Sqlgrid User Guide

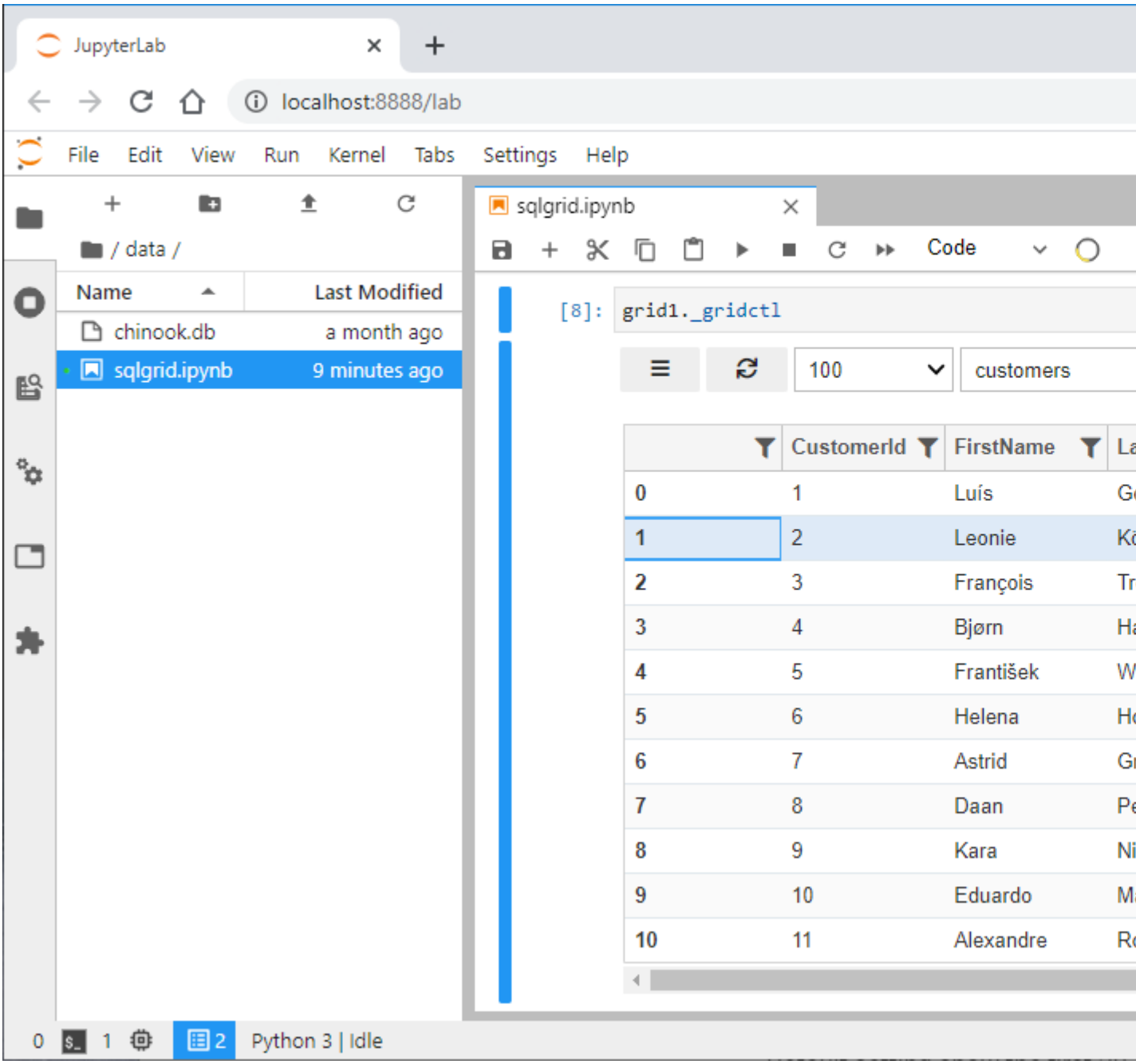
Gilbert Brault

Sea Dev

Table of Contents

SQLGRID¶

Browsing SQL Database from Jupyter Notebooks



Note

SQLGRID is derived from Qgrid work.

SQLGRID Features and Usage¶

Browsing SQL Database from Jupyter Notebooks

Grid layout features¶

- Select any table from the database (optional)
- Use database table column names as headings
- Show/hide column
- Resize column width
- Default setting show the first 20 columns
- Placing Column by grabbing the heading, dropping where needed
- Column sorting
    - Ascending
    - Descending
- Filtering based upon column content
- Filtering widgets depending upon column data type
    - Date (date range with date Picker)
    - String
    - Number (slider)
    - Boolean
- Convert column type for ease of filtering
    - from string to date
    - number to string

How to setup in a notebook?¶

SQLGRID module brings two objects the user must setup

```
sqlData     # The database connection and context.
sqlgrid     # The UI grid layout and controller.
```

sqlData setup¶

To use an sqlData object, you must first import its definition from the sqlgrid module

```
import sqlgrid
from sqlgrid.sqlData import sqlData
import ipywidgets as widgets
out = widgets.Output(layout=widgets.Layout(border='1px solid black'))
databasePath = ...
sql = sqlData(path=databasePath, out=out)
```

And then you need to pass

- The database connection string (Path)
- a widget output (ipywidgets.Output) where to write any feedback (out)

The Path is a SQLAlchemy database Url. See the paragraph Database Urls in the SQLAlchemy documentation (scroll a bit from this top page and go to the paragraph) to learn about the format of those string connexions.

If the Jupyter notebook and the database are in the same directory -- for example the chinook database (chinook.db)

```
databasePath = "sqlite:///./chinook.db"
```

is valid.

For more information about sqlData object, after executing the above code, use:

```
help(sql)
```

sqlgrid setup¶

the sqlgrid object requires the following variables

- an sqlData object providing the database connection and context (see above)
- a grid option object setting main behavior for the grid (see thereafter)
- tableScan is the option to see the list of all database table and control table selection

```
# create the sqlgrid widget
grid = sqlgrid.gridctl(sql, grid_options=grid_options, tableScan=True )
# display the grid widget in the current cell output
grid._gridctl
```

grid options setup¶

the grid options is a dictionnary with two main set of variables

- the one controlling the SlickGrid features
- the other controlling the Qgrid (SQLGRID) features

SlickGrid is the javascript and CSS libray controlling the frontend. Please see full documentation here

```
grid_options = {
    # SlickGrid options
    'fullWidthRows': False,
    'syncColumnCellResize': True,
    'forceFitColumns': True,
    'defaultColumnWidth': 150,
    'rowHeight': 28,
    'enableColumnReorder': True,
    'enableTextSelectionOnCells': True,
    'editable': True,
    'autoEdit': False,
```

```
    'explicitInitialization': True,
    'enableCellNavigation': True,

    # Qgrid options
    'maxVisibleRows': 10,
    'minVisibleRows': 8,
    'sortable': True,
    'filterable': True,
    'highlightSelectedCell': True,
    'highlightSelectedRow': True
}
```

Find thereafter the full set of SlickGrid options according to SlickGrid Wiki

Option	Default	Description
asyncEditorLoading	false	Makes cell editors load asynchronously after a small delay. This greatly increases keyboard navigation speed.
asyncEditorLoadDelay	100	Delay after which cell editor is loaded. Ignored unless asyncEditorLoading is true.
asyncPostRenderDelay	50	
autoEdit	true	Cell will not automatically go into edit mode when selected.
autoHeight	false	This disables vertical scrolling.
cellFlashingCssClass	"flashing"	A CSS class to apply to flashing cells via flashCell().
cellHighlightCssClass	"selected"	A CSS class to apply to cells highlighted via setHighlightedCells().
dataItemColumnValueExtractor	null	
defaultColumnWidth	80	
defaultFormatter	defaultFormatter	The default formatter if no other formatter is specified. See [[custom formatter
editable	false	
editCommandHandler	queueAndExecuteCommand	*Not listed as a default under options in slick.grid.js*
editorFactory	null	A factory object responsible to creating an editor for a given cell. Must implement getEditor(column).
editorLock	Slick.GlobalEditorLock	A Slick.EditorLock instance to use for controlling concurrent data edits.
enableAddRow	false	If true, a blank row will be displayed at the bottom - typing values in that row will add a new one. Must subscribe to onAddNewRow to save values.

Option	Default	Description
enableAsyncPostRender	false	If true, async post rendering will occur and asyncPostRender delegates on columns will be called.
enableCellRangeSelection	null	**WARNING**: Not contained in SlickGrid 2.1, may be deprecated
enableCellNavigation	true	Appears to enable cell virtualisation for optimised speed with large datasets
enableColumnReorder	true	
enableRowReordering	null	**WARNING**: Not contained in SlickGrid 2.1, may be deprecated
enableTextSelectionOnCells	false	
explicitInitialization	false	See:Example: Explicit Initialization
forceFitColumns	false	Force column sizes to fit into the container (preventing horizontal scrolling). Effectively sets column width to be 1/Number of Columns which on small containers may not be desirable
forceSyncScrolling	false	
formatterFactory	null	A factory object responsible to creating a formatter or Custom-Formatter for a given cell. Must implement getFormatter(column).
fullWidthRows	false	Will expand the table row divs to the full width of the container, table cell divs will remain aligned to the left
headerRowHeight	25	
leaveSpaceForNewRows	false	
multiColumnSort	false	See:Example: Multi-Column Sort
multiSelect	true	
rowHeight	25	
selectedCellCssClass	"selected"	
showHeaderRow	false	
syncColumnCellResize	false	If true, the column being resized will change its width as the mouse is dragging the resize handle. If false, the column will resize after mouse drag ends.
topPanelHeight	25	

User Interface