Intro…

Code on demand is a method to deliver scripts from server side to end user side by using automated scripts triggered by end users. When a script requested by user’s actions such as clicking, hovering, or scrolling, the script will be loaded as an executable code from the server side, and it runs on the end user device to achieve certain object, for example animation, uploading data, and loading web-based application etc. Many implementations can achieve this feature including JavaScript, FlashPlayer by Adobe, and ActionScript etc. The most advanced way to implement code on demand is to use WebAssembly, which allows webs to run binary code and has better performance comparing with JavaScript.

Concept…

Code on demand is a very abstract concept due to the variety of its scope. In distributed computing, this is widely described as a technology that sends executable software codes from server to client device based on the client’s request. Code on demand could be a process that an end-user loads a whole single web application such as scanning the QR code presented on the What’s App web application. It could also be described as a process that the user calls a function from to execute a code on-demand, the executable code may be triggered by the user’s click on the page to load data or do something else.

Application…

Many years ago, when Adobe FlashPlayer used to be a major method to present the multi-media elements on webpages after the failure of Java applets (REFERENCE), it implemented this technology to load content based on user’s demand. This includes web animations and widgets, simple applications and games etc. However, as the world began to give up on Adobe FlashPlayer in 2017, the majority of web browsers started to encourage developers and users to use HTML5 as a standard to replace FlashPlayer. The Increase in Internet bandwidth and client devices’ performance created a massive demand on webpage design and development.

Nowadays the application of code on demand are mainly used in the following aspects in webpage development:

1. Front end rendering including frameworks

The concept of code on demand helps developer to achieve dynamically updated webpages and attractive interactions via JavaScript based frameworks such as Vue, WebAssembly and React. For example, users do not need to go to another link to browse additional contents. Instead, the webpage and its background scripts load these contents and update the page for the user.

1. Single page application (SPA)

SPA describes a web-based application has only one primary page, and all contents are loaded within in this primary page without redirect. When a user interacting with the page, functions and scripts will be loaded dynamically by controller module. This means SPA takes rendering job to user’s end devices, servers are only responsible for providing data and contents.

1. Code splitting

As end users’ demanding, webpages and applications become so fancy and power to use. The expansion of web applications requires massive bundle to support them to run, especially if the developer imports a large third-party repository without using CDN for some reason. End users need to wait a long period to load if the packed application is too large. Code splitting is a good solution to prevent this happening by achieve “lazy loading” to only load necessary parts. So, the concept of code on demand improves the flexibility.

Pros and Cons…

Based on the discussion above regarding the application of the concept of code on demand, it is very obviously to summarise its pros and cons:

Pros

1. Save resources and improve performance

Traditional webpages are created by server-side templates, and they are delivered as rendered html files from server by user’s request. In comparison, webpages using code on demand concept significantly reduces the server-side workload by giving rendering tasks to client side. Additionally, the client-side device only loads requested items and contents by user’s action. This also saves client-side resources by avoiding unnecessarily loading useless files and scripts.

1. Easy to manage

As a characteristic of this concept, code on demand separates rendering tasks and providing data. This allows the developer to focus on content delivery and script creation separately, and it gives the developer huge number of combinations of different modules. By only supporting data and contents, existing modules can render contents automatically, and the developer can easily remove or add a module to update the page.

Cons

1. Security concerns

Due to the increased number of communications between server and client because of code on demand concept. Both sides may start to show their vulnerability. If the server side does not implement anti-XSS (Cross-site scripting) attack, unauthorized JavaScript codes running on client side may lead to serious security issues. For example, without proper protection, a user can easily run his own code on the page…

Moreover, because the feature of code on demand, webpages using this concept will be highly relied on CDN (content delivery network) network to rendering pages. This creates a huge security concern regarding the security of CDN security. It is hard to count how many potential connections may be established between CDNs and end users for a very complicated webpage. These potential connections are hidden risks put end users and server sides under risks. If contents in CDN are intentionally replaced with malicious scripts, it will be very hard to manage.

1. High volumes request and long waiting time.

As we discussed above, a complex webpage using the concept of code on demand requires to establish lots of connections between end users and the server. If the webpage is poorly designed by creating too many interaction scripts with large files or codes waiting to load, plus there are tens of thousands of users interacting with the server. This will impact the stability and availability of the server side. Moreover, the long waiting time of the primary page may reduce the user’s experience if a large web application is not properly split into small packages.

Summary…

As webpages are becoming more powerful, loading contents from the server is not a simple action as before. The concept of code on demand brought developers to a new era. From the basic Applet to the most advanced WebAssembly, the concept of code on demand leads the development of JavaScript to an unprecedented level.

Examples…