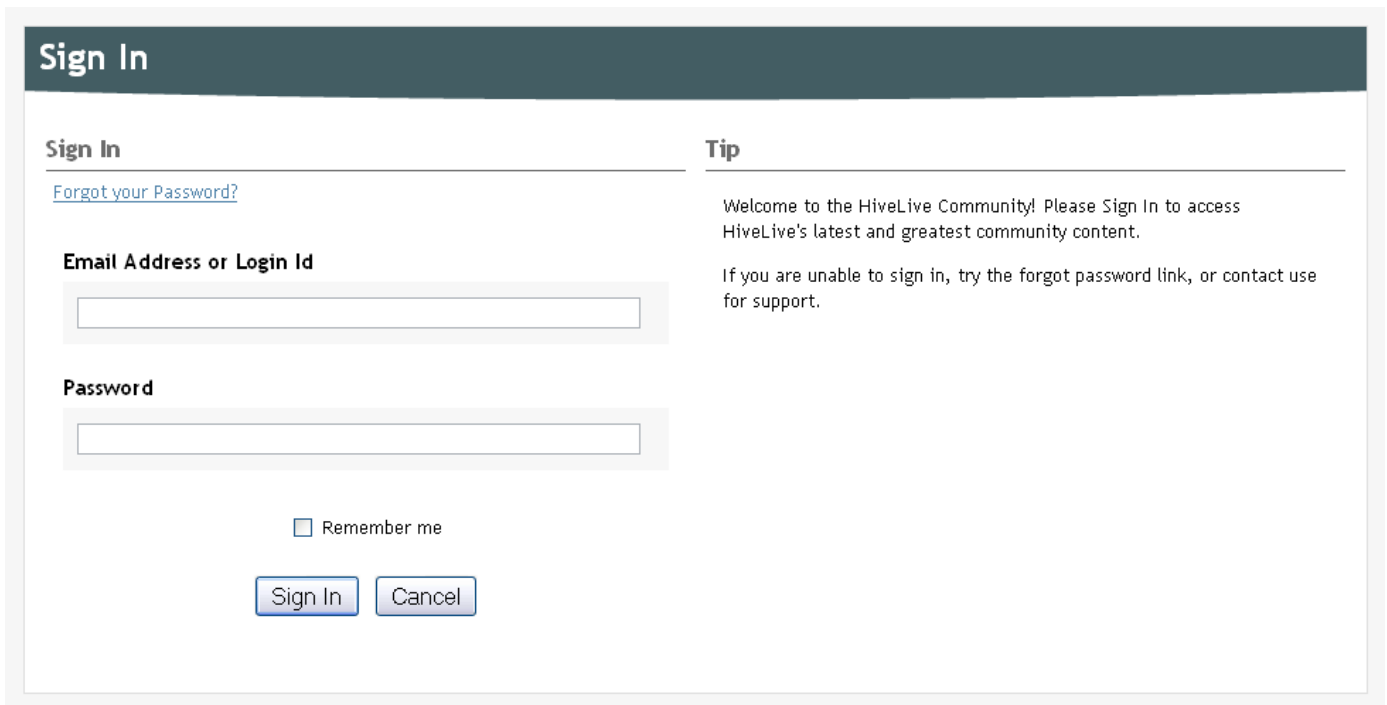


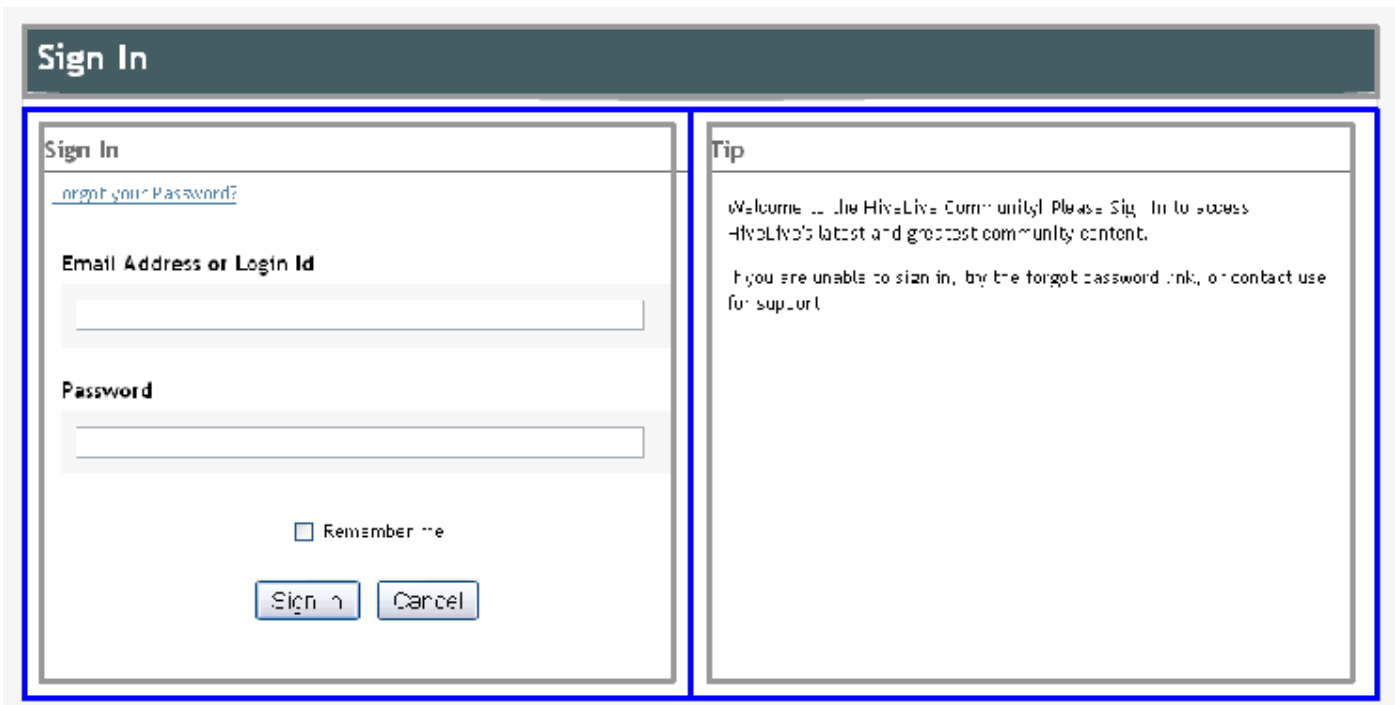
# Pages



The image shows a simplified 'Sign In' page layout. It features a dark blue header with the text 'Sign In'. Below the header, the page is divided into two main sections. The left section, titled 'Sign In', contains a link for 'Forgot your Password?', a label 'Email Address or Login Id' above a text input field, a label 'Password' above another text input field, a checkbox labeled 'Remember me', and two buttons: 'Sign In' and 'Cancel'. The right section, titled 'Tip', contains two paragraphs of text: 'Welcome to the HiveLive Community! Please Sign In to access HiveLive's latest and greatest community content.' and 'If you are unable to sign in, try the forgot password link, or contact use for support.'

*Figure 1: Sign In Page (simplified)*

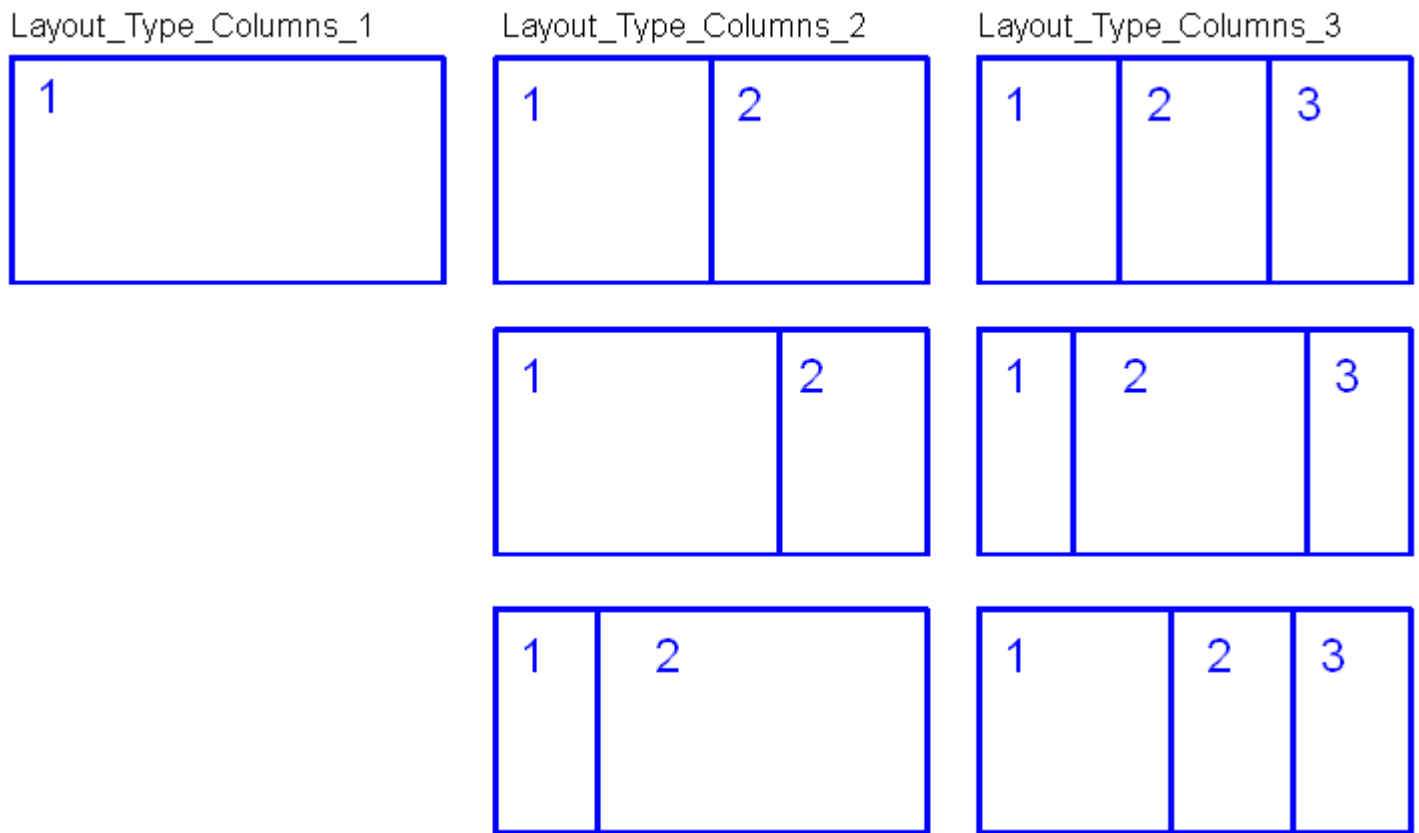
- As a User in the HiveLive LiveConnect Community Platform, everything you interact with is a **Page**
- All Pages are built with two main building blocks: **Layouts** and **Widgets**
- Figure 1 shows a simplified version of the Sign In Page. Figure 2 highlights its 3 Widgets and 1 Layout.
- The **gray** lines show the 3 Widgets, while the **blue** lines show the Two Column Layout



This image is identical to Figure 1, but it highlights the underlying structure of the page. A thick blue border outlines the entire page content area, representing the 'Two Column Layout'. Three gray borders outline individual components: the 'Sign In' form on the left, the 'Tip' text box on the right, and the 'Forgot your Password?' link area at the top left of the form. These gray-bordered areas represent the 'Widgets' used to build the page.

*Figure 2: Sign In Page with Widgets and Layout*

## Layouts



*Figure 3: Example Layout Types*

- Layouts are containers for Widgets that define one or more **Frames**
- Figure 3 shows example Layout Types
- Column Layouts stack Frames horizontally and call them Columns
- Layouts with multiple Columns have variations which define the dimensions of the Columns
- Row Layouts are not supported as vertical stacking happens automatically

## Widgets

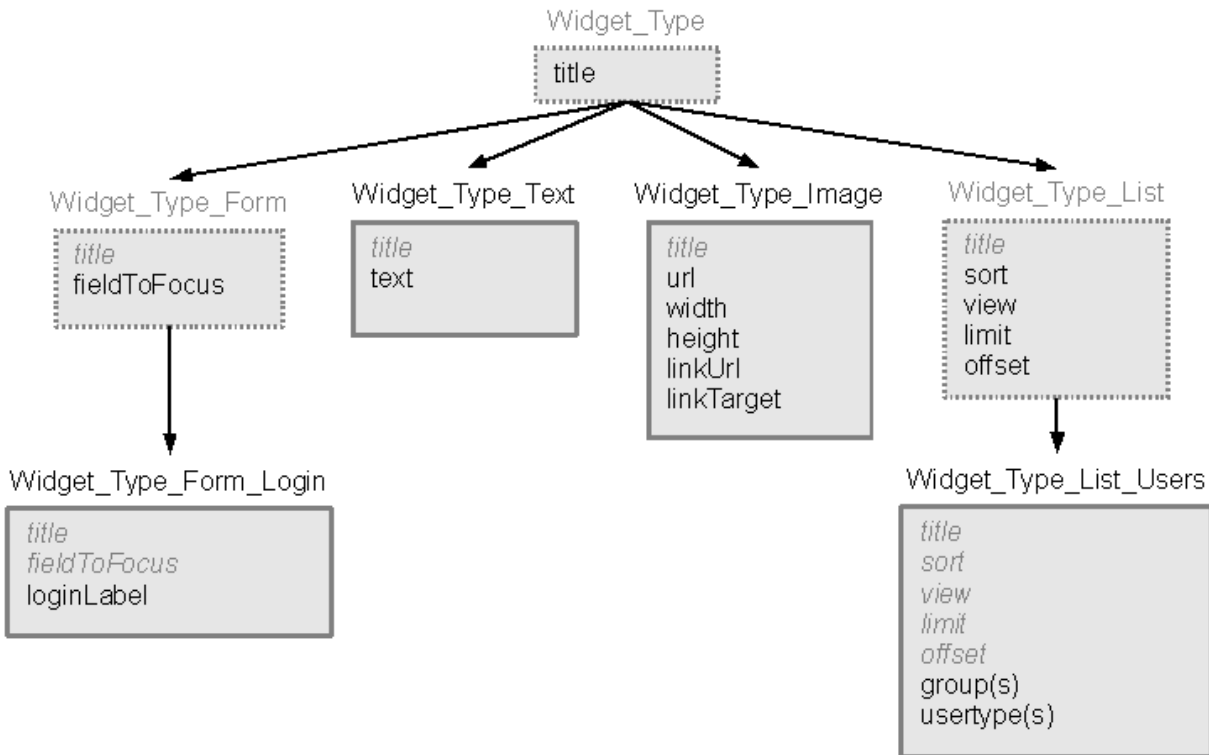


Figure 4: Widget Type Tree (example)

- All content on all Pages exists in Widgets
- Widgets are self-contained pieces of content
- Widget Types have fields and a corresponding form for editing them
- Widget instances know how to render themselves based on their field values
- Widget Types are hierarchical, and they inherit common fields from their ancestors
- Figure 4 shows a small portion of the Widget Type Tree
  - Widget Types with dotted borders are abstract and define common fields
  - Those with solid borders are proper Widget Types that support instances
- Figure 5 shows example Widget instances from the Sign In Page
  - Widget Fields are optional, so not every Widget instance will have a value for every field

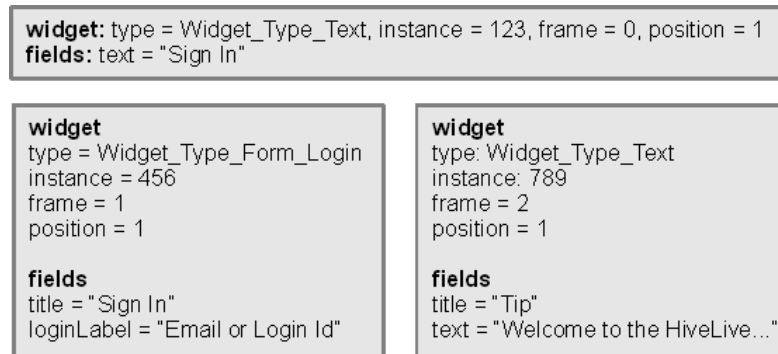
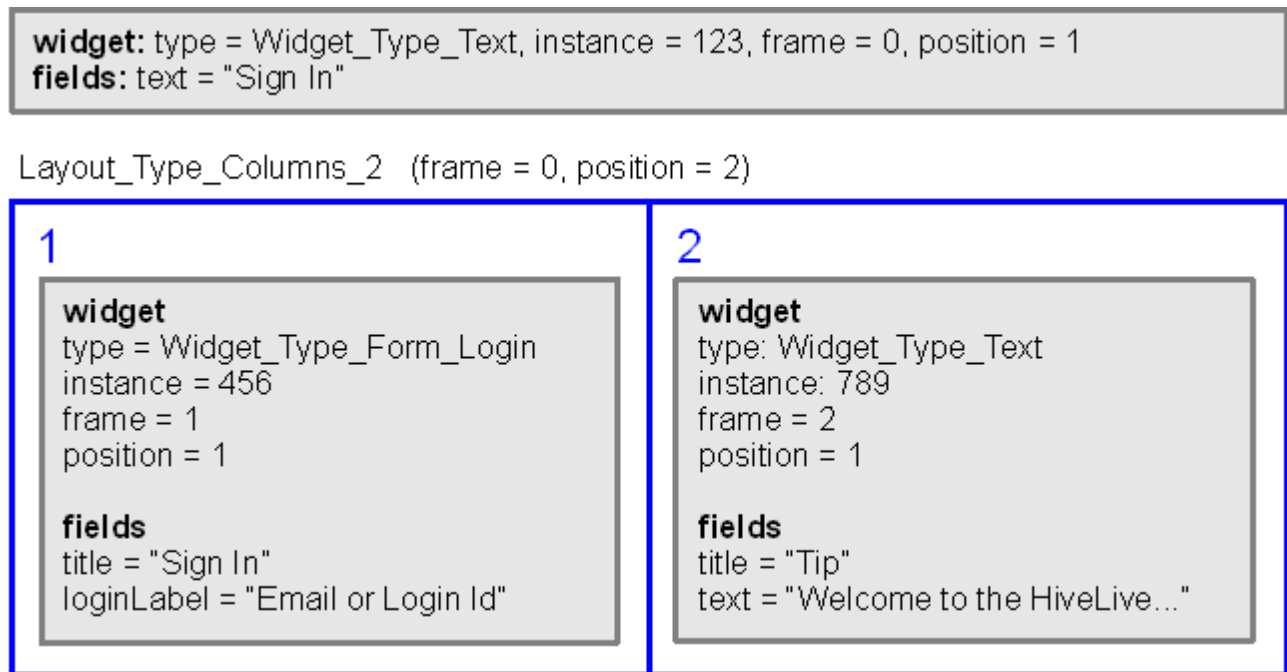


Figure 5: Example Widget Instances

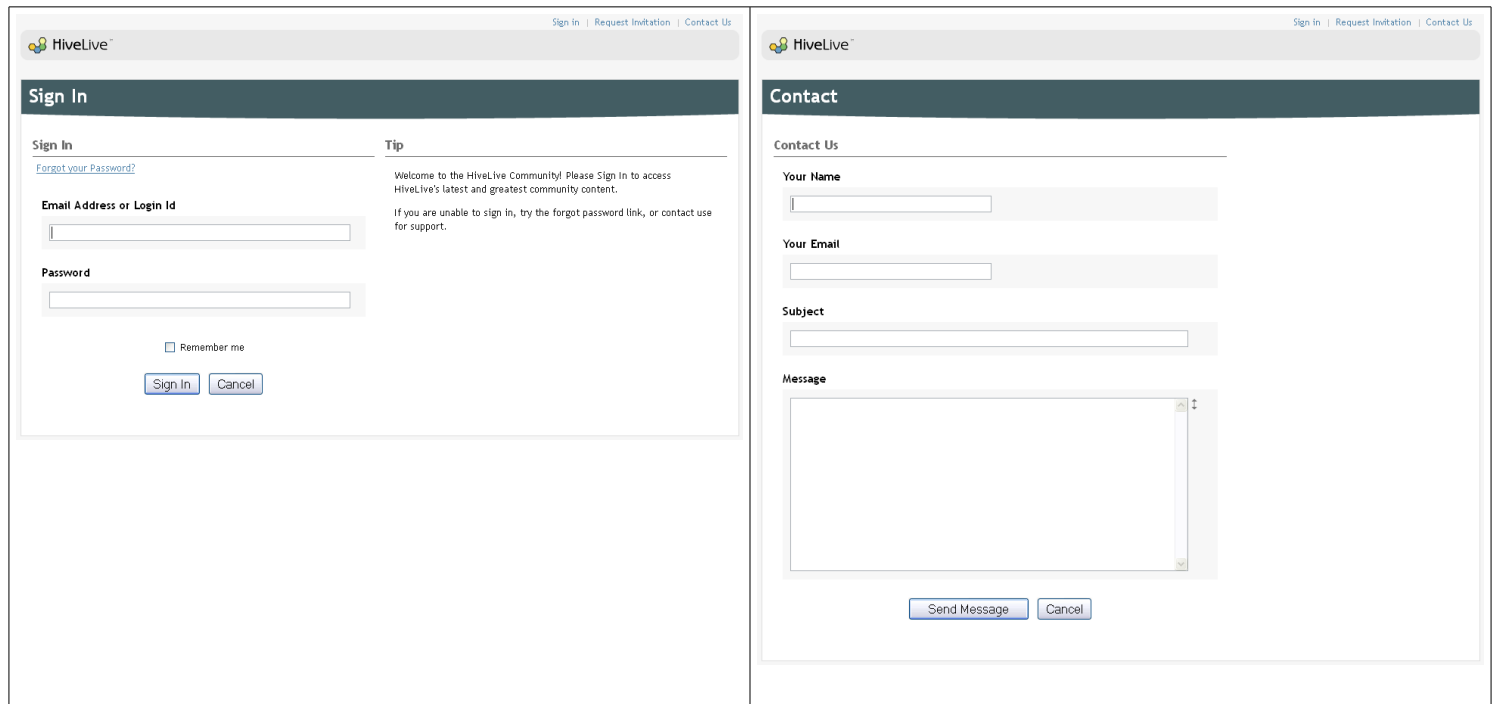
## Pages = Layouts + Widgets



*Figure 6: Sign In Page Layouts and Widgets*

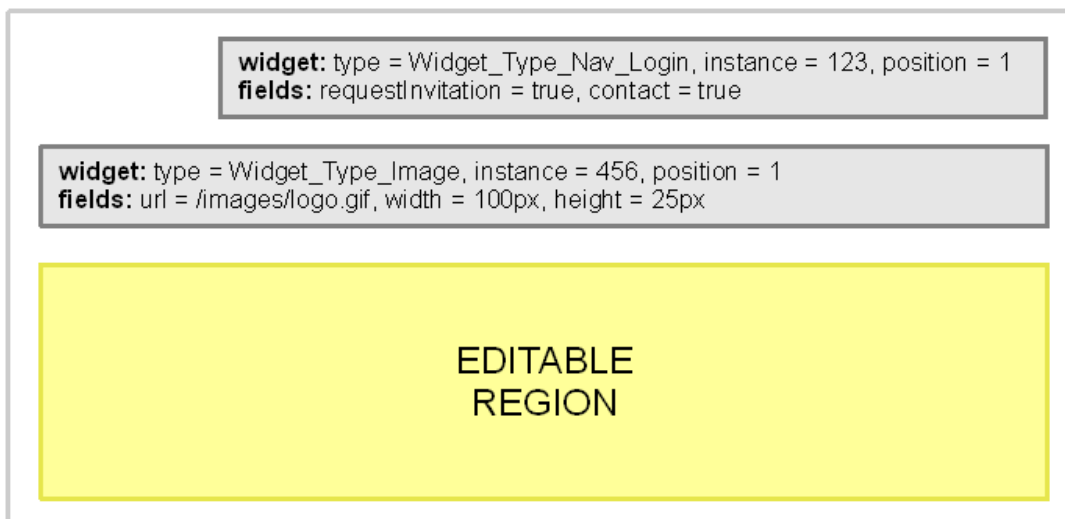
- Figure 6 shows the 3 Widgets and 1 Layout used on the Sign In Page
- The top Widget is a single Widget instance of Widget\_Type\_Text
  - Widget\_Type\_Text is a generic Widget Type for displaying arbitrary text
  - All Widget instances store:
    - 1) their Widget Type
    - 2) a unique instance id
    - 3) the Layout Frame it exists in (0 for topmost)
    - 4) a position in its Frame
  - Widget instances also contain field values for some or all of its Type's fields
- The next item is an instance of Layout\_Type\_Columns\_2
  - Layout Frames can contain any number of Widgets or other Layouts
  - Column 1 contains an instance of Widget\_Type\_Form\_Login
    - Widget\_Type\_Form\_Login will likely only exist on the Sign In Page
    - In fact, an instance of Widget\_Type\_Form\_Login must exist on the Sign In Page. Else, it wouldn't be the Sign In Page.
  - Column 2 optionally contains an instance of Widget\_Type\_Text for providing a Tip for signing in
    - Even in the simple Sign In Page example, there are two instances of Widget\_Type\_Text
    - Some Widget Types (Text, Image, List, Nav, etc) will be used a lot, while other Widgets (Login Form, etc) will likely only be used once

# Templates



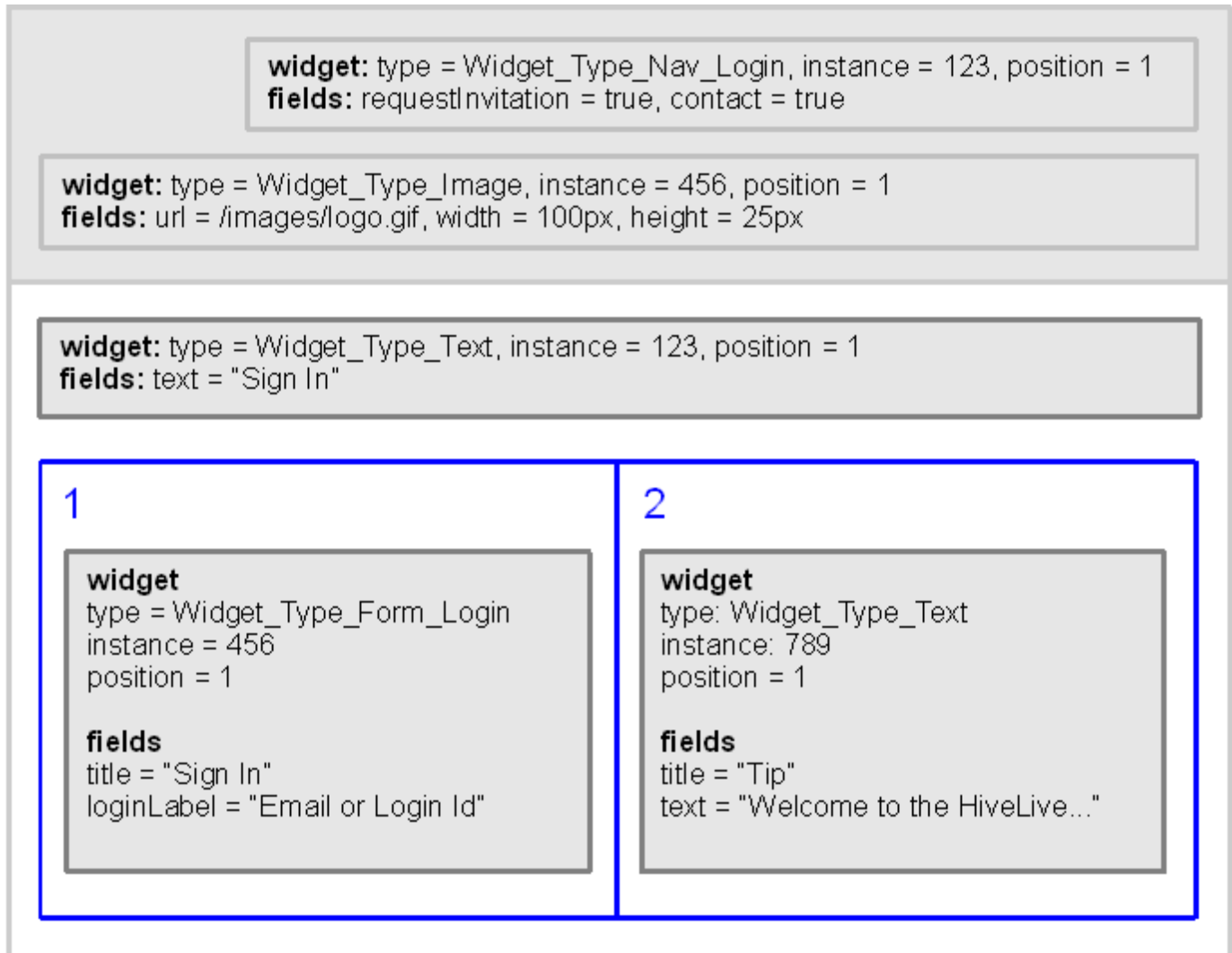
*Figure 7: Common Widgets on Sign In and Contact Pages*

- Templates help eliminate redundancy when two or more Pages share common Widgets
- Figure 7 shows how the Sign In and Contact Pages share the top two Widgets
  - The links in the upper right are an instance of `Widget_Type_Nav_Login`
  - The logo is an instance of `Widget_Type_Image`
- To avoid defining identical Widgets in two places, both Pages can use a common Template, which defines the common Widgets only once
- Figure 8 shows what such a Template could look like
  - Templates can contain Widgets and Layouts, all of which will be inherited by Pages that use them
  - The bottom yellow box is a special Editable Region Layout that defines where Pages can add content



*Figure 8: Community Template*

Pages = Templates + Layouts + Widgets



*Figure 9: Sign In Page using the Community Template*

- Figure 9 shows the Sign In Page with the top two Widgets inherited from its Template
- Every Page must have a Template
- Unlike Templates, Pages cannot define Editable Regions

## Nested Templates

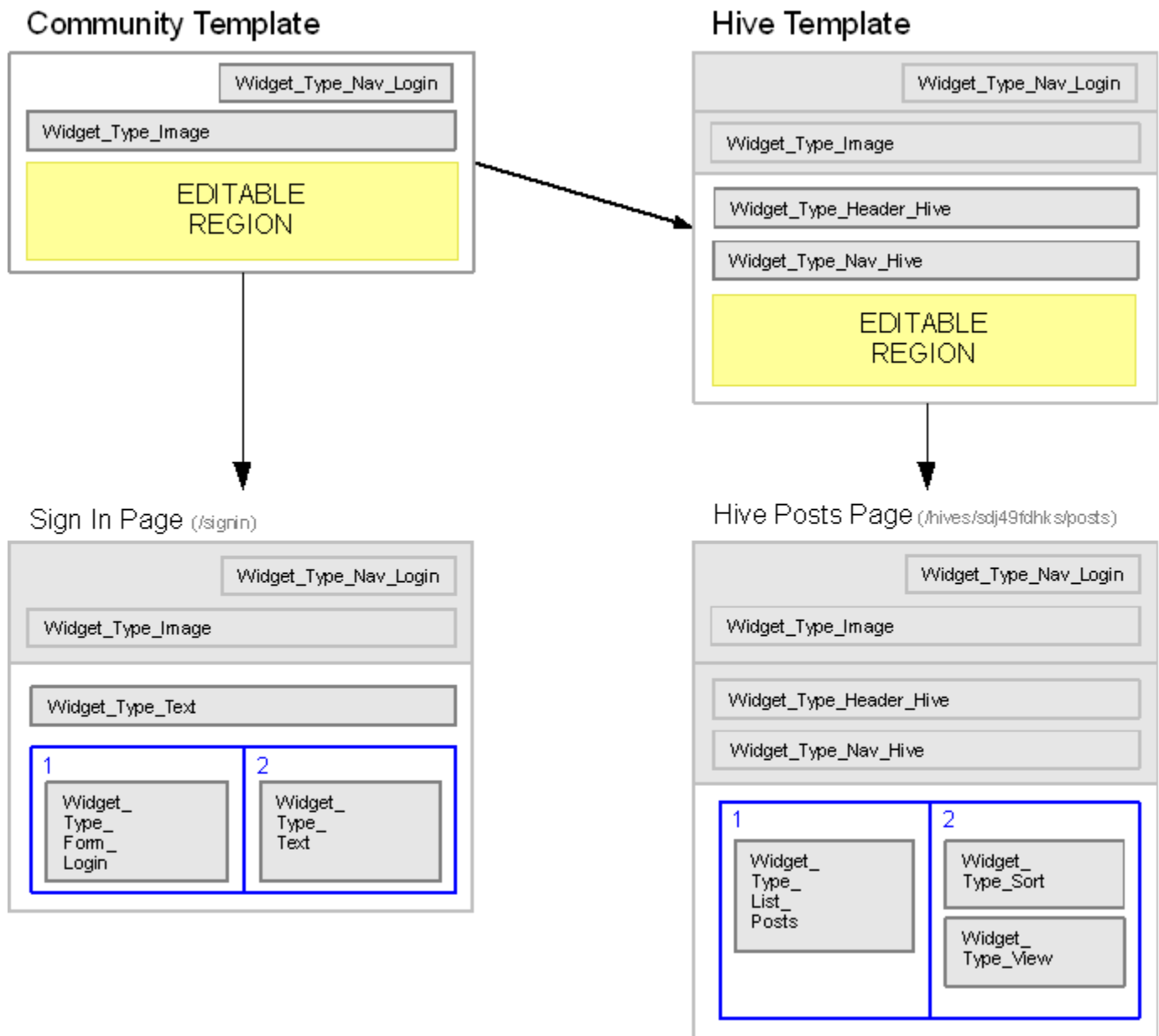


Figure 10: Nested Templates

- Templates can be nested indefinitely
- All Templates must define one or more Editable Regions
- Nested Templates (like Pages) can only add content in Editable Regions of their parent Templates
- Templates without parents are called Master Templates, of which there may be many
- Figure 10 shows 1 Master Template, 1 Nested Template and 1 Page that uses each