بسم الله الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

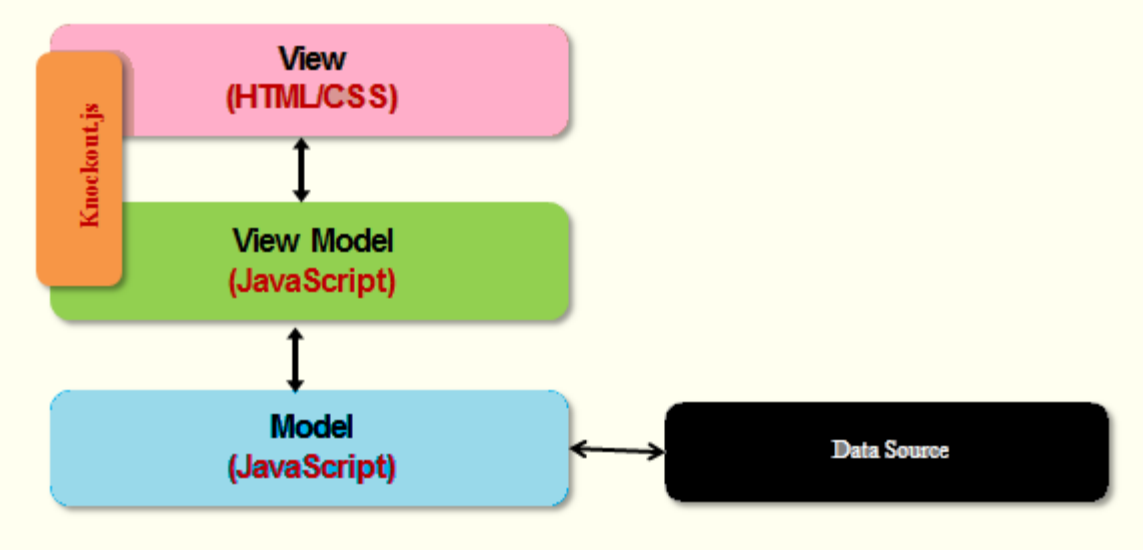
شروع کار با Knockout.js

مدرس : مهندس افشین رفوآ

شروع کار با Knockout.js

مقدمه

Knockout.js یا به اختصار KO، یک کتابخانه JavaScript بسیار قدرتمند است و استفاده از آن هر روز در حال افزایش است. از کتابخانه Knockout.js برای ساخت برنامه های تحت وب تعاملی و responsive استفاده می شود. Knockout.js توسط Steve Sanderson توسعه یافت. Steve Sanderson یک از کارمندان مایکروسافت است. هر وقت بخش هایی از UI داشته باشیم که باید به صورت خودکار بروزرسانی شوند، KO می تواند برای پیاده سازی راحت تر آن به ما کمک کند.



ویژگی های Knockout.js

Declarative bindingها

Declarative bindingها یک روش ساده و موثر برای اتصال بخش های مختلف UI به مدل داده است. می توانیم به راحتی یک UI پیچیده و پویا را به راحتی با استفاده از contextهای تودر تو به هم متصل (nested binding contexts)، بسازیم. با Declarative bindingها می توانیم DOM را تغییر دهیم و تمام بخش های به هم وابسته، متصل به هم باقی می مانند. در Declarative binding می توانیم با قراردادن ویژگی data-bind در عناصر DOM، داده ها را به DOM وصل کنیم. اصلی ترین مورد استفاده Knockout.js برای طراحی رابط کاربری (User Interface) مقیاس پذیر و داده محور است.

ردیابی وابستگی Elegant (Elegant dependency tracking)

زمانی که مدل داده تغییر میکند، Elegant dependency tracking باعث می شود، به صورت خودکار بخش مورد نظر در UI تغییر کند.

توسعه جزئی (Trivially extensible)

با استفاده از توسعه جزئی، برای اتصالات اعلانی جدید (declarative bindings)، رفتار دلخواه را انجام میدهیم، تا با چند خط کد Knockout.js به سادگی قابل استفاده مجدد باشد.

مزایای Knockout.js

Knockout.js بسیار پیچیده است. چون پس از فشرده سازی حدود 13kb خواهد بود.

مزیت دیگر آن این است که با هر مرورگری مانند Safari یا Chrome سازگاری دارد.

Knockout.js یک کتابخانه JavaScript محض است که قادر به کار کردن با هر تکنولوژی server-side و client-side ای می باشد.

دانلود و نصب Knockout.js

Knockout.js جاوا اسکریپت محض است و به هیچ کتابخانه دیگری وابسته نیست. بنابراین اگر بخواهید با Knockout.js کار کنید، باید آخرین نسخه از فایل Knockout JavaScript را دانلود کنید.

با استفاده از تگ <script> در هر جایی از صفحات HTML می توانید به فایل مورد نظر ارجاع یا Reference بدهید.

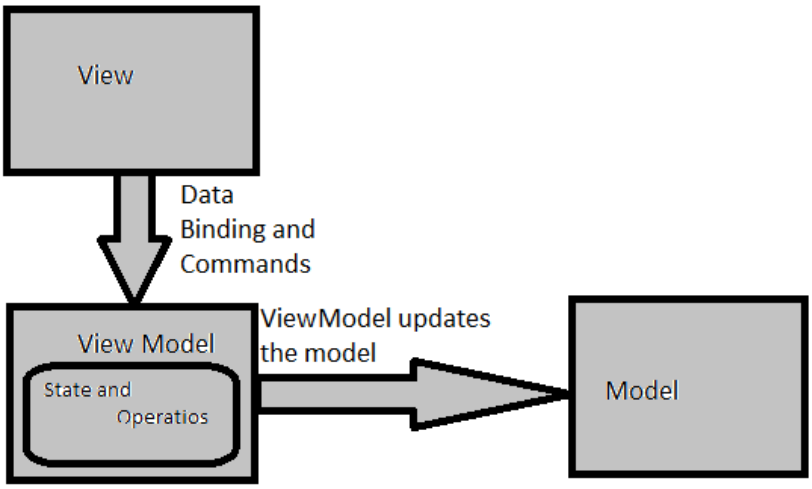
مثال

```
<script type="text/javascipt' src='knockout-2.js'">
</script>
```

حالا می توانیم از Knockout.js استفاده کنیم.

MVVM و View Model در Knockout.js

Model-View View Model یا به اختصار MVVM، در اصل یک الگوی طراحی (Design Pattern) برای ساخت رابط کاربری است. MVVM به ما توضیح می دهد که چطور یک UI پیچیده را با سه بخشی کردن آن، حفظ و نگهداری کنیم:



شکل1 MVVM

Model

ما در برنامه ها داده ها را ذخیره می کنیم و این داده ها یکسری عملیات و اشیاء در business domain فراهم میکنند. اجازه بدهید نمونه ای از حساب های بانکی را بررسی کنیم که می توانند پول را انتقال دهند. بنابراین Model مستقل از هر UI ای خواهد بود. وقتی از KO استفاده می کنیم، درواقع AJAX را در کد سمت سرور فراخوانی می کنیم تا داده های ذخیره شده در مدل را بخواند و بنویسد. اساساً MVVM، domain model پیاده سازی شده برنامه است و دربرگیرنده validation و business نیز می باشد. شی انتقال دهنده داده (data transfer object) و اشیاء ساده و قدیمی CLR، یا به اختصار POCOS، نمونه هایی از Model هستند.

View

اصلی ترین کاربرد View برای طراحی ساختار، طرح (Layout) و ظاهر چیزیسیت که کاربر، نهایتاً بر روی صفحه نمایش خواهد دید. اساساً View با استفاده از XAML تعریف می شود. در برنامه های موبایل تحت ویندوز، View صفحه ای از برنامه است و حتی می تواند یک جزء فرعی (sub-component) از View والد باشد. هر

View، Model والد خاص خودش را دارد و داده ها را با استفاده از یکسری binding و یا فراخوانی توابعی از View Model، فراهم می کند. برای پاسخگویی به تعاملات View، متد های مختلفی کد موجود در view model را اجرا میکنند. مانند کلیک بر روی یک دکمه و یا انتخاب آیتم. اگر view model هیچ آرگومانی نداشته باشد، می تواند در View، به عنوان Data Context متعلق به View معرفی شود.

View Model

View Model واسط بین View و Model است و دارای قابلیت کنترل منطق viewها می باشد. ViewModel داده ها را از Model برمی دارد و آنها را برای View آماده می کند. علاوه بر این ViewModel قادر به تعیین تغییر حالت منطقی است که یکسری جنبه های نمایشی view را تحت تاثیر قرار می دهد.

مزایای MVVM

استفاده از MVVM برای طراحی مجدد UI برنامه، بدون تغییر کد، بسیار ساده است. چون View به صورت کامل با XAML طراحی می‌شود. نسخه جدید از View با View Model ای که از قبل وجود دارد، کار میکند.

زمانی که توسعه دهندگان در حال ایجاد برنامه هستند، توسعه دهندگان و طراحان به صورت همزمان و مستقل از هم کار می کنند. زمانیکه توسعه دهندگان در حال کار کردن بر روی View Model و اجزاء Model هستند، طراحان می توانند به راحتی با نمونه دادهای (sample data) تولید شده کار کنند.

توسعه دهنگان می توانند تست های واحد (Unit Test) را، بدون استفاده از view، برای view model اجرا کنند. هر تست واحد برای view model قابل اجرا است و کارایی آن دقیقاً مشابه زمانی است که با view استفاده شود.

خلاصه

در این مقاله به مبانی و مزایای Knockout.js، MVVM و View Model در Knockout.js و مزایای MVVM پرداختیم.