# Sidebar Widgets

XperienCentral widgets are plugins that contain a widget. A widget is an "element of interaction" in a user interface. Basic widgets can be buttons, input fields checkboxes and so forth. Widgets can be combined together to form more complex functionality, for example  a widget consisting of multiple input fields, buttons and a radio box. The user interface in XperienCentral is written in JavaScript which makes use of a framework called the Dojo Toolkit, usually referred to as simply Dojo. Dojo defines a way of implementing these kinds of compound widgets using browser functionality such as HTML, CSS and JavaScript. Except for standard HTML tags, other widgets can be added in a similar fashion to build up new client-side components from the bottom up. The widgets used in a component can be standard Dojo widgets, 3rd party widgets, XperienCentral widgets or custom widgets. Dojo-based widgets are used throughout the XperienCentral user interface.

It is also possible to write custom code using this technique. The most common way to use widgets in custom code in XperienCentral is by adding new Sidebar widgets. It is also possible to create new interactive maintenance panels. A Sidebar widget is a visible component which can be docked in the left or right-hand Sidebar in the XperienCentral Workspace.

This section contains a topic on creating Sidebar widgets by beginning with an archetype which is provided as a starting point for development. The code generated by the archetype is explained in detail. The generated code is explained, including the Dojo toolkit basics, which will help you get started with your own development.

### Other pages in this topic

- Developing Sidebar Widgets

### Related pages

- Plugins
- Archetypes

### External related pages

- The Dojo Toolkit
- Dojo documentation

---

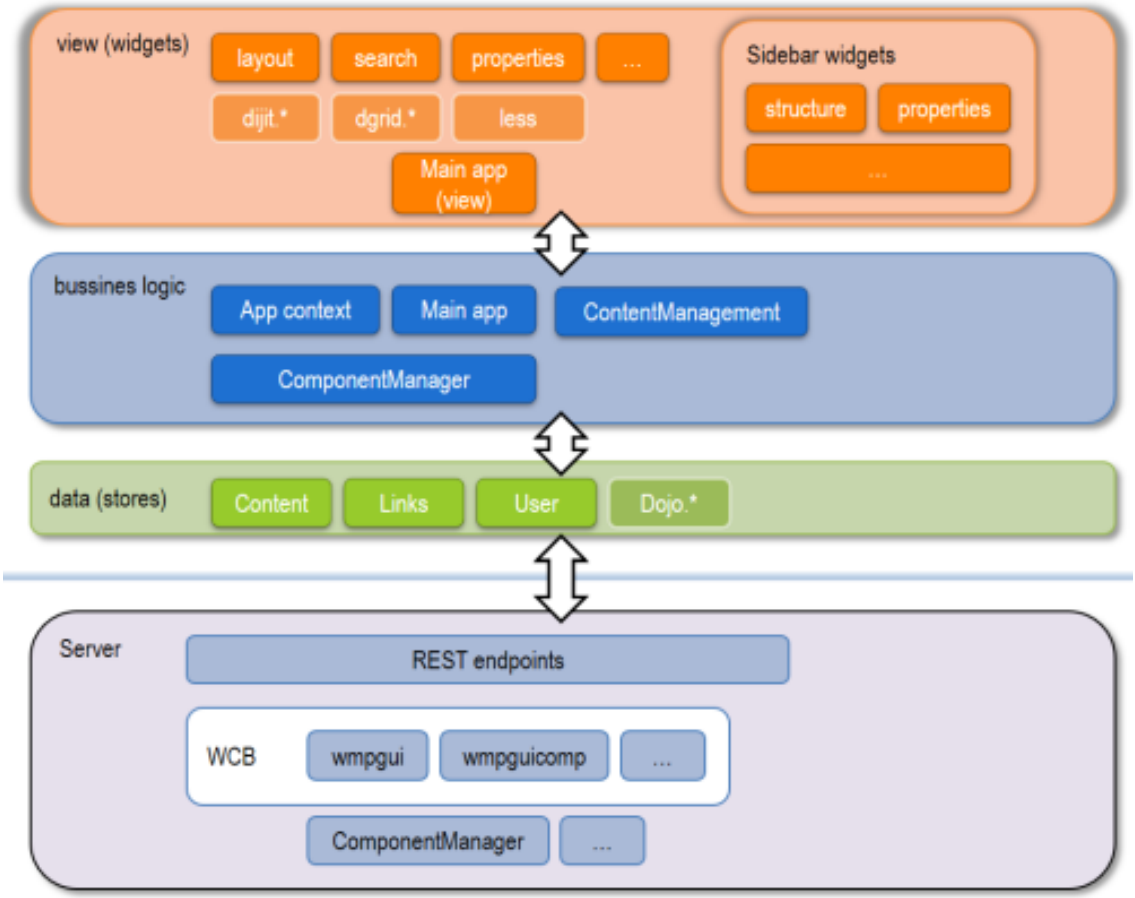
### In This Topic



---

## XperienCentral Client-side Architecture

The client-side architecture introduced in XperienCentral makes it possible to add functionality to the Workspace using Dojo widgets. The advantave of this architecture is that you can add functionality which reacts to changes in the Workspace and Editor, such as when starting editing or viewing content. The figure below shows a depiction of the XperienCentral client-side architecture.




From top to bottom:

- The user interface of XperienCentral consisting of Dojo widgets.
- The business logic classes provide an API for managing things such as content (the client-side API).
- The communication between the client (browser) and the server is streamlined using Dojo data stores.
- The client-side implementation is backed by server-side REST endpoints which provide access to content in XperienCentral.

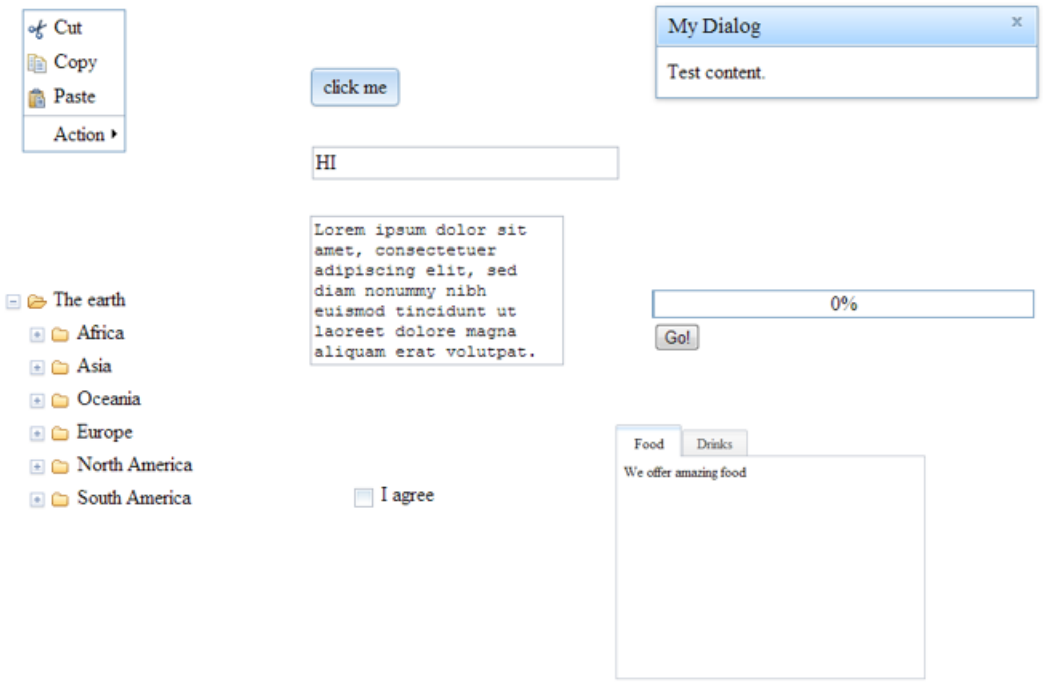
**External references:** Dojo / Data Stores in DOJO

Back to top

---

## The Dojo Toolkit

The Dojo toolkit is an open source modular JavaScript library, or, more specifically a JavaScript toolkit which is designed for the rapid development of cross-platform, JavaScript/AJAX-based applications and websites. The following are examples of Dojo widgets you can develop:




### Documentation

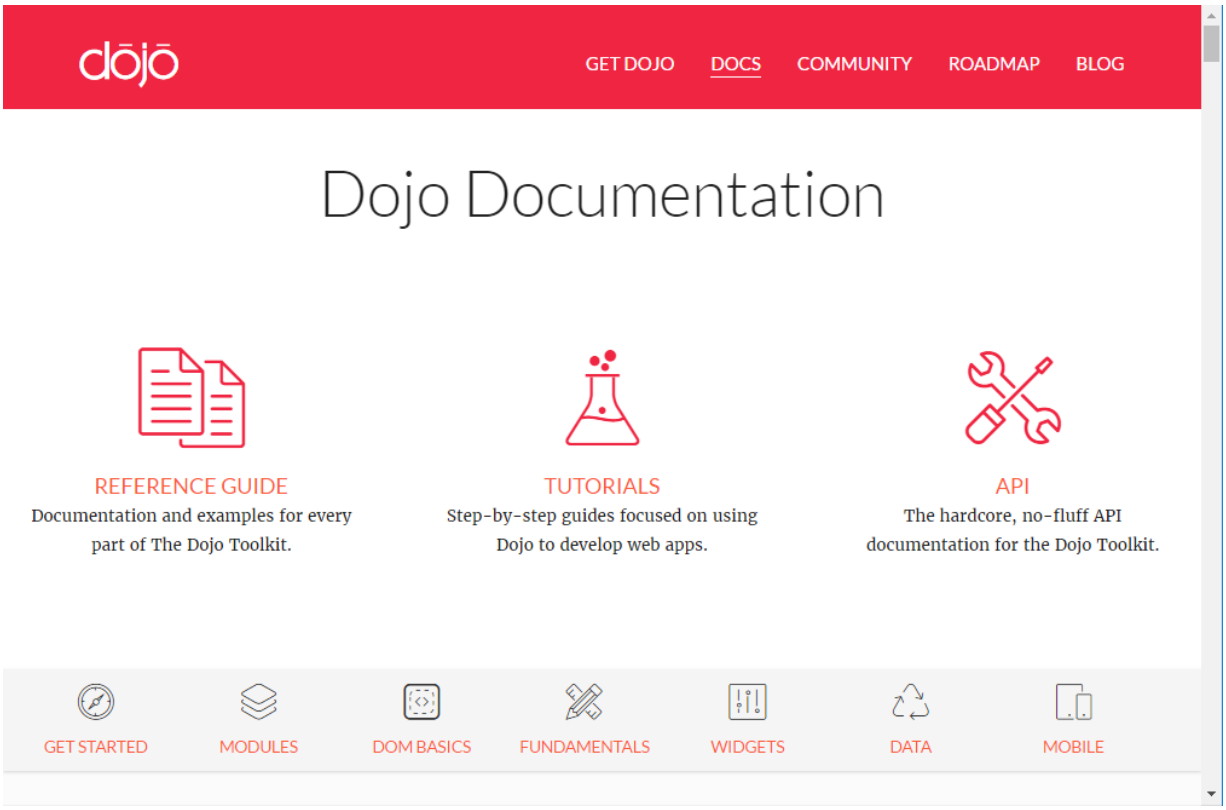
- Menu
- Button
- TextBox
- Dialog (ConfirmDialog)
- Tree
- TextArea widget
- CheckBox
- ProgressBar
- TabContainer

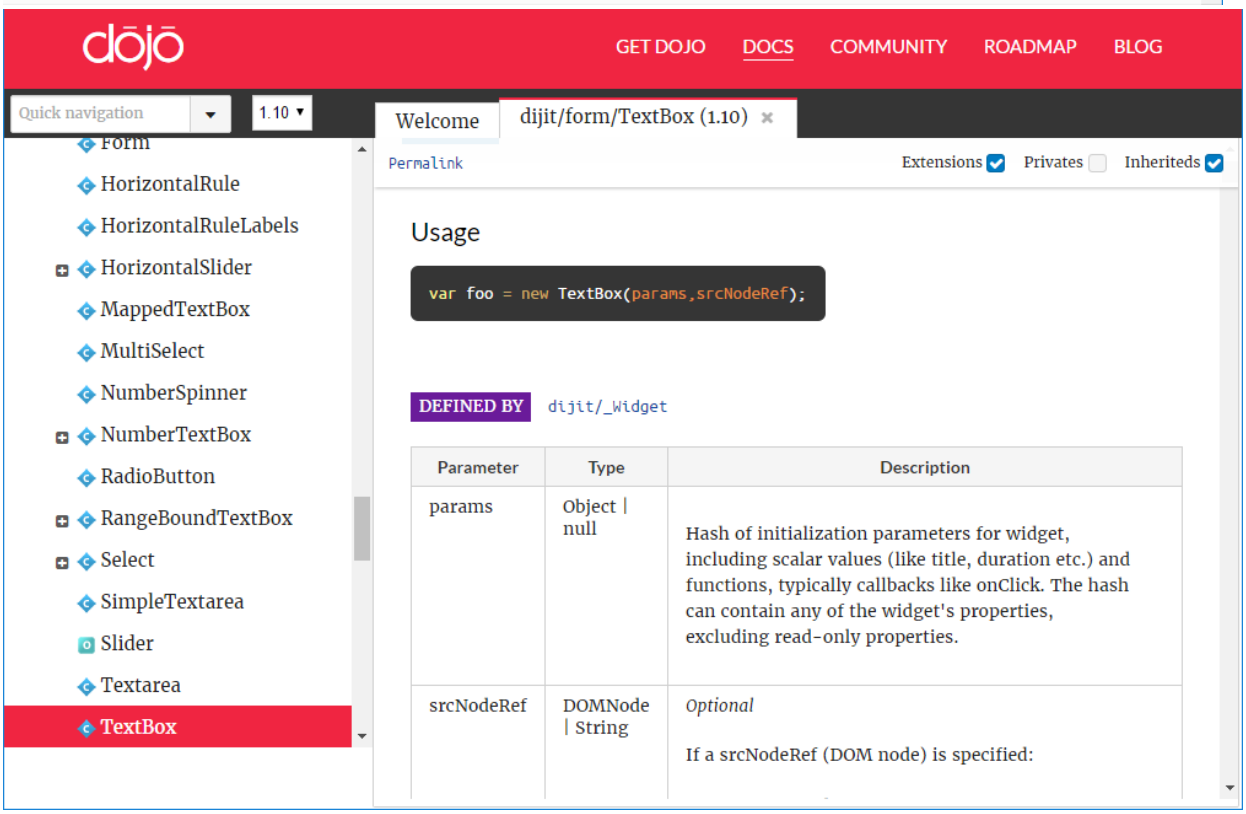
## Dojo Documentation

There is a lot of documentation available on Dojo. If you're not familiar with the Dojo toolkit, have a look at the tutorials. Another approach is to dive right into it by creating a sidebar widget based on the archetype and analyze the code using using the information available in the Developing Sidebar Widgets topic.

**References:**

- Website of the Dojo foundation: http://dojotoolkit.org
- Dojo documentation: http://dojotoolkit.org/documentation
- The Dojo API: http://dojotoolkit.org/api/

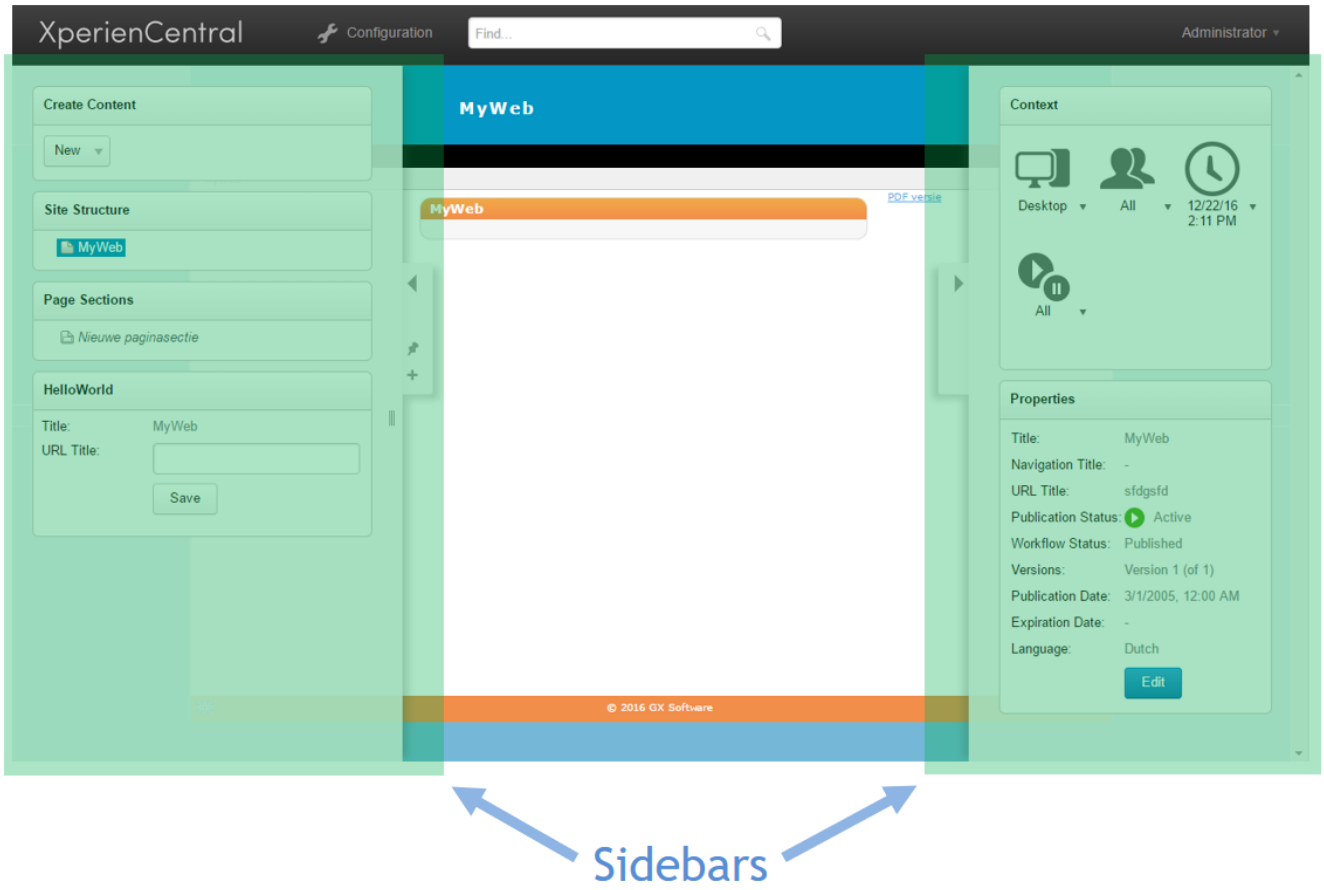

dojo
GET DOJO
DOCS
COMMUNITY
ROADMAP
BLOG
Dojo Documentation
REFERENCE GUIDE
Documentation and examples for every part of The Dojo Toolkit.
TUTORIALS
Step-by-step guides focused on using Dojo to develop web apps.
API
The hardcore, no-fluff API documentation for the Dojo Toolkit.
GET STARTED
MODULES
DOM BASICS
FUNDAMENTALS
WIDGETS
DATA
MOBILE


dojo
GET DOJO
DOCS
COMMUNITY
ROADMAP
BLOG
Quick navigation
1.10
Welcome
dijit/form/TextBox (1.10)
Permalink
Extensions
Privates
Inheriteds
Form
HorizontalRule
HorizontalRuleLabels
HorizontalSlider
MappedTextBox
MultiSelect
NumberSpinner
NumberTextBox
RadioButton
RangeBoundTextBox
Select
SimpleTextarea
Slider
Textarea
TextBox
Usage
var foo = new TextBox(params,srcNodeRef);
DEFINED BY
dijit/_Widget
Parameter
Type
Description
params
Object | null
Hash of initialization parameters for widget, including scalar values (like title, duration etc.) and functions, typically callbacks like onClick. The hash can contain any of the widget's properties, excluding read-only properties.
srcNodeRef
DOMNode | String
Optional
If a srcNodeRef (DOM node) is specified:

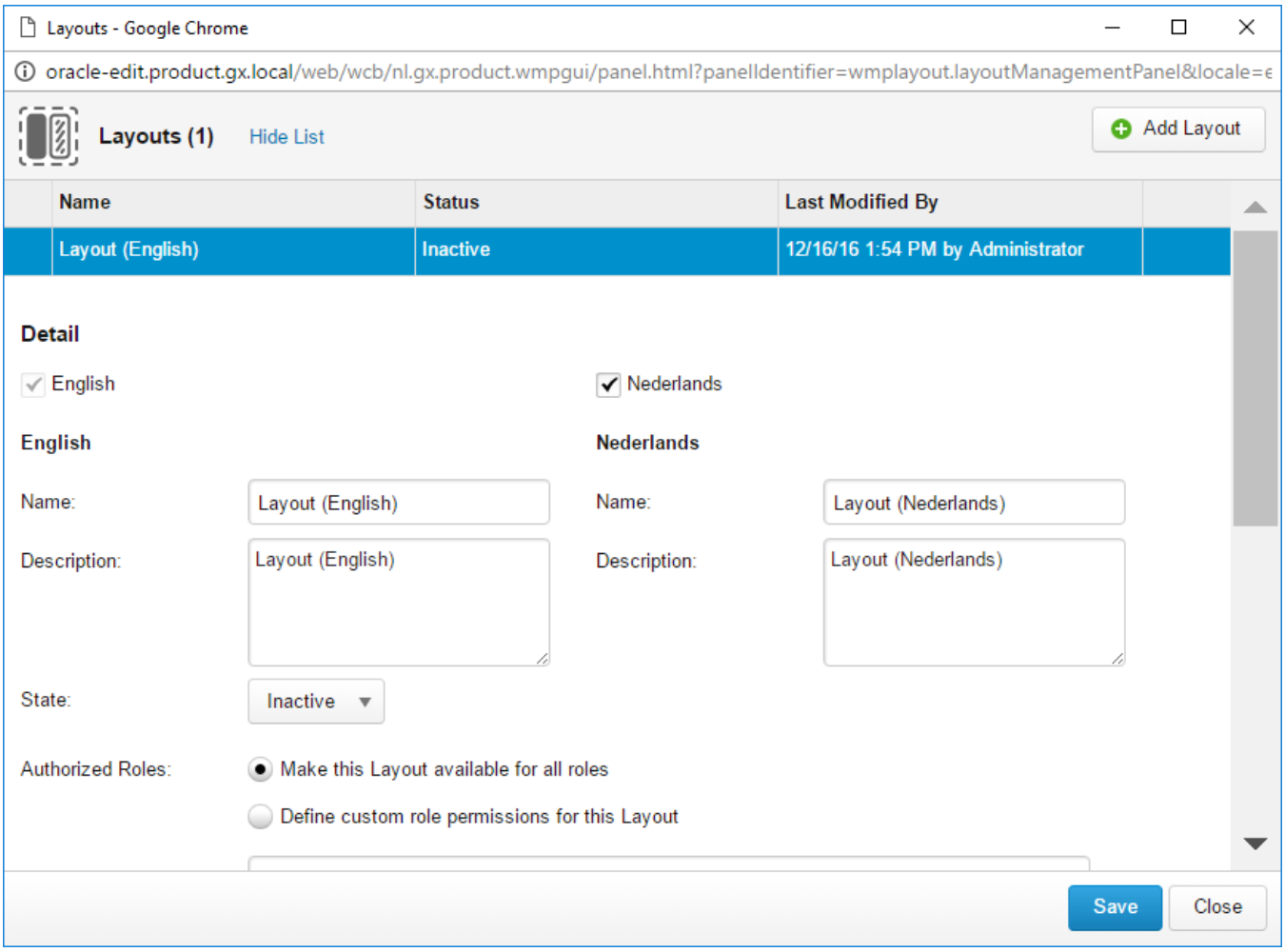
Back to top

# Widget Plugins in XperienCentral

There are two types of widget plugins in XperienCentral: Sidebar widgets and panel widgets. Sidebar widgets are widgets that can be docked in either the left or right Sidebar in the XperienCentral Workspace:




Sidebar widgets are described in detail in the Developing Sidebar Widgets topic, including how to create one using an archetype. Panel widgets are widgets that can be opened via a menu item in the Configuration menu, such as the Layout panel:




At this time there is no archetype or specific documentation for creating custom panel widgets. Contact your GX Software consultant for help in creating a new panel widget.

Back to top

---

## Dojo Widget Composition

Keep the following in mind when planning the composition of Dojo widgets:

- Make sure that each widget is self-contained.
- The widget should expose public properties, methods and events. This ensures that your widget can be reused in other widgets.
- Divide responsibility ("granularity"). In general it is better to create a few smaller widgets with a specific function than one large widget with lots of user interface elements.
- Implement the "deepest" widgets first and work upwards towards the final functionality.
- Avoid using experimental widgets as much as possible. The DojoX package was originally intended for this purpose.
- Check for existing widgets. Don't reinvent the wheel, re-use as much as possible.
- Keep reusability in mind. Make sure your widgets can be reused in other widgets later on. Also write clear API documentation for your own widget.

Back to top