Why Does the Page Need to be Refreshed to Display Updates on the Website?

FlexTable - Sheets to WP Table Live Sync · https://wppool.dev/docs/why-page-needs-refresh-to-display-updates/

Our plugin, FlexTable, fetches data from a Google Sheet via Google AppsScript and the Sheet's export URL in CSV format. When changes are made to the Google Sheet, those updates are reflected in the exported CSV file. However, to display the updated data on your WordPress site, the plugin needs to retrieve the latest CSV file and process it into a table format.

This process is initiated by a fresh request to the Google Sheets export URL, which fetches and parses the updated data. To trigger this data-fetching process, the page must be refreshed. A page reload ensures that the plugin pulls the most up-to-date data and displays it correctly. Without this refresh, the plugin may continue displaying outdated information, as it needs to initiate a new request to pull in the updated CSV file.

Why Not Real-Time Synchronization?

Google AppsScript is built around a request-response model and is not designed to support real-time, persistent connections or live streaming of data. Each time the plugin fetches data, it sends a request to the Google Sheet and processes the response. This approach, while effective for retrieving updated data, does not allow for the live synchronization of changes. Here are some reasons why real-time syncing isn't feasible:

1. **Execution Time Limits**: Google AppsScript executions typically have a time limit of 6 minutes. Maintaining a persistent connection for real-time updates could cause timeouts and prevent the script from processing requests successfully.
2. **Resource Consumption**: Continuously polling Google Sheets for updates would require frequent requests to check for changes. This can consume a significant amount of server and client-side resources, which could negatively impact performance and slow down the user experience.

3. **Lack of Support for Persistent Connections**: Google Sheets and Google AppsScript do not support technologies like WebSockets or other methods for real-time communication. Implementing such persistent connections to Google Sheets is not feasible with the current infrastructure.

Why a Page Reload Works

Refreshing the page triggers the data-fetching process by initiating a new HTTP request to the Google Sheet’s export URL. This ensures that the plugin retrieves the most recent data and renders it accurately. The page reload serves as a "reset," helping to avoid issues like incomplete or outdated data being displayed due to connection errors, broken responses, or network problems.

In summary, while real-time updates would be ideal in some cases, the limitations of Google AppsScript and the nature of HTTP requests make a page reload necessary to ensure that FlexTable always displays the latest data.