

Change request log

1. Concept Location

| Step # | Description | Rationale |
|--------|--|--|
| 1 | We started the system. | |
| 2 | We interacted with the system: logged in as admin, and looked at watch list, and point details interfaces. | To understand where the point value need to be truncated and what can be the name of the variable. |
| 3 | We looked in the URL and spotted "DataPointDetails". | To identify the name of the JSP page being rendered. |
| 4 | Searched for term DataPointDetails and found 22 results in 8 files, from which selected view.js file and DataPointDetails.jsp. | Because the file view.js file had the function watchlist.setDataImpl, pointDetails.setData which were directly rendering the data in the frontend. Other file dataPointDetails.jsp is rendering history table using getHistoryTable() method. Other files are not directly connected to render the data at view level. |
| 5 | We inspected the file DataPointDetails.jsp file and looked into getHistoryTableData() method. | We noticed that the getHistoryTableData() method is rendering the table and values into it. |
| 6 | We inspected view.js file, and visited mango.view.pointDetails.setData() function and mango.view.watchList.setDataImpl() function. | We noticed the values in watchlist and also in pointdetails values are being generated using this two functions. |
| 7 | We have put debugging point into these three locations. | To understand the values are being rendered from these three functions or not. |
| 8 | We marked "value.js" and "DataPointDetails.jsp" as the concept location. | We confirmed that these two files need to be modified in order to make changes at view level. |

Time spent (in minutes): 39

Classes and methods inspected:

- \mangoSource\war\WEB-INF\jsp\dataPointDetails.jsp
 - getHistoryTableData()
- \mangoSource\war\resources\view.js
 - mango.view.watchList.setDataImpl()
 - mango.view.pointDetails.setData()
- \mangoSource\src\com\serotonin\mango\web\mvc\controller\DataPointDetailsController.java
- \mangoSource\src\com\serotonin\mango\web\dwr\MiscDwr.java
- \mangoSource\src\com\serotonin\mango\web\dwr\DataPointDetailsDwr.java

2. Impact Analysis

| Step # | Description | Rationale |
|--------|---|---|
| 1 | We marked view.js and DataPointDetails.jsp, as changed, and the neighbors of it as to be inspected. | To track down the impact of the change we will make to other files. |
| 2 | We searched view.js files using the IDE's search tool and found only PublicView.jsp as a result. | Because we need to find out the neighbors of the view.js files |

| | | |
|---|---|---|
| 3 | We inspected PublicView.jsp file, and we marked as inspected and unchanged. | Because PublicView.jsp importing the view.js file as a source and not directly interacting with the concept locations. |
| 4 | We tried to look for DataPopintDetails.jsp neighbors. But no direct result was being found. | Because the file is view file, which will be rendered using view resolver, so the changes in the DataPointDetails.jsp will be limited only to the view level. |

Time spent (in minutes): 40

- \mangoSource\war\WEB-INF\springDispatcher-servlet.xml
- \mangoSource\src\com\serotonin\mango\web\mvc\controller\DataPointDetailsController.java
- \mangoSource\war\WEB-INF\jsp\dataPointDetails.jsp
 - getHistoryTableData()
- \mangoSource\war\resources\view.js
 - mango.view.watchList.setDataImpl()
 - mango.view.pointDetails.setData()
- \mangoSource\war\WEB-INF\jsp\publicView.jsp

3. Actualization

| Step # | Description | Rationale |
|--------|--|--|
| 1 | We added the wrapper to truncate the float value up to 2 digits after the decimal. | In order to fulfill the change requirement, we have added the wrapper to the variable that was rendered to the frontend. |
| 2 | parseFloat(value).toFixed(2) is added around the variable. | parseFloat will convert it to the float value and the toFixed(2) will limit the 2 digits after the decimal point. |
| 3 | Implementation for the detailed view. | We have implemented this for the detailed view of the values also. Also, the values will be truncated for the values present in the History table in the file dataPointDetails.jsp |
| 4 | Implementation for the home page tabular view. | We have implemented this for the values that appears on the table in the file view.js |
| 5 | We performed manual testing for the new change and performed functional testing. | To make sure everything works. |

Time spent (in minutes): 22

Inspected: -

- \mangoSource\war\WEB-INF\springDispatcher-servlet.xml
- \mangoSource\src\com\serotonin\mango\web\mvc\controller\DataPointDetailsController.java
- \mangoSource\war\WEB-INF\jsp\dataPointDetails.jsp
 - getHistoryTableData()
- \mangoSource\war\resources\view.js
 - mango.view.watchList.setDataImpl()
 - mango.view.pointDetails.setData()
- \mangoSource\war\WEB-INF\jsp\publicView.jsp

Changed: -

- \mangoSource\war\WEB-INF\jsp\dataPointDetails.jsp
 - getHistoryTableData()
- \mangoSource\war\resources\view.js

- function truncate()
- mango.view.watchList.setDataImpl()
- mango.view.pointDetails.setData()

4. Validation

| Step # | Description | Rationale |
|--------|--|--|
| 1 | Test case defined: Watch list component values should be truncated. Expected output: Values truncated to 2 decimal places | This is the regular expected behavior. The test passed. |
| 2 | Test case defined: Point details component values should be truncated. Expected output: Values truncated to 2 decimal places | This is the regular expected behavior. The test passed. |
| 3 | Test case defined: History Table component values should be truncated. Expected output: Values truncated to 2 decimal places | This is the regular expected behavior. The test passed. |
| 4 | Test case defined: Statistics Table component values should not be truncated. Expected output: Values are NOT truncated to 2 decimal places | This is the regular expected behavior. The test passed. |

Time spent (in minutes): 15

5. Summary of the change request

| Phase | Time (minutes) | No. of classes inspected | No. of classes changed | No. of methods inspected | No. of methods changes |
|------------------|----------------|--------------------------|------------------------|--------------------------|------------------------|
| Concept location | 39 | 5 | 0 | 3 | 0 |
| Impact Analysis | 40 | 5 | 0 | 3 | 0 |
| Actualization | 22 | 5 | 3 | 3 | 3 |
| Verification | 15 | 5 | 2 | 3 | 0 |
| Total | 114 | 20 | 5 | 12 | 3 |

6. Conclusions

For this change, concept location was relatively difficult because the system is quite complex and multiple sub-changes were involved in different locations. Since quite a few classes and methods were involved, Impact analysis also had to be thorough and rigorous. The Actualization was quite simple once the Concept location was identified. Also, as the change is implemented at the view level of the application, actual values remain the same only but will be truncated for the display to the user. Then Manual and Functional Testing was performed as per the test cases and the functionality required to verify its proper working.