Claris Engage 2020

Making a New Friend: Virtual vs. JSON

Demo File: Virtual_v_JSON

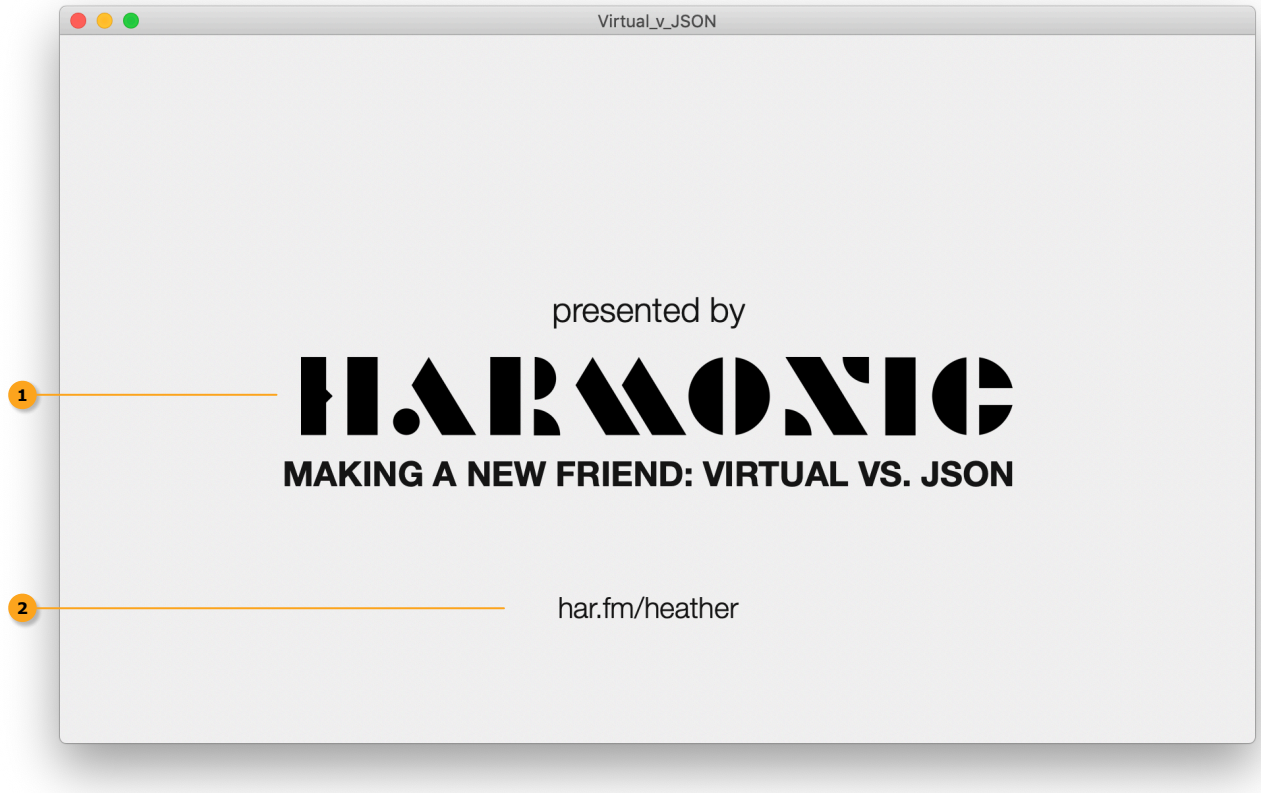
v1.0 August 3, 2020

Read Me

Quick Tour

Splash Screen

When first opening the demo file, you will see the splash screen for a few moments before moving to the JSON Report..
You will have the option to return to the Splash screen from the Menu.

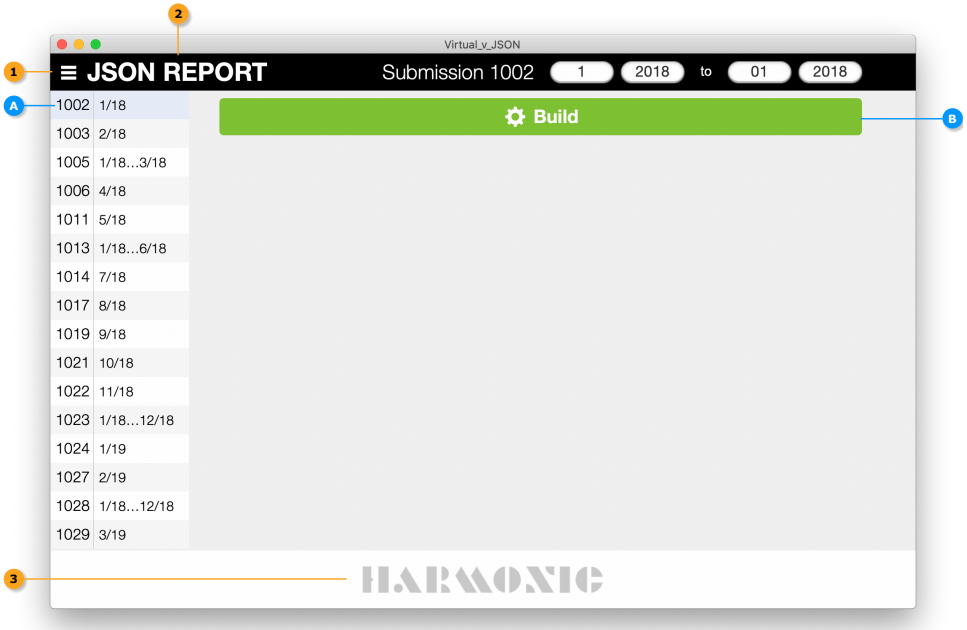


Options

1. Click to return to the JSON Report.
2. Click to open har.fm/heather in your browser.

All primary layouts include the following

1. Open Menu.
2. Layout Title: Click to adjust window size.
3. Click to visit Harmonic

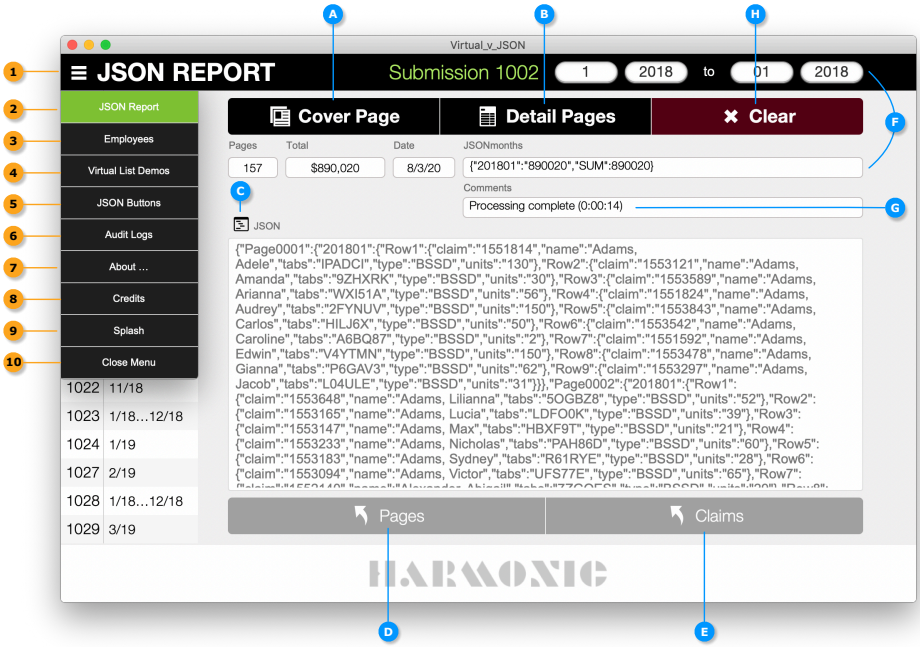


JSON Report

A. Select from the 37 Submissions listed.
B. Select to build the JSON-based report for your selected Submission

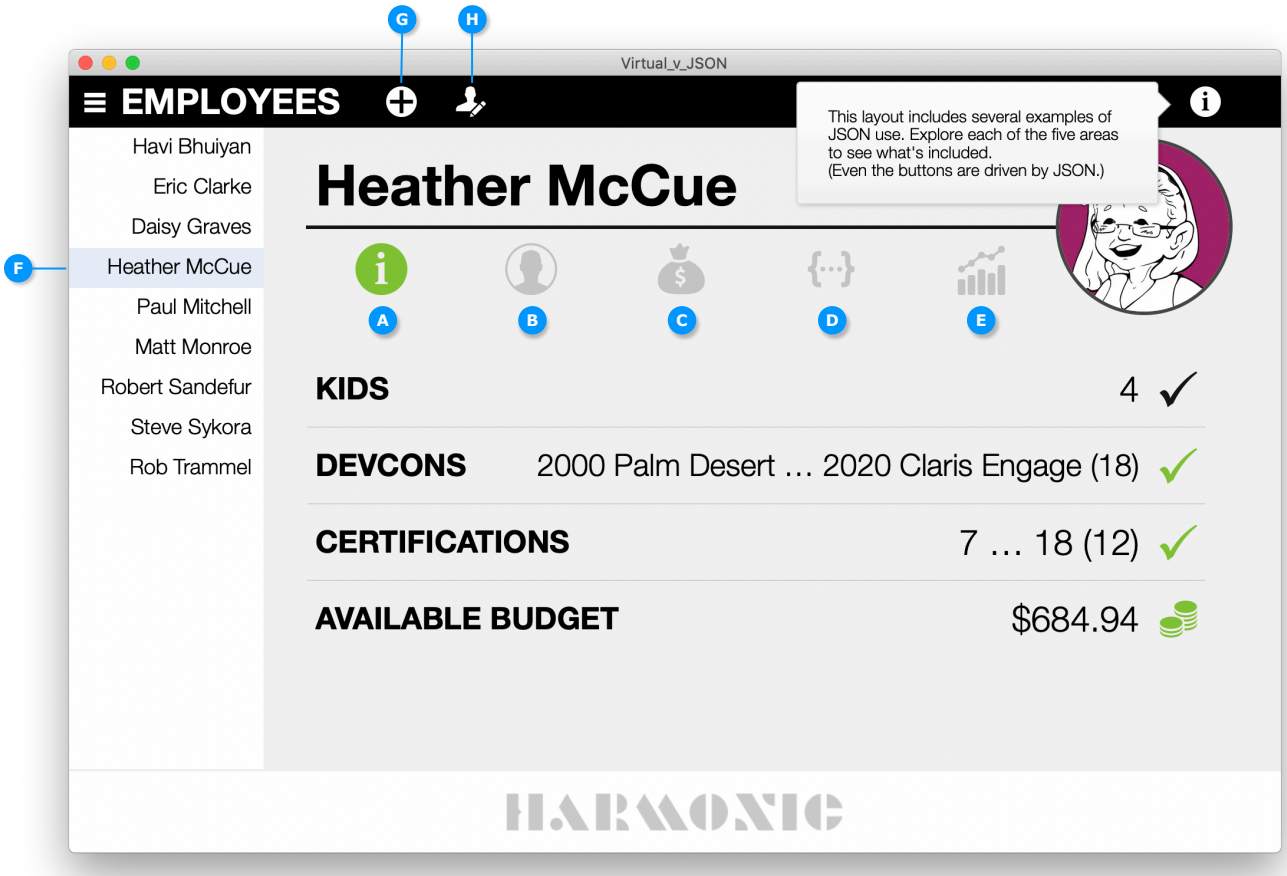
Post-Build

A. View the Cover Page.
B. View the Detail Pages.
C. View the full JSON, formatted.
D. Go to the related StatePages records.
E. Go to the related ClaimSubmission records.
F. Month range covered by the Submission.
G. The processing time is recorded in the comment.
H. Select to clear the current Submission entirely so it can be re-run.



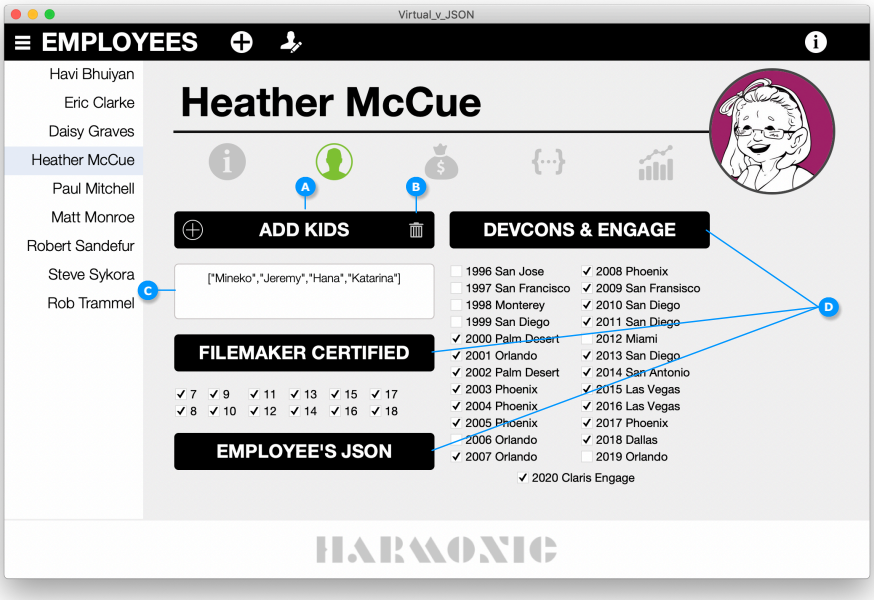
Menu

1. Open Menu.
2. Go to JSON Report demo.
3. Go to Employees for multiple JSON demos.
4. Go to the Virtual List demos.
5. Go to the JSON Buttons, where all of the JSON-driven demo instructions reside.
6. Open the Audit Log demo.
7. Go to the About... page, where you'll find more information about what's included on each of the demo pages.
8. View the Credits (includes links to individuals and resources).
9. Return to the Splash screen.
10. Close the menu.



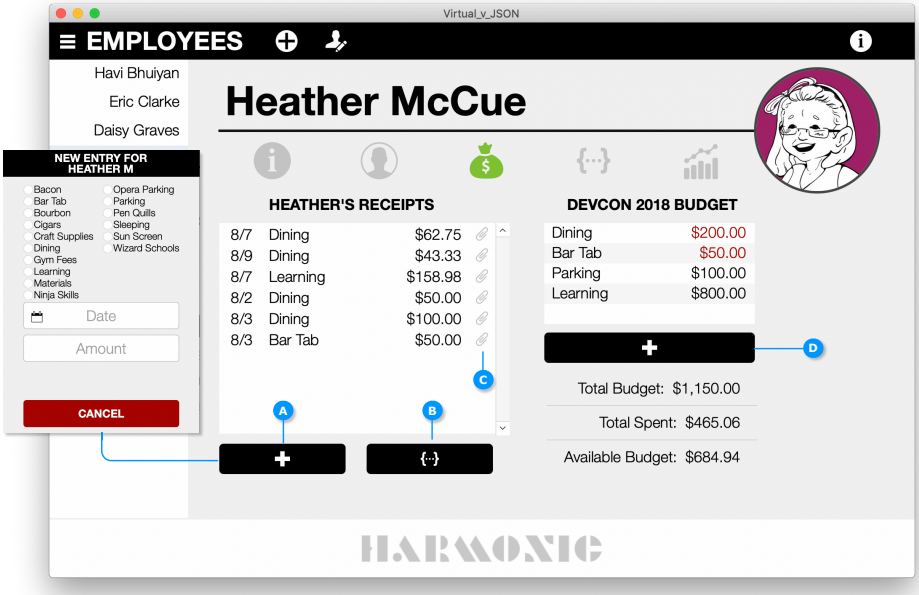
Employees — Summary

A. Summary dashboard, with data derived from JSON.
B. Metadata, where the details behind the Summary are modified.
C. Employee Budget, demonstrates how expenses are validated using a “One Truth” JSON source.
D. Here you’ll find demos of the FileMaker JSON Functions, and a catalog describing some of our favorite custom functions.
E. JSON-driven web viewer-based report.
F. Select a different employee to play with.
G. Add a new employee record.
H. Add/Edit the employee name.



Employees — Metadata

A. Select to add children to the JSON.
B. Clear all kids.
C. JSON array of the employee's kids.
D. Click to view the values in JSON



Employees — Budget

A. Add an expense for the employee.
   - Select a category
   - Enter date and amount, then submit
   - If the employee has sufficient funds remaining in their budget for the selected category, the expense will be added; if not you will be advised as to why validation failed.

B. View the JSONFunds code and formatted results.
C. Attach a receipt.
D. Add a new item to the employee's budget.

Employees — JSON Function Demos

A. Click to update the JSON using the currently selected FileMaker function.

B. The JSON results.

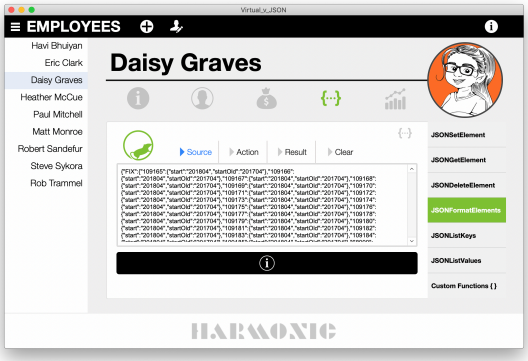
C. View the formatted JSON results.

D. View the actual code used by the demo buttons on your current panel.

E. Select a different JSON function to test.

Demos for JSONFormatElements, JSONListKeys, and JSONListValues

Each of these demos use large data samples that come from the *MouseWheel*™, a JSON-driven engine used by one of our production systems.



- The *Source* JSON contains original values from records that have been queued to test for possible corrections.

  The MouseWheel evaluates each value for discrepancies (based on new, distantly related activity) and returns instructions on which need to be fixed.

- The *Action* JSON contains instructions; which records to fix, original start date, and correct start date.

  The MouseWheel follows the instructions to update values that have changed.

- The *Result* JSON contains a reporting of the changed fields, their corrected values, and the error status for each fix processed by the MouseWheel.

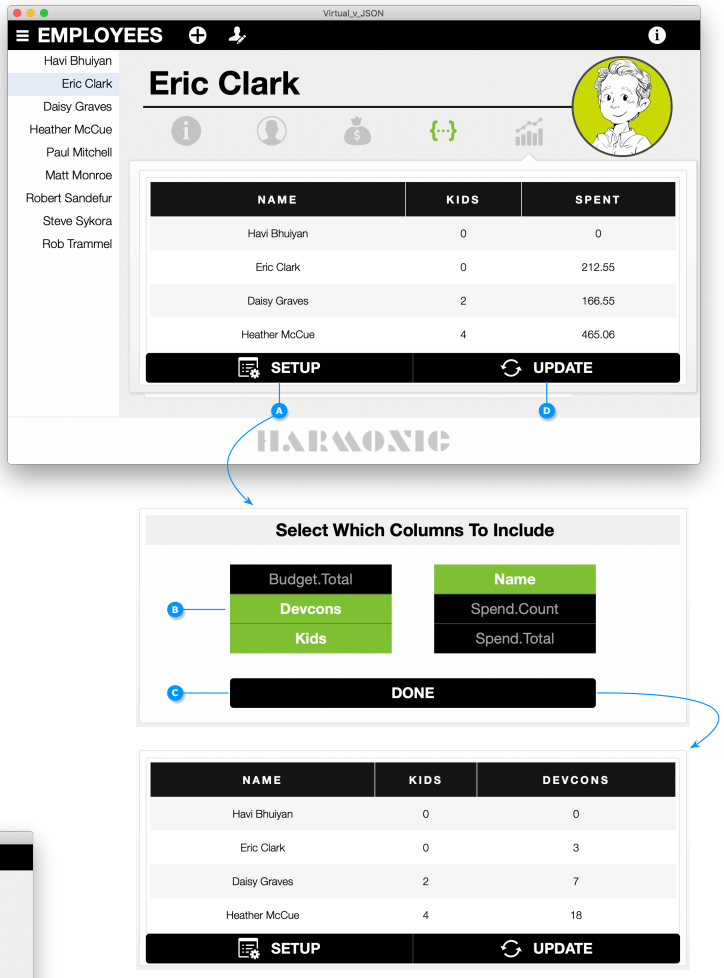
JSONFormatElements — Select *Source*, *Action*, or *Result* to populate the JSON values, then select the info button to view the formatted JSON.

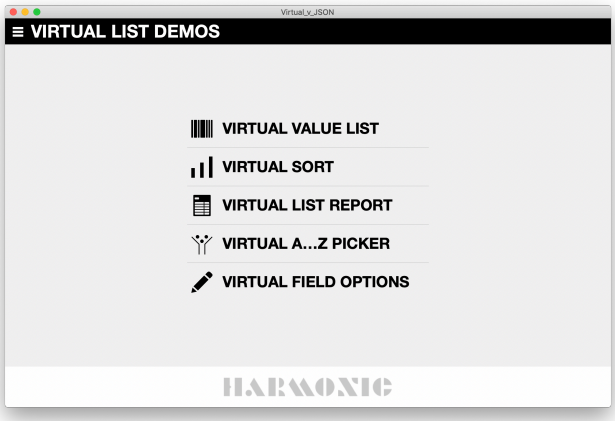
JSONListKeys — Select *Source*, *Action*, or *Result* to populate the JSON values, then select *Top*, *2nd*, *3rd*, and *4th Level* to see the Keys at each level of the JSON. Not all of the JSON examples include four levels, so try each one to see the differences. Select the info button to view the different keyPaths used for each JSON example.

JSONListValues — This demo expects your sample JSON to come from the MouseWheel *Result*, so if you see a red *MouseWheel*, return to one of the previous demo tabs, select *Result*, then return to JSONListValues where you can now select the button labeled "From Results JSON …" to see the actual calculation being used along with its results.

Employees —
Web Viewer Report

A. Select to view the setup options.
B. Select different columns to include on the report.
C. Return to the report.
D. Refresh web viewer, if needed.





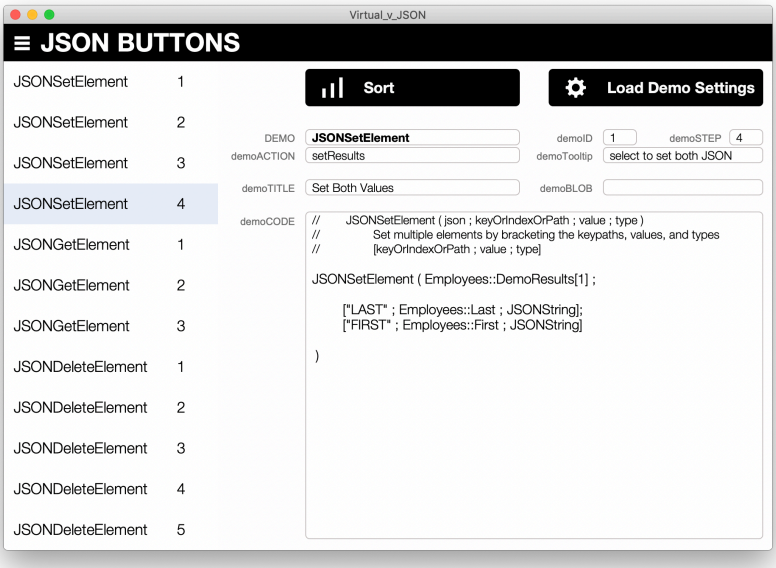
Virtual List Demos

- **Virtual Value List**
  - Demo of the Virtual Value List as used in a production system to prevent expenses from being allocated against unavailable budgets.
- **Virtual Sort**
  - Demo of how the Virtual Sort technique can be used to provide end users with the ability to sort portals as they wish, without complicated hard-coded sort orders in either the relationship or portal setup.
- **Virtual List Report**
  - Generate the Virtual List version of the same State Billing Report shown in my session. Compare this to the JSON versions for the same 37 Submission examples.
- **Virtual A...Z Picker**
  - See how the Virtual List can be implemented as a functional UI tool. (Under the hood, check out how the portal relationship filters the results using our permutate custom functions.)
- **Virtual Field Options**
  - See additional options for Virtual List fields.

JSON Buttons

Each record in the JSON table supports a specific button action from the Employees' JSON Function Demos. The data here defines multiple attributes associated with each button.

- Action type to perform (demoACTION)
- Button label (demoTITLE)
- Tooltip (demoTooltip)
- Script Parameters to use (demoCODE)
- Sample data to use, when applicable (demoBLOB)

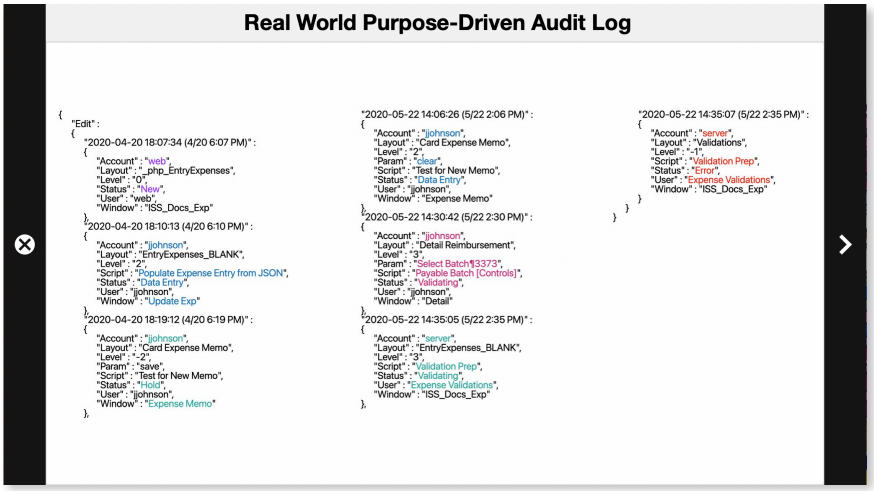


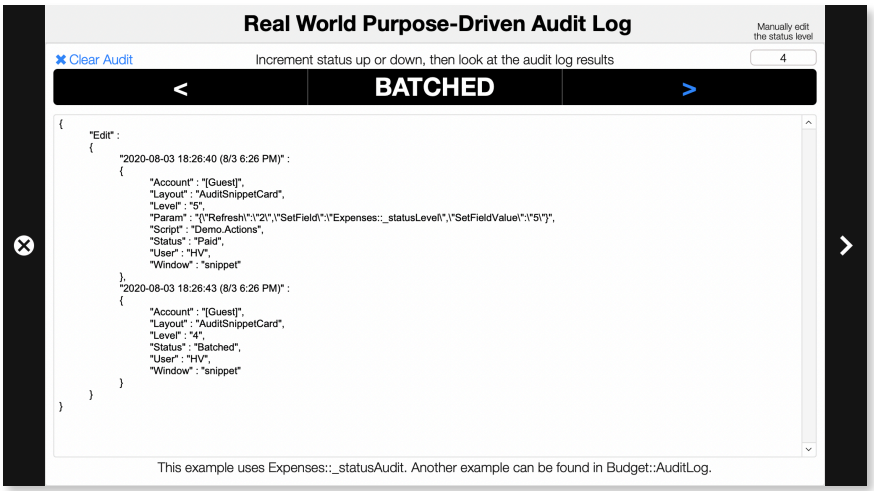
Audit Logs

This section includes a production example from a status-driven audit log.

Move to the next panel to test it yourself. Every time the status is changed, the audit log updates.

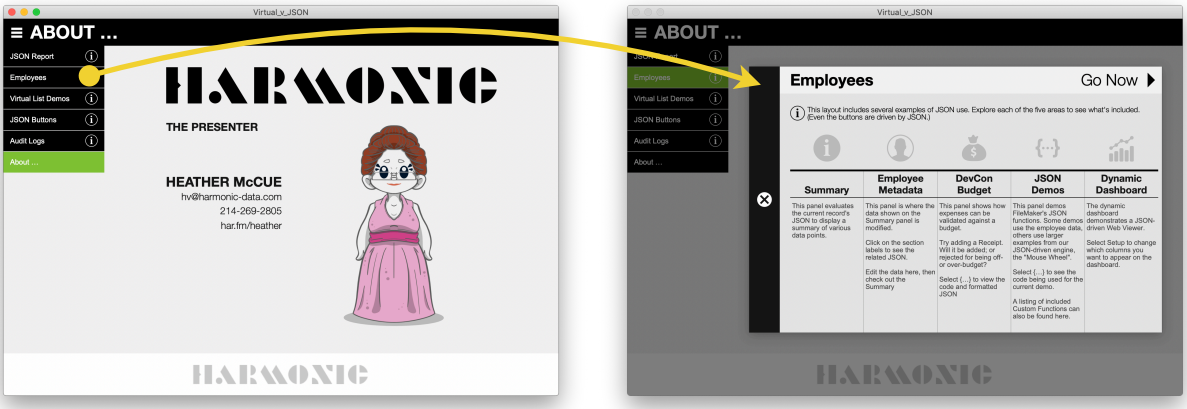
- Increment the status forward.
- Increment the status back.
- Manually edit the status to see the difference in what's recorded by the Audit log.
- Clear the audit log results entirely and start all over.





About ...

This section includes an overview of what's in each area of the demo file.



Custom Functions Catalog

#Assign (parameters)
Parses a Let dictionary of name-value parameters into a series of locally scoped $variables.

#Set (name ; value)
Returns a name-value pair in Let notation for use by #Assign.

•Error
A shorthand function to avoid typing "Get (LastError)".

•Rep
A shorthand function to avoid typing "Get (CalculationRepetitionNumber)".

Audit (FieldList ; AuditJSON ; ExcludeList ; i)
Returns a JSON Array of field name-value pairs used to capture an audit when the users change a record.

DateFromISO (iso8601Date)
Parses date data formatted according to ISO 8601, and casts it to a FileMaker date. Dates with limited accuracy will return the first date in the specified period.

DemoTooltip (ButtonID)
Pulls each demo button's tooltips from the $$DEMO JSON.

DollarsFormat (Number)
Converts unformatted number into dollars with commas rounded to the nearest penny.

ErrorDescription (errorCode)
Returns the English description for the error number returned by the Get(LastError) function.

FormatTimestamp (format ; tstamp)
Converts a FileMaker-formatted timestamp into an alphabetically sortable format, or any other permutation desired.

GetAsQueryDate (FMDate)
Converts a FileMaker date value into a date format for use in a SQL query.

GetDeliniatedValue (Text ; Value ; Delimiter)
Returns the requested value from a string delimited by a named value.

IsFrontTab (TabName)
Used to confirm that a specific tab or slider panel is currently active. Applicable to conditional formatting, hide conditions, scripting, etc.

ISOFromDate (theDate)
Returns a date rendered in ISO 8601 format: YYY-MM-DD

JSONArrayEval (Array ; FMfunction)
Evaluates a JSON array of values using common FM functions.

JSONArrayFromList (_list)
Converts a return-delimited list of values into a properly formatted JSON array.

JSONArrayFromList_Numeric (_list)
Similar to JSONArrayFromList, but specific to numeric values where quotes are unnecessary or otherwise undesirable.

JSONArrayToList (Array)
Converts a JSON array into a return-delimited list.

JSONError (response)
Tests for invalid JSON results.

JSONgetAllValues (json ; paths ; path)
Returns a list containing all values from a JSON blob.

JSONListAllPaths (JSON ; starting_key ; final_paths ; filter_paths)
This function recurses through a JSON object and returns all final paths.

JSONSetList (JSON ; RelatedKey ; RelatedJSON)
Recursively sets a related array into a JSON object.

JSONSetListCore (JSON ; Related_Key_List ; Related_JSON_List)
This function recursively sets a related array into a JSON object (used by JSONSetList).

JSONtoHTMLTable (JSON ; ColumnJSON ; r ; Result)
Returns an HTML Table from a 2 dimensional JSONArray with HTML cell tags.

JSONtoHTMLTableCell (JSON ; ColumnJSON ; r ; c)
Returns a cell from a 2 dimensional JSONArray with HTML cell tags.

JSONToHTMLTableRow (JSON ; ColumnJSON ; r ; c ; Result)
Returns a row from a 2 dimensional JSONArray with HTML row tags.

leftVal (text)
Returns the left-most value from a list, without a trailing return.

ListToString (array ; delimiter)
Converts a return-delimited list into a custom-delimited string.

middleVal (text ; startVal)
Performs a similar operation to FileMaker's MiddleValues function, but without the ¶ character on the end. Unlike the MiddleValues ( ) function which can return multiple values, this returns a specific single value.

notEmpty (field)
A shorthand replacement for: not IsEmpty (field).

OutlineJSON (json ; RunningList ; RunningPath ; i)
Returns a list of all keys from within the JSON.

permutate_letters (text ; result)
Takes a given string and returns an array where each value is truncated by one additional character (used by permutate_words).

permutate_words (text ; result)
Takes a given string and returns an array where each value truncated by one additional character, and repeats this process truncating the string by one additional word.

Repeat (text ; repeatCount)
Takes a given string and concatenates it onto itself a specified number of times.

ReverseJSONArray (Array ; i ; NewArray)
Returns a JSONArray in the reverse order of the supplied JSON Array.

rightvalremain (values)
Returns the remaining values in a stack, without the first.

SortButton (portalKey ; sortField ; portalNumber)
Formats the various key-value pairs for submission of the script parameter required by the script, Virtual Sort by SQL v2.

SortOrder (fieldName ; sortID)
Returns the current sort order of the specified field when using a Virtual Sort. Useful for UI indicators.

SQLFieldName (field)
Creates a SQL-safe field reference that will not have to be manually updated if the field name changes.

SQLTableName (field)
Creates a SQL-safe table reference that will not have to be manually updated if the table name changes.

StateType (BillType)
Returns the correct classification for billing submission to the state. (Specific to a client's use case.)

TrimReturns (text)
Strips away all leading, ending, and doubled returns from a text string.

valPos (array ; value)
Returns the first location of a given value within a list. (Faster in some instances than similar PatternCount-based functions because this stops evaluating with the first identified instance.)

VirtualSort... and VirtualValueList...
Several custom functions required by the VirtualSort and VirtualValueList techniques are also included.

ValueExists (listofValues ; searchString)
Returns a 1 or 0 for true or false if the 'searchstring' value exists with in the 'listofValues' (used by Audit).

win_margins (popupHeight ; popupWidth ; relativeposition)
Calculates the margins for positioning a new window relative to the current window.

win_size (trigger)
Sets global variables for all window positioning values, or Title.

YYYYMM (AnyDate)
Converts any date into a YYYYMM-formatted text string (which will also sort in date order if used as a JSON key).

YYYYMMDD (AnyDate)
Converts any date into a YYYMMDD-formatted text string (which will also sort in date order if used as a key).

YYYYMMtoDate (YYYYMM)
Converts a YYYYMM-formatted value into a valid FileMaker date.