Technical documentation

Project description

To do app is a task management app which empowers users to track activities and thing they need to get done. The user can enter a task and when submitted, the task is stored in a local or remote database, meaning this data can be access whenever the user wish to.

How the project works

The app works on a desktop as follow: the user can add an item to the list, mark an item active or completed and delete an item.

To add a task, a user enter the task in the text area of the app and press enter to save the task in the database. When the task is saved in the task need to mark it as active when the user has started the task but not yet finish it.

When the task has been executed, the user can move an active task into completed task by marking it by pressing a button.

The task can be removed from the database by pressing the delete button on the far right of each item.

Application performance

The website content is made of html, css, image, JavaScript and xmlhttprequest. The total size of the website content is 1.3 MGB, 74 % of content size is made up of JavaScript and XmlHttprequest, images and html take up 23% of the total size and the rest of the content share the 3% remaining.

During the performance check, we made 69 requests 29 request was for images (43%), 22 were requesting scripts (33 %) and 10 html (14%) and the remaining request were for other contents type.

We run the performance check for 2 minutes and 30 seconds. 2 % of this time was for loading, 9 % for scripting, 16% for rendering components and the rest of the time the application was idle. We can notice from this data that the loading is relatively quick and the rendering is relatively slow.

To improve the website performance, we will need to add expire headers, make fewer http request, use cookies free domain and reduce DNS look up.

Why do we need to do it to improve performance?

As web page become complex with a variety of component to load, first time visit to a page require several http request to load all the necessary components. Expire header will allow

components to be cashable and improve the speed of loading component by reducing http request needed.

It is obvious that decreasing the number of request will improve the speed of page load and improve the overall performance. This can be achieved by reducing the number of components: by combining file, multiple scripts into one script, multiple stylesheet into one and use CSS Sprite and image maps.

To work around the fact of sending unnecessary cookies to the server, we have to make sure static component make cookies-free request by creating a subdomain and hosting them there.