

پیش از اینکه آموزش AngularJS را شروع کنیم بهتر است با مفهوم برنامه‌های تک صفحه ای وب و یا Single Page Web Applications آشنا شویم؛ چرا که AngularJS برای توسعه هر چه ساده‌تر و قوی‌تر این گونه برنامه‌ها متولد شده است.

Single Page Application

برای درک چگونگی کارکرد این برنامه ها، مثالی را میزنیم که هر روزه با آن سرو کار دارید، یکی از نمونه‌های کامل و قدرتمند برنامه‌های Single Page Application و یا به اختصار SPA، سرویس پست الکترونیکی Google و یا همان Gmail است. اجازه بدهید تا ویژگی‌های SPA را با بررسی Gmail انجام دهیم، تا به درک روشنی از آن برسید:

Reload نشدن صفحات

در Gmail هیچ گاه صفحه Reload و یا اصطلاحا بارگیری مجدد نمی‌شود. برای مثال، وقتی شما لیست ایمیل‌های خود را مشاهده میکنید و سپس بر روی یکی از آن‌ها کلیک میکنید، بدون اینکه به صفحه ای دیگر هدایت شوید؛ ایمیل مورد نظر را میبینید. در حقیقت تمامی اطلاعات در همان صفحه نمایش داده می‌شوند و بر عکس وب سایت‌های معمول است که از صفحه ای به صفحه ای دیگر هدایت میشوند، در یک صفحه تمام کارهای مورد نیاز خود را انجام می‌دهید و احتیاجی به بارگیری مجدد صفحات نیست. با توجه به این صحبت‌ها برای توسعه دهنده‌های وب آشکار است که تکنیک AJAX، نقشی اساسی در این گونه برنامه‌ها دارد، چون کله‌ی عملیات ارتباط با سرور در پشت زمینه انجام می‌شوند.

تغییر URL در نوار آدرس مرورگر

وقتی شما بر روی یک ایمیل کلیک می‌کنید و آن ایمیل را بدون Reload شدن مجدد صفحه مشاهده می‌کنید، آدرس صفحه در مرورگر نیز تغییر می‌کند. خب مزیت این ویژگی چیست؟ مزیت این ویژگی در این است که هر ایمیل شما دارای یک آدرس منحصر به فرد است و به شما امکان Bookmark کردن آن لینک، باز کردن آن در یک Tab جدید و یا حتی ارسال آن به دوستان خود را دارید. حتی اگر این مطلب را جدا از Gmail در نظر بگیریم، به موتورهای جست و جو کمک می‌کند، تا هر صفحه را جداگانه Index کنند؛ جدا از اینکه وبسایت ما SPA است. همچنین این کار یک مزیت مهم دیگر نیز دارد؛ و آن کار کردن کلیدهای back و forward مرورگر، برای بازگشت به صفحات پیمایش شده قبلی است.

شاید قبل از بیان این ویژگی با خود گفته باشید که پیاده سازی Reload نشدن صفحات با AJAX آن چنان کار پیچیده ای نیست. بله درست است، اما آیا شما قبل از این راه حلی برای تغییر URL اندیشیده بودید؟ مطمئنا شما هم صفحات وب زیادی را دیده اید که همه‌ی صفحات آن دارای یک URL در نوار آدرس مرورگر هستند و هیچگاه تغییر نمی‌کنند و با باز کردن یک لینک در یک Tab جدید، باز همان صفحه‌ی تکراری را مشاهده می‌کنند! و یا بدتر از همه که دکمه‌ی back مرورگر غیر عادی عمل می‌کند. بله، اینها تنها تعدادی از صدها مشکلات رایج سیستم‌های نوشته شده ای است که سعی کردند همه‌ی کارها در یک صفحه انجام شود.

Cache شدن اطلاعات دریافتی

شاید خیلی‌ها ویژگی‌های فوق را برای یک SPA کافی بدانند، اما تعدادی هم مانند نگارنده وجود یک کمبود را حس می‌کنند و آن کش شدن اطلاعات دریافتی در مرورگر است. Gmail این امکان را به خوبی پیاده سازی کرده است. لیست ایمیل‌های دریافتی در بار اول از سرور دریافت می‌شود، سپس شما بر روی یک ایمیل کلیک و آن را مشاهده می‌کنید. حال به لیست ایمیل‌های دریافتی بازگردید، آیا رفت و برگشتی به سرور انجام می‌شود؟ مسلما خیر. حتی اگر دوباره بر روی آن ایمیل مشاهده شده، کلیک کنید، بدون رفت و برگشتی به سرور آن ایمیل را مشاهده می‌کنید.

کش شدن اطلاعات سبب می‌شود که بار سرور خیلی کاهش یابد و رفت و آمدهای بیهوده صورت نگیرد. کش شدن داده‌ها یک مزیت دیگر نیز دارد و آن تبدیل برنامه‌های معمول وب stateless به برنامه‌های شبه دسکتاپ state full است. تکنیک AJAX در پیاده سازی امکانات فوق نقشی اساسی را بازی می‌کند. کمی به عقب برمیگردیم یعنی زمانی که AJAX برای اولین بار مطرح شد و هدف اصلی به وجود آمدن آن پیاده سازی برنامه‌های وب به شکل دسکتاپ بود و این کار از طریق انجام تمامی ارتباطات سرور با XMLHttpRequest امکان پذیر می‌شد. شاید آن زمان با توجه به محدودیت تکنولوژی‌ها موجود این کار به صورت

تمام و کمال امکان پذیر نبود، اما امروزه به بهترین شکل ممکن قابل پیاده سازی است.

شاید اکنون این سوال پیش بیاید که چرا باید وبسایت خود را به شکل SPA طراحی کنیم؟

برای پاسخ دادن به این سوال باید گفت که سیستم‌های وب امروزی به دو دسته‌ی زیر تقسیم می‌شوند:

- Web Documents و یا همان وب سایت‌های معمول

- Web Applications و یا همان Single Page Web Applications

اگر هدف شما طراحی یک وب سایت معمول است که هدف آن، نمایش یک سری اطلاعات است و به قولی دارای محتواست، مطمئناً پیاده سازی این سیستم به صورت SPA کاری بیهوده به نظر می‌آید؛ ولی اگر هدفتان نوشتن سیستم‌هایی مثل Gmail، Google Maps، Azure، Facebook و ... است، پیاده سازی آن‌ها به صورت وب سایت‌های معمولی، غیر معقول به نظر می‌آید. حتی بخش‌های مدیریتی یک وبسایت هم می‌تواند به خوبی توسط SPA پیاده سازی شود، چرا که واقعا برای مدیریت اطلاعات یک وب سایت احتیاجی نیست، که از این صفحه به آن صفحه جا به جا شد.

معرفی کتابخانه‌ی AngularJS

AngularJS فریم ورکی متن باز و نوشته شده به زبان جاوا اسکریپت است. هدف از به وجود آمدن این فریم ورک، توسعه هر چه ساده‌تر SPAها با الگوی طراحی MVC و تست پذیری هر چه آسان‌تر آن‌ها است. این فریم ورک توسط یکی از محققان Google در سال 2009 به وجود آمد. بعدها این فریم ورک تحت مجوز MIT به صورت متن باز در آمد و اکنون گوگل آن را حمایت می‌کند و توسط هزاران توسعه دهنده در سرتاسر دنیا، توسعه داده می‌شود.

قبل از اینکه به بررسی ویژگی‌های Angular بپردازم، بهتر است ابتدا مطلبی درباره‌ی به کارگیری Angular از Brad Green که کارمند گوگل است، بیان کنم.

در سال 2009 تیمی در گوگل مشغول انجام پروژه ای به نام Google Feedback بودند. آن‌ها سعی داشتند تا در طی چند ماه، به سرعت کدهای خوب و تست پذیر بنویسند. پس از 6 ماه کدنویسی، نتیجه‌ی کار 17000 خط کد شد. در آن موقع یکی از اعضای تیم به نام Misko Hevery، ادعا کرد که می‌تواند کل این پروژه را در دو هفته به کمک کتابخانه‌ی متن بازی که در اوقات فراغت توسعه داده است، بازنویسی کند. Misko نتوانست در دو هفته این کار را انجام دهد. اما پس از سه هفته همه‌ی اعضای تیم را شگفت زده کرد. نتیجه‌ی کار تنها 1500 خط بود! همین باعث شد که ما بفهمیم که، Misko بر روی چیزی کاری میکند که ارزش دنبال کردن دارد.

پس از آن قضیه Misko و Brad بر روی Angular کار کردند و اکنون هم Angular توسط تیمی در گوگل و هزاران توسعه دهنده‌ی متن باز حرفه ای در سرتاسر جهان، در حال توسعه است.

فکر کنم همین داستان ذکر شده، قدرت فوق العاده زیاد این فریم ورک را برای همگان آشکار سازد.

ویژگی‌های AngularJS:

- قالب‌های سمت کاربر (Client Side Templates): انگولار دارای یک template engine قدرتمند برای تعریف قالب است.

- پیروی از الگوی طراحی MVC: انگولار، الگوی طراحی MVC را برای توسعه پیشنهاد می‌دهد و امکانات زیادی برای توسعه هر چه راحت‌تر با این الگو فراهم کرده است.

- Data Binding: امکان تعریف انقیاد داده دوطرفه (Two-Way Data Binding) در این فریم ورک به راحتی هرچه تمام، امکان پذیر است.

- Dependency Injection: این فریم ورک برای دریافت وابستگی‌های تعریف شده، دارای یک سیستم تزریق وابستگی توکار است.

- تعریف Service‌های سفارشی: در این فریم ورک امکان تعریف سرویس‌های دلخواه به صورت ماژول وجود دارد. این ماژول‌های مجزا را به کمک سیستم تزریق وابستگی توکار Angular، به راحتی در هر جای برنامه می‌توان تزریق کرد.

- تعریف Directive‌های سفارشی: یکی از جذاب‌ترین و قدرتمندترین امکانات این فریم ورک، تعریف Directive‌های سفارشی

است. Directive ها، امکان توسعه HTML را فراهم کرده اند. توسعه‌ی HTML اکنون در قالب Web Components ها فراهم شده است، اما هنوز هم خیلی از مرورگرهای جدید نیز از آن پشتیبانی نمی‌کنند.

- **فرمت کردن اطلاعات با استفاده از فیلترهای سفارشی:** با استفاده از فیلترها می‌توانید چگونگی الحاق شدن اطلاعات را برای نمایش به کاربر تعیین کنید ؛ انگولار همراه با فیلترهای گوناگون مختلفی عرضه میشود که میتوان برایه مثال به فیلتر ، currency ، date ،uppercase کردن رشته‌ها و اشاره کرد همچنین شما محدود به فیلترهای تعریف شده در انگولار نیستید و آزادید که فیلترهای سفارشی خودتان را نیز تعریف کنید.

- **سیستم Routing:** دارا بودن سیستم Routing قدرتمند، توسعه SPA ها را بسیار ساده کرده است.

- **سیستم اعتبار سنجی:** Angular دارای سیستم اعتبار سنجی توکار قدرتمند برای بررسی داده‌های ورودی است.

- **سرویس تو کار برای ارتباط با سرور:** Angular دارای سرویس پیش فرض ارتباط با سرور به صورت AJAX است.

- **تست پذیری:** Angular دارای بستری آماده برای تست کردن برنامه‌های نوشته شده است و از Unit Tests و Integrated End-to-End Test هم پشتیبانی می‌کند.

- **جامعه‌ی متن باز بسیار قوی**

این‌ها فقط یک مرور کلی بر توانایی‌های این فریم ورک بود و در ادامه هر کدام از این ویژگی را به صورت دقیق بررسی خواهیم کرد.

در مقاله‌ی بعدی، به چگونگی نصب AngularJS خواهیم پرداخت. سپس، اولین کد خود را با استفاده از آن خواهیم نوشت و مطالب Client Side Templates و MVC را دقیق‌تر بررسی خواهیم کرد.

نظرات خوانندگان

نویسنده:

دادخواه

تاریخ:

۱۹:۲۷ ۱۳۹۲/۰۶/۰۶

سلام

آیا استفاده از این فریمورک باعث این می‌شود که دیگر نیازی به JQuery نباشد؟ و یا باز کمبود هایی دارد که به استفاده از JQuery نیاز باشد؟
تشکر

نویسنده:

سالار

تاریخ:

۲۳:۱۹ ۱۳۹۲/۰۶/۰۶

به شدت دنبال آموزش angularjs بودم. سپاس و چند سوال.
- آیا با استفاده از angularjs برای یک SPA دیگر نیازی به asp.net mvc خواهد بود؟
- کامپایلر و ویژوال استودیو به خوبی برای razor کار میکند و بسیار intellisense قوی دارد و امکانات Strongly typed نیز دارد.
آیا این امکانات برای angular نیز موجود است تا خطایابی راحت‌تر شود؟
- من یک اپلیکیشن دارم که کاربر هر بار یکی از آیتم‌های منو را انتخاب میکند و برای آن یک ویو بصورت partialview درون یک تب لود میشود. هر ویو، فایل جاوا اسکریپت مخصوص به خود دارد. کاربر می‌تواند چندین ویوی مجزا را درون تب‌ها باز کند و بین آنها جابه‌جا شود. اما مدیریت دستی آنها بسیار سخت شده است. آیا انگولار امکان نمایش چند ویو را بطور همزمان و جابه‌جا شدن بین آنها را میدهد. (بدون بروز تداخل بین فایل‌های جاوا اسکریپت مربوط به هر ویو).
با تشکر.

نویسنده:

مهدی سعیدی فر

تاریخ:

۸:۰۰ ۱۳۹۲/۰۶/۰۷

سلام؛

کتابخانه‌ی JQuery عموماً برای دو کار استفاده می‌شود:

1- دستکاری یا پردازش DOM

2- ارتباط با سرور به صورت AJAX

انگولار سرویسی برای ارتباط AJAX دارد که شما را کاملاً از JQuery بی‌نیاز می‌کند.
برای پردازش DOM و الصاق رویداد به عناصر صفحه، انگولار دارای API‌های خیلی خوبی است که می‌تواند تا حدی جایگزین JQuery باشد. اما فراموش نکنیم که JQuery علاوه بر پردازش DOM دارای کلی API متنوع است که انگولار به پای آن نمی‌رسد. دلیل هم مشخص است؛ چرا که هر دو کتابخانه برای کاری متفاوت نوشته شده‌اند و مقایسه آن‌ها اصلاً کار درستی نیست. همچنین به این نکته هم توجه داشته باشید که برای JQuery هزاران افزونه‌ی مختلف نوشته‌اند که اجرای آن‌ها، وابسته به JQuery است. پس هیچ کتابخانه‌ای در این زمینه موازی کاری نمی‌کند، چون توانایی رقابت ندارد. (البته تعدادی را هم با کاملاً انگولار بازنویسی کرده‌اند). انگولار در مورد دستکاری DOM هم یک مورد را شما گوشزد می‌کند: اگر می‌خواهید DOM را دستکاری کنید، این کار را باید از طریق Directive‌ها انجام دهید. منظور این است کدهای مورد نظرتان را با هر کتابخانه‌ای که می‌خواهید بنویسید، ولی آن‌ها را درون یک Directive سفارشی بنویسید و روی هوا کد ننویسید تا قابل تست کردن باشد.
در ادامه کاملاً متوجه منظورم خواهید شد.

نویسنده:

مهدی سعیدی فر

تاریخ:

۸:۳۱ ۱۳۹۲/۰۶/۰۷

- AngularJS یک فریم ورک سمت کلاینتی به زبان جاوا اسکریپت است و ASP.NET MVC یک فریم ورک سمت سرور و این دو نمی‌توانند جای یکدیگر را پر کنند و ربطی به هم ندارند.

- در اینجا View‌های شما یک HTML ساده است و مفاهیم Strongly Typed View هیچ معنی خاصی ندارد. Model در انگولار یک

کلاس جاوا اسکریپتی ساده (POJO) است. view هم یک فایل HTML که قالب کار را مشخص می‌کند و Controller هم باز یک کلاس ساده‌ی جاوا اسکریپت است.

- اتفاقاً یک بار همچنین چیزی را از من خواستند (بدون انگولار) و من انجامش دادم. البته این طراحی اشتباه است. آیا شما ده بخش را در چند TAB مختلف باز می‌کنید، آدرس منحصر به فردی هم در نوار آدرس مرورگر برای آن‌ها در نظر گرفته اید؟ این کار در آن زمان که همه‌ی برنامه با یک URL کار می‌کرد خیلی عالی بود، ولی اکنون که شما می‌توانید برای View یک آدرس منحصر به فرد داشته باشید و آن‌ها را Tab خود مرورگر مدیریت کنید، احتیاجی به آن کارها نیست. به هر حال، در انگولار این جمله بی معنی است که: بین فایل‌های جاوا اسکریپت تداخل ایجاد شود! در انگولار هر View دارای Controller متناظر خود است. اعمال تجاری مشترک هم در سرویس‌ها تعریف می‌شوند و در کنترلرها تزیق می‌شوند. پس کد جاوا اسکریپتی روی هوا نوشته نمی‌شود (حتی یک خط)

صبر کنید تا کمی پیش برویم. این فقط مقدمه بود...

نویسنده: سالار
تاریخ: ۱۳۹۲/۰۶/۰۷ ۹:۵۲

سپاس فراوان.
آیا در یک وب اپلیکیشن و نه وب سایت عمومی، واقعاً نیازی به داشتن یک url مجزا برای هر ویو داریم. مگر هدف angular ایجاد برنامه‌هایی شبیه برنامه‌های دسکتاپ نیست. در برنامه‌های دسکتاپ که هر ویو آدرس مجزایی ندارد. با تشکر.

نویسنده: دادخواه
تاریخ: ۱۳۹۲/۰۶/۰۷ ۱۰:۲۱

سلام
تشکر از جواب خوبتان
من متوجه شدم که دانسته‌های شما نسبت به فریمورک‌های جاوا اسکریپت بسیار خوب است. پس اگر لطف کنید درباره چند تا از فریم ورک‌های دیگر بخصوص Backbone اطلاعاتی بدهید که هر کتابخانه در اصل برای چه کاری ساخته شده است. منظورم این است که به مقایسه بین فریم ورک‌های معروف که بیشتر برای چه کاری تهیه شده اند. اگر چنین مقایسه‌ای به صورت فارسی هست معرفی کنید.
تشکر

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۰۶/۰۷ ۱۰:۲۶

اغلب کارمندان یک سازمان یک عادت عمومی دارند، بر روی لینکشان کلیک راست می‌کنند و گزینه‌ی open in a new tab را می‌زنند و یا بدتر از کلیدهای back و forward مرورگر برای عقب جلو کردن صفحات پیمایش شده استفاده می‌کنند. حالا هر چقدر که توی گوششان بخوانید، که فلان لینک را باید روش فقط کلیک کنی، باز هم به خرجشان نمی‌رود. خیلی از کارمندان برای مثال لینک درج صورت حساب جدید را bookmark کرده اند تا به سرعت به آن دسترسی داشته باشند. همچنین autocomplete مرورگر هم در یافتن صفحات پیمایش شده به آن‌ها خیلی کمک می‌کند. یادمه همین مشکل را توی یک برنامه‌ی نوشته شده با سیلورلایت دیدم و مشتری قبول نمی‌کرد که نمی‌تواند در یک tab جدید صفحه باز کند و ...
برنامه‌های شبیه دسکتاپ یعنی اینکه رفرش شدن صفحات را برای تغییر view از بین ببرد. هوز هم ماهیت وب است و باید همیشه این را در نظر داشت، تا بهترین تجربه‌ی رابط کاربری را فراهم کرد. نباید گفت که برنامه‌ی من فلان جور طراحی شده؛ همینی هست که هست، باید با رابط کاربری عموم وب اینترفیس‌ها هماهنگ باشد.

نویسنده: _mehdi
تاریخ: ۱۳۹۲/۰۶/۲۱ ۱۲:۳۸

سلام

چطوری میشه به فایل‌های راهنما و آموزش‌های موجود در فایل دریافتی از angular js دسترسی داشت . (پسوند فایل‌ها ngdoc)

نویسنده: محسن خان

تاریخ: ۱۳:۵۶ ۱۳۹۲/۰۶/۲۱

These files are parsed by our docs parser (nodejs script), source can be found here:
<https://github.com/angular/angular.js/tree/master/docs/src>
 It is combined together with docs parsed from the source and result is html, served at
docs.angularjs.org

[ماخذ](#)

نویسنده: _mehdi

تاریخ: ۲۰:۳۵ ۱۳۹۲/۰۶/۲۱

سلام

ممنون بابت پاسختون ولی من متوجه نشدم که باید چطوری این فایل‌ها را به فایل‌های html تبدیل کرد . من باید nodejs را نصب کنم و با این فایل‌ها که در بالا ذکر کردید (البته کدومشون) و دادن مسیر پوشه راهنما فایل‌ها را به html تبدیل کنم لطفاً دقیق توضیح بدید اگه براتون امکان داره.

نویسنده: آریو

تاریخ: ۸:۳۸ ۱۳۹۲/۰۸/۱۵

سلام. با تشکر از مطالب بسیار خوبتون یه سوال داشتم.

من خیلی علاقه دارم روی الگوی MVVM کار کنم و تا اینجا به دو انتخاب رسیدم یکی KnockoutJS و دیگری AngularJS . سوالی که دارم اینه که با توجه به اینکه من روی ASP.NET MVC کار میکنم کدوم کتابخانه برای این فریمورک مناسبتره. درسته که این کتابخانه‌ها جاوااسکریپت هستند و تکنولوژی سمت سرور مطرح نیست. اما با توجه به اینکه میخوام روی MVVM وقت بزارم به نظرتون با در نظر گرفتن امکانات ، منابع آموزشی ، میزان استفاده در توسعه دهندگان ایرانی و در نهایت قدرت عملکرد روی کدوم کار کنم؟

آیا اینها دقیقاً یک کار اما با امکانات متفاوت انجام میدن یا کاربردهای متفاوتی دارند؟
 باتشکر

نویسنده: محسن خان

تاریخ: ۸:۵۲ ۱۳۹۲/۰۸/۱۵

[مقایسه 8 قسمتی AngularJS vs Knockout](#)

نویسنده: گل رز

تاریخ: ۱۰:۱۹ ۱۳۹۲/۰۸/۱۵

منابع آموزشی برای هر کدام وجود دارد . اگه قرار به یادگیری باشه از کجا و از چه منبعی باید شروع کنیم ؟ منبع خاصی مد نظرتون هست ؟

نویسنده: وحید نصیری

تاریخ: ۱۰:۲۴ ۱۳۹۲/۰۸/۱۵

- در سایت جاری: [AngularJS Knockout](#)

- در سایت‌های دیگر: [AngularJS Knockout SPA](#)