تر خواهد شد. همچنین در اینجا باید اشاره کرد که این الگو مشخص نمی‌کند که از چه نوع فناوری دسترسی به اطلاعاتی باید استفاده شود. می‌توان از بانک‌های اطلاعاتی، وب سرویس‌ها، صف‌ها و یا هر نوع دیگری از اطلاعات استفاده کرد. به علاوه در اینجا در مورد نحوه طراحی Model نیز قیدی قرار داده نشده است. این الگو تنها جهت ساخت بهتر و اصولی «رابط کاربری» طراحی شده است و بس.

تفاوت مهم پردازشی ASP.NET MVC با ASP.NET Web forms

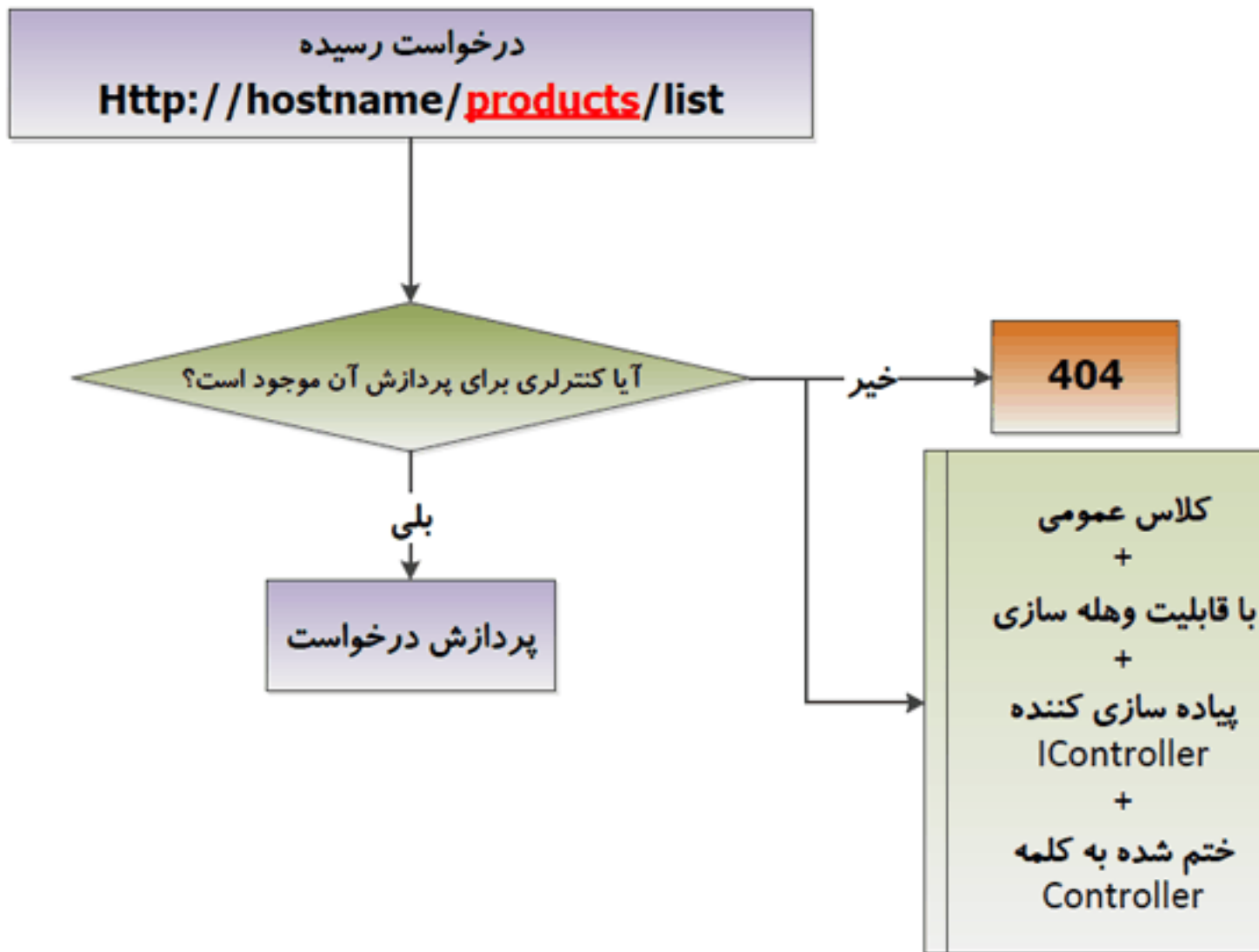
اگر پیشتر با ASP.NET Web forms کار کرده باشید اکنون شاید این سؤال برایتان وجود داشته باشد که این سیستم جدید در مقایسه با نمونه قبلی، چگونه درخواست‌ها را پردازش می‌کند.



همانطور که مشاهده می‌کنید، در وب فرم‌ها زمانیکه درخواستی دریافت می‌شود، این درخواست به یک فایل موجود در سیستم مثلاً `default.aspx` ارسال می‌گردد. سپس ASP.NET یک کلاس وهله سازی شده معرف آن صفحه را ایجاد کرده و آن را اجرا می‌کند. در اینجا چرخه طول عمر صفحه مانند `page_load` و غیره رخ خواهد داد. جهت انجام این وهله سازی، View به فایل `code behind` خود گره خورده است و جدا سازی خاصی بین این دو وجود ندارد. منطق صفحه به `markup` آن که معادل است با یک فایل فیزیکی بر روی سیستم، کاملاً مقید است. در ادامه، این پردازش صورت گرفته و HTML نهایی تولیدی به مرورگر کاربر ارسال خواهد شد. در ASP.NET MVC این نحوه پردازش تغییر کرده است. در اینجا ابتدا درخواست رسیده به یک کنترلر هدایت می‌شود و این کنترلر چیزی نیست جز یک کلاس مجزا و مستقل از هر نوع فایل ASPX ایی در سیستم. سپس این کنترلر کار پردازش درخواست رسیده را شروع کرده، اطلاعات مورد نیاز را جمع آوری و سپس به View ایی که انتخاب می‌کند، جهت نمایش نهایی ارسال خواهد کرد. در اینجا View این اطلاعات را دریافت کرده و نهایتاً در اختیار کاربر قرار خواهد داد.

آشنایی با قرارداد یافتن کنترلرهای مرتبط

تا اینجا دریافته‌ایم که نحوه پردازش درخواست‌ها در ASP.NET MVC بر مبنای کلاس‌ها و متدها است و نه بر مبنای فایل‌های فیزیکی موجود در سیستم. اگر درخواستی به سیستم ارسال می‌شود، در ابتدا، این درخواست جهت پردازش، به یک متد عمومی موجود در یک کلاس کنترلر هدایت خواهد شد و نه به یک فایل فیزیکی ASPX (برخلاف وب فرم‌ها).



همانطور که در تصویر مشاهده می‌کنید، در ابتدای پردازش یک درخواست، آدرسی به سیستم ارسال خواهد شد. بر مبنای این آدرس، نام کنترلر که در اینجا زیر آن خط قرمز کشیده شده است، استخراج می‌گردد (برای مثال در اینجا نام این کنترلر `ProductsController` است). سپس فریم ورک به دنبال کلاس این کنترلر خواهد گشت. اگر آن را در اسمبلی پروژه بیابد، از آن خواهد خواست تا درخواست رسیده را پردازش کند؛ در غیراینصورت پیغام 404 یا یافت نشد، به کاربر نمایش داده می‌شود.

اما فریم ورک چگونه این کلاس کنترلر درخواستی را پیدا می‌کند؟ در زمان اجرا، اسمبلی اصلی پروژه به همراه تمام اسمبلی‌هایی که به آن ارجاعی دارند جهت یافتن کلاسی با این مشخصات اسکن خواهند شد:

- 1- این کلاس باید عمومی باشد.
- 2- این کلاس نباید `abstract` باشد (تا بتوان آن را به صورت خودکار و هله سازی کرد).
- 3- این کلاس باید اینترفیس استاندارد `IController` را پیاده سازی کرده باشد.
- 4- و نام آن باید مختوم به کلمه `Controller` باشد (همان بحث `Convention over configuration` یا کار کردن با یک سری قرار داد از پیش تعیین شده).

برای مثال در اینجا فریم ورک به دنبال کلاسی به نام `ProductsController` خواهد گشت.

شاید تعدادی از برنامه نویسی‌های ASP.NET MVC تصور کنند که فریم ورک در پوشه‌ی استاندارد به نام Controllers به دنبال این کلاس خواهد گشت؛ اما در عمل زمانیکه برنامه کامپایل می‌شود، پوشه‌ای در این اسمبلی وجود نخواهد داشت و همه چیز از طریق Reflection مدیریت خواهد شد.

نظرات خوانندگان

نویسنده: امین
تاریخ: ۱۳۹۱/۰۱/۰۵ ۰۱:۱۲:۲۰

با سپاس،
در مورد مدل، طبق فرمایشات شما می شه از DTO موجود در ساختار لایه ای برنامه استفاده کرد، درسته؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۰۵ ۰۸:۵۱:۳۶

بله. در کل هیچ قید خاصی نداره این قسمت.

نویسنده: علی پناهی
تاریخ: ۱۳۹۱/۰۲/۰۱ ۱۷:۲۳:۲۵

خیلی ممنون از آموزشها، اما اگه میشه کلمه "وهله سازی" را بکار نبرید یا یه توضیح یا معادل برای اون بگید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۱ ۱۷:۳۹:۲۳

instance = وهله
instantiation = وهله سازی

نویسنده: علی پناهی
تاریخ: ۱۳۹۱/۰۲/۰۱ ۲۱:۱۱:۲۵

خیلی ممنون. حالا متنتون رو بهتر می فهمم. اما چرا از "نمونه" و "نمونه سازی" استفاده نکردید ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۱ ۲۲:۲۲:۰۲

برای اینکه در متون فارسی شیءگرایی، «وهله» و «وهله سازی» بسیار مرسوم هستند.

نویسنده: علی اژدری
تاریخ: ۱۳۹۱/۰۵/۰۲ ۱۴:۷

با سلام ، با تشکر از آقای نصیری ، استفاده از متون تخصصی ترجمه شده واقعا خواندن مقاله رو سخت میکنه چون هیچ چیز استاندارد وجود نداره ، نمونه این مساله در کتابهای پایگاه داده به وضوح دیده میشه ، چرا باید گفت وهله؟ آیا لازمه به این پیچیدگی ها ی موجود چیز دیگه ای اضافه بشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۲ ۱۴:۱۹

«وهله» یک کلمه متداول هست در متون فارسی مرتبط و اگر با آن آشنایی ندارید مطالعه خودتون رو بیشتر کنید.

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۱/۰۵/۱۴ ۳:۴۱

نحوه ارتباط view با model چگونه است؟

طبق تعریف view اطلاعاتی که کنترلر از model دریافت کرده را، از کنترلر مربوط دریافت و در مکان‌های مناسبی قرار می‌دهد. پس فلش که در شکل ابتدایی نشان داده شده است چه ارتباطی را بین view و model نشان می‌دهد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۴ ۸:۴۸

برای توضیحات بیشتر مراجعه کنید به [قسمت پنجم](#). یک قسمت کامل به آن اختصاص داده شده.