

RESTAPI

REST API Proposal

JSON Data Structures

Page

```
{
  id: "",
  name: "",
  ownerId: "",
  layoutId: "",
  type: "",
  subPages: [ {..full page object..} ],
  regions: [
    {
      id: "",
      locked: true/false,
      regionWidgets: [
        {
          id: "",
          name: "",
          widgetId: "",
          collapsed: true/false,
          locked: true/false,
          chrome: true/false,
          preferences: [
            {
              name: "",
              value: ""
            }
          ]
        }
      ]
    }
  ]
}
```

Notes on Page object:

- Region render order is driven by the order of the region is in the list
- RegionWidget name is an override to the widget name it's based on
- RegionWidget render position is driven by the order the region widget is in the list
- RegionWidget render order has been removed - no use case for being able to render the 3rd widget in the region prior to the 1st

REST Methods

As a general rule if you update an object that has nested objects as values (i.e. page), if you don't specify the key for the nested object (i.e. regions) then the regions would not be modified at all. If you do specify the key, then the values as part of that key will replace the entire sub-tree. Modifying sub items is much simpler by using the subresource paths under the parent object (i.e. pages/{id}/regions/...)

Pages

```
/pages - GET/POST - all pages that don't have parent pages
/pages/{id} - GET/PUT/DELETE - single page that doesn't have parent page
/pages/{id}/layout - GET - page layout object
/pages/{id}/regions - GET/POST - all regions
/pages/{id}/regions/{id} - GET/PUT/DELETE - single region
/pages/{id}/regions/{id}/position - GET/PUT - gets/sets the position the region is in the page
/pages/{id}/regions/{id}/regionwidgets - GET/POST - all the region widgets
/pages/{id}/regions/{id}/regionwidgets/{id} - GET/PUT/DELETE - single region widget
/pages/{id}/regions/{id}/regionwidgets/{id}/position - GET/PUT - gets/sets the position the regionwidget is in
the page
/pages/{id}/regions/{id}/regionwidgets/{id}/widget - GET - widget object referenced by this region widget
/pages/{id}/regions/{id}/regionwidgets/{id}/preferences - GET/POST/PUT - preferences
/pages/{id}/subpages - GET/POST - all sub pages
/pages/{id}/subpages/{id} - GET/PUT/DELETE - single sub page (how many levels of sub pages are allowed? We can
support unlimited but don't know if that's realistic)
/pages/{id}/subpages/{id}/layout - GET
/pages/{id}/subpages/{id}/regions - GET/POST
/pages/{id}/subpages/{id}/regions/{id} - GET/PUT/DELETE - single region
/pages/{id}/subpages/{id}/regions/{id}/position - GET/PUT - gets/sets the position the region is in the page
/pages/{id}/subpages/{id}/regions/{id}/regionwidgets - GET/POST - all the region widgets
/pages/{id}/subpages/{id}/regions/{id}/regionwidgets/{id} - GET/PUT/DELETE - single region widget
/pages/{id}/subpages/{id}/regions/{id}/regionwidgets/{id}/position - GET/PUT - gets/sets the position the
regionwidget is in the page
/pages/{id}/subpages/{id}/regions/{id}/regionwidgets/{id}/widget - GET - widget object referenced by this
region widget
/pages/{id}/subpages/{id}/regions/{id}/regionwidgets/{id}/preferences - GET/POST/PUT - preferences
```