spooky-console Documentation

Robert Russell

Oct 19, 2018

CONTENTS

CHAPTER

ONE

API REFERENCE

1.1 main Module

TODO

1.2 exceptions Module

TODO

exception `spookyconsole.exceptions.`**`AbortPromptLoop`**
TODO

exception `spookyconsole.exceptions.`**`TableFormatError`**
TODO

1.3 ntutils Module

TODO

class `spookyconsole.ntutils.`**`BooleanArrayParamType`**(*enclosing_chars=”*, *require_enclosing_chars=False*, *item_sep_char=’, ’*)

`item_hook`(*item*)
Called in itemize whenever an item is generated. Allows for user modifications to the item (e.g. casting). :param item: The unmodified item string. :return: The item after user modification. Defaults to the original, unmodified string.

class `spookyconsole.ntutils.`**`DoubleArrayParamType`**(*enclosing_chars=”*, *require_enclosing_chars=False*, *item_sep_char=’, ’*)

`item_hook`(*item*)
Called in itemize whenever an item is generated. Allows for user modifications to the item (e.g. casting). :param item: The unmodified item string. :return: The item after user modification. Defaults to the original, unmodified string.

class `spookyconsole.ntutils.`**`NTEntryParamType`**

`convert`(*value*, *param*, *ctx*)
Converts the value. This is not invoked for values that are *None* (the missing value).

class spookyconsole.ntutils.**NTPathParamType**

convert (*value*, *param*, *ctx*)
Converts the value. This is not invoked for values that are *None* (the missing value).

class spookyconsole.ntutils.**NTTableParamType**

convert (*value*, *param*, *ctx*)
Converts the value. This is not invoked for values that are *None* (the missing value).

1.4 utils Module

TODO

spookyconsole.utils.**find_nth** (*haystack*, *needle*, *n*)
From StackOverflow.

1.5 gui.core Module

Core functionality for the GUI component of spooky-console.

class spookyconsole.gui.core.**BBox** (*x*, *y*, *w*, *h*)

h
Alias for field number 3

w
Alias for field number 2

x
Alias for field number 0

y
Alias for field number 1

class spookyconsole.gui.core.**Cell** (*column*, *row*)

column
Alias for field number 0

row
Alias for field number 1

class spookyconsole.gui.core.**DockableEntry** (*id*, *cell*)

cell
Alias for field number 1

id
Alias for field number 0

class spookyconsole.gui.core.**DragPoint** (*dockable*, *master=None*, *unbind_parent=True*, **args*, ***kwargs*)

class spookyconsole.gui.core.**Grid**(*master*, *width*, *height*, *cell_width=50*, *cell_height=50*, *column_padding=0*, *row_padding=0*, *resize_protocol=1*, *highlight_visual=VisualSpec(bd_width=2, bd_colour='#A00000'*, *fill='#303030')*, *grid_visual=VisualSpec(bd_width=1*, *bd_colour='#101010'*, *fill='#808080')*, **args*, ***kwargs*)

TODO

dockable_resized(*dockable*)

Signal to this grid that the `DockableMixin` `dockable`'s `DockableMixin.col_span` and/or `DockableMixin.row_span` has been changed, and thus it must be redrawn and its position must be recalculated. If it must be moved, it will be moved as little as possible. If no where on the grid satisfies its new size, the grid is expanded horizontally as needed to accommodate the `dockable`.

Parameters **dockable** (*DockableMixin*) – The dockable that has been resized.

geometry = None

The grid's current `GridGeometry`.

grid_visual = None

The current `VisualSpec` for the grid drawn while a `DockableMixin` is dragged.

highlight_visual = None

The current `VisualSpec` for the highlighting effect.

move_dockable(*dockable*, *cell*)

Move the given `DockableMixin` `dockable` to the given `Cell` `cell`. If there is a grid conflict, this will fail and no changes will be made.

Parameters

- **dockable** (*DockableMixin*) – The dockable to move.
- **cell** (`Cell`) – The cell to move the dockable to.

Returns Whether or not the move was successful.

Return type bool

orig_geometry = None

Stores the geometry (as a `GridGeometry`) before any changes from window resizing are made.

register_dockable(*dockable*)

Register a `DockableMixin` to be managed by this grid. Its tkinter parent must already be this grid.

Registered dockables should never be manually (i.e. `dockable.configure(...)`) positioned, have their width and height manually changed, etcetera. Instead use the appropriate methods on `DockableMixin` objects.

Parameters **dockable** (*DockableMixin*) – The dockable to be managed by this grid.

remove_dockable(*dockable*)

Remove the given `DockableMixin` `dockable` from the grid. This method is functionally the opposite of `Grid._place_dockable`.

Parameters **dockable** (*DockableMixin*) – The dockable to remove.

resize_protocol = None

An integer representing the protocol to employ when the parent window is resized such that the grid's canvas is granted more space than required by its current scrollregion.

May be one of the following:

- `Grid.RESIZE_PROTO_NONE`: Do nothing when the window is resized.

- `Grid.RESIZE_PROTO_EXPAND_CELLS`: Expand each cell equally to fit any new space when the window is resized.
- `Grid.RESIZE_PROTO_ADD_PADDING`: Add padding equally in between each column or row when the window is resized.

`set_geometry`(*width=None*, *height=None*, *cell_width=None*, *cell_height=None*, *column_padding=None*, *row_padding=None*)

Request the grid's geometry (`Grid.geometry`) to be changed.

Note that `cell_width` and `cell_height` have minimums, and, in the event that a requested change to them violates these minimums, they will be clamped greater than the minimums. The minimums are `Grid.MIN_CELL_WIDTH` and `Grid.MIN_CELL_HEIGHT`.

Also, `width` and `height` cannot be changed so as to "clip off" any `DockableMixin``s on the grid. Therefore, once again, the ``width` and `height` will be clamped to be greater than the minimum width and height of the grid state (`GridState.min_width` and `GridState.min_height`).

To check if any of the requested new dimensions have been denied, consult the returned `GridGeometry` object, which contains the geometries actually deployed.

Parameters

- **`width`** (*`int`*) – The new width of the grid, or None for no change.
- **`height`** (*`int`*) – The new height of the grid, or None for no change.
- **`cell_width`** (*`int`*) – The new cell width of the grid, or None for no change.
- **`cell_height`** (*`int`*) – The new cell height of the grid, or None for no change.
- **`column_padding`** (*`int`*) – The new column padding of the grid, or None for no change.
- **`row_padding`** (*`int`*) – The new row padding of the grid, or None for no change.

Returns The new `GridGeometry` object.

Return type *GridGeometry*

`set_grid_visual`(*bd_width=None*, *bd_colour=None*, *fill=None*)

Set one or more of the visual aspects of the grid effect.

Parameters

- **`bd_width`** – The new border width, or None for no change.
- **`bd_colour`** – The new border colour, or None for no change.
- **`fill`** – The new fill colour, or None for no change.

`set_highlight_visual`(*bd_width=None*, *bd_colour=None*, *fill=None*)

Set one or more of the visual aspects of the highlighting effect.

Parameters

- **`bd_width`** – The new border width, or None for no change.
- **`bd_colour`** – The new border colour, or None for no change.
- **`fill`** – The new fill colour, or None for no change.

`set_resize_protocol`(*protocol*)

Set the resize protocol (`Grid.resize_protocol`) to the given one.

Parameters **`protocol`** – The resize protocol to employ.

signal_drag_motion(*mouse_x*, *mouse_y*)

Called by a `DockableMixin` widget currently being dragged to signal that the mouse has moved.

Parameters

- **mouse_x** (*int*) – The mouse's x coordinate relative to the root widget.
- **mouse_y** (*int*) – The mouse's y coordinate relative to the root widget.

signal_drag_start(*dockable*)

Called by the `DockableMixin` dockable widget to signal that it is being dragged.

Parameters **dockable** (*DockableMixin*) – The dockable who's being dragged.

signal_drag_stop()

Called by a `DockableMixin` widget to signal that it has stopped being dragged.

class spookyconsole.gui.core.**GridGeometry**(*width*, *height*, *cell_width*, *cell_height*, *column_padding*, *row_padding*)

cell_height

Alias for field number 3

cell_width

Alias for field number 2

column_padding

Alias for field number 4

height

Alias for field number 1

row_padding

Alias for field number 5

width

Alias for field number 0

class spookyconsole.gui.core.**GridState**

class spookyconsole.gui.core.**Point**(*x*, *y*)

x

Alias for field number 0

y

Alias for field number 1

class spookyconsole.gui.core.**ScrollCanvas**(*master*, *width*, *height*, *bind_all=False*, *scroll_wheel_scale=0.5*, *scroll_press_scale_x=0.5*, *scroll_press_scale_y=0.5*, *scroll_press_delay=50*, **args*, ***kwargs*)

class spookyconsole.gui.core.**Size**(*width*, *height*)

height

Alias for field number 1

width

Alias for field number 0

class `spookyconsole.gui.core.`**`VisualSpec`**(*bd_width*, *bd_colour*, *fill*)

`bd_colour`
Alias for field number 1

`bd_width`
Alias for field number 0

`fill`
Alias for field number 2

class `spookyconsole.gui.core.`**`Window`**(*root*, **args*, ***kwargs*)

1.6 gui.plot Module

TODO

class `spookyconsole.gui.plot.`**`DockablePlot`**(*master*, *row_span=9*, *col_span=9*, **args*, ***kwargs*)

class `spookyconsole.gui.plot.`**`Plot`**(*master*, *nrows=1*, *ncols=1*, *style='ggplot'*, **args*, ***kwargs*)

TODO: some of the "get" calls might have a significant overhead in extreme cases; it may be worth employing a
cache system if performance is an issue.

class `spookyconsole.gui.plot.`**`PlotToolbar`**(*master*, *plot*, **args*, ***kwargs*)

1.7 gui.widgets Module

TODO

class `spookyconsole.gui.widgets.`**`DockableButton`**(*parent_grid*, *col_span*, *row_span*, **args*, ***kwargs*)

class `spookyconsole.gui.widgets.`**`DockableCanvas`**(*parent_grid*, *col_span*, *row_span*, **args*, ***kwargs*)

class `spookyconsole.gui.widgets.`**`DockableCheckbutton`**(*parent_grid*, *col_span*, *row_span*, **args*, ***kwargs*)

class `spookyconsole.gui.widgets.`**`DockableEntry`**(*parent_grid*, *col_span*, *row_span*, **args*, ***kwargs*)

class `spookyconsole.gui.widgets.`**`DockableFrame`**(*parent_grid*, *col_span*, *row_span*, **args*, ***kwargs*)

class `spookyconsole.gui.widgets.`**`DockableLabel`**(*parent_grid*, *col_span*, *row_span*, **args*, ***kwargs*)

class `spookyconsole.gui.widgets.`**`DockableLabelFrame`**(*parent_grid*, *col_span*, *row_span*, **args*, ***kwargs*)

class `spookyconsole.gui.widgets.`**`DockableListbox`**(*parent_grid*, *col_span*, *row_span*, **args*, ***kwargs*)

class `spookyconsole.gui.widgets.`**`DockableRadiobutton`**(*parent_grid*, *col_span*, *row_span*, **args*, ***kwargs*)

class `spookyconsole.gui.widgets.`**`DockableScale`**(*parent_grid*, *col_span*, *row_span*, **args*, ***kwargs*)

class spookyconsole.gui.widgets.**DockableSpinbox**(*parent_grid*, *col_span*, *row_span*, **args*, ***kwargs*)

class spookyconsole.gui.widgets.**DockableText**(*parent_grid*, *col_span*, *row_span*, **args*, ***kwargs*)

1.8 commands.generic Module

TODO

1.9 commands.networktables Module

TODO

class spookyconsole.commands.networktables.**ListElement**(*is_entry*, *key*, *parent*)

is_entry
Alias for field number 0

key
Alias for field number 1

parent
Alias for field number 2

PYTHON MODULE INDEX