This page will demonstrate client side and server side rendering performance difference through test, for discussion advantage between using client side and server side rendering under different conditions. The server side rendering performance is constantly similar during many years technology improvement. Firebug shows that total time spend on sever side between 70~200 milliseconds (for test below between 2008~2013). But the client side rendering performance is very different depending on CPU, browser types, and browser versions. Using DOM tech, IE6, Q8400 CPU, it takes over 30 seconds. IE10, i7-2600 CPU, it is 7~14 seconds. Chrome26, i7-2600 CPU, it spent 1~2 seconds. Champion is that, client side js generates html using text composition, FireFox 20, i7-2600 CPU, it takes only 35 milliseconds. But the same thing takes 7 seconds using IE10.

Client side rendering has advantage on

* When network band width is limited
* Smaller area to be render

Server side rendering has advantage on

* When using fast network
* Larger size and complicated UI to be render
* Client side CPU is slow
* OOP html generation and corresponding javascript auto-registration. (less restriction for architecture design)
* View logic on server side. (Do not need to transfer to client. Lower development and maintenance cost)

Using this UI framework, developers are free to choose server side render or client side render on each action.

**Dynamic Loading**

Base on Dynamic Loading technology, logically, we can build whole web application on one page. Because HTML and corresponding javascript and css are loaded incrementally when a user action involved with them, there is no preload pressure when page first time loading. So logically we can load unlimited things into a page. There are some obvious advantages by using this technique.

* **Good user experience**. For example, user will never been redirected away from current page.
* **High performance**. In real word, the typical user action response speed is 100~600 milliseconds (including network latency).
* **Reusability**. Because a page is composed by many smaller page parts dynamically. These page parts are also able to be re-combined or loaded on other pages for other purpose. For example, in TMK Quick Setup page, users are able to edit their personal information on a profile preview directly. That profile preview part is also used for Market Snapshot report page as report header.
* **Design flexibilities**.
* **Self-contain**.