# Build and Source Control - Feature #8362

## implement a web-based UI for "project creation" and deliver it as a docker image

02/28/2024 08:57 AM - Greg Shah

| | | | | |
|---|---|---|---|---|
| **Status:** | New | | **Start date:** | |
| **Priority:** | Normal | | **Due date:** | |
| **Assignee:** | | | **% Done:** | 0% |
| **Category:** | | | **Estimated time:** | 0.00 hour |
| **Target version:** | | | | |
| **billable:** | No | | **version:** | |
| **vendor_id:** | GCD | | | |

**Description**

## History

**#1 - 02/28/2024 09:53 AM - Greg Shah**

The idea is that it will be a standard public image that can be executed with a single command line by a customer, and will allow the creation of a new FWD project on the host filesystem. It would be a cleaner, better way than we currently have:

- web based instead of a hacked up prepare command line script
- hide the complexity inside FWD and the docker image instead of forcing the user to deal with all the project mess

Before we do this, I want to clean up the project structure and approach. Just implementing the same approach here is not OK. By this I mean tasks such as #6256, #6320, #5586, #6552.

If we go this way, then the web UI might be primarily about defining the project-level .json file. Then a subsequent docker image can be run to use the results of this created "project". Inside that file, it would define how to find the 4GL code, the .df files and other key configuration.

The "dream scenario" is that our entire project definition is inside a single .json file. The conversion code (in Java) would actually create the destination directories (the directory structure for the intermediate and final output) as needed, so long as it is told the top-level project directory which wouldn't be in the .json so that the same .json can be used with any top-level project dir.