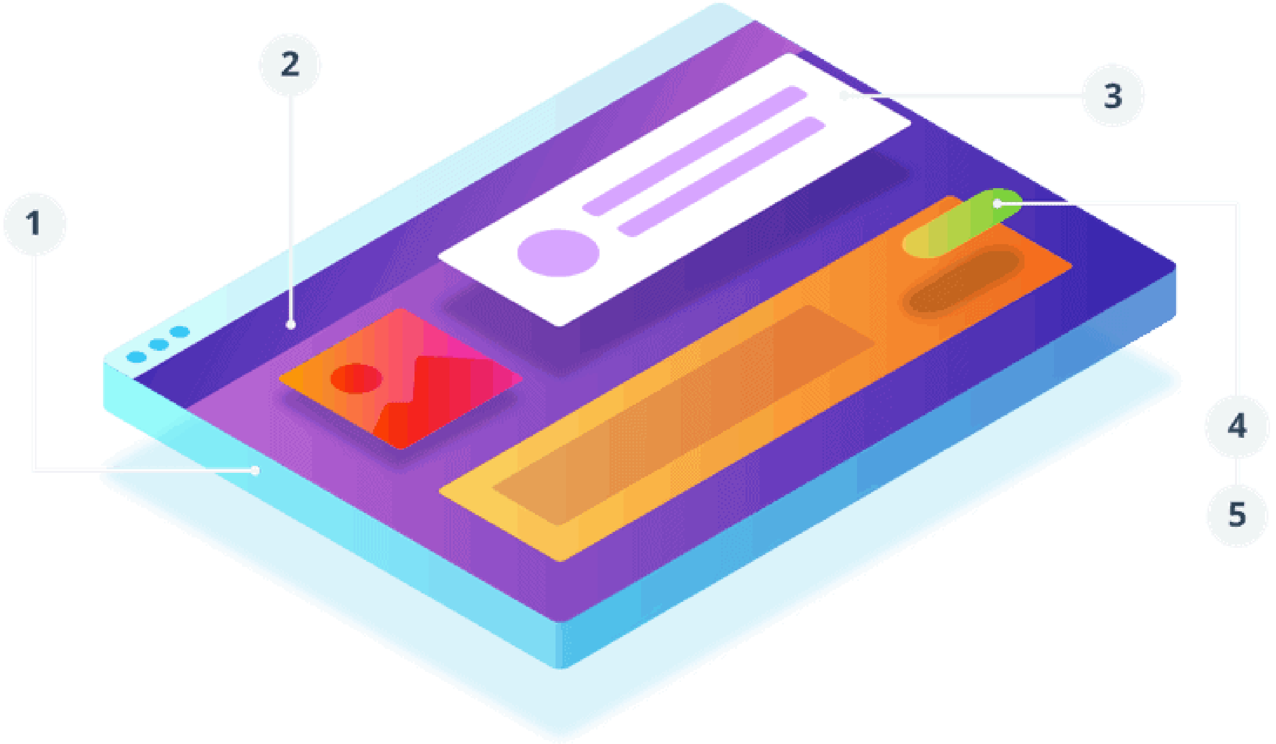
UI Design in Mendix - Widgets, Navigation Layouts, Page Templates

Skip to main content

UI Design

Table of contents



How Can I Design Rich User Interfaces?

User interfaces are built via the combination of a drag-and-drop interface with a WYSIWYG page editor. Our Mendix Studio Pro environment can be used with the Atlas UI framework, which is a fully responsive, open-source, and extensible design language that facilitates a design-thinking approach to building user interfaces.

Common page patterns such as searchable lists, data views, dashboards, and wizards are made available as starter templates that help accelerate development and ensure a consistent, standard user interface across multiple devices.

- **Navigation Layouts**

  When building a Mendix app, the first thing you do is choose a navigation layout. These layouts are the frame within which your dynamic pages are housed, and they provide a consistent structure throughout your app.

Navigation layouts are built using the platform layout editor, which allows you to create the layout patterns needed for your pages.

Layouts use specific controls—such as layout grids, content placeholders, scrolling containers, sliding menus bars, and snippets—to centrally define the required page layout patterns that will be consumed when building the your application's UI.

- ## Page Templates

  Page templates are pre-designed collections of building blocks that can be used as-is or enhanced with custom building blocks and widgets.

- ## Building Blocks

  Building blocks are single-purpose UI elements comprised of multiple widgets and styling. Multiple building blocks are usually used together on one page. You can drag a building block into any of the respective mobile, tablet, or responsive pages to implement common UI patterns, which substantially speeds up the development of custom pages.

- ## Widgets

  Widgets are small user interface elements (for example, alerts, buttons, charts) used to enhance existing building blocks.

  Mendix provides many widgets out of the box, and there are hundreds of open-source widgets available in the Mendix Marketplace. You can also build your own widgets using JavaScript, as discussed in the section How Can I Extend the Mendix Front-End? of *Front-End*.

- ## Design Properties

  You can further customize widgets by changing their design properties. Colors, text, and many other variables can be altered to customize widgets on a case-by-case basis if the styling or placement are not quite right. Theme developers can implement their own (company) design properties so that other users can easily change the look and feel of widgets to match the company's brand.

What UI Elements Does Mendix Provide Out of the Box?

Mendix provides a growing list of out-of-the-box widgets. This is a selection of the most common ones:

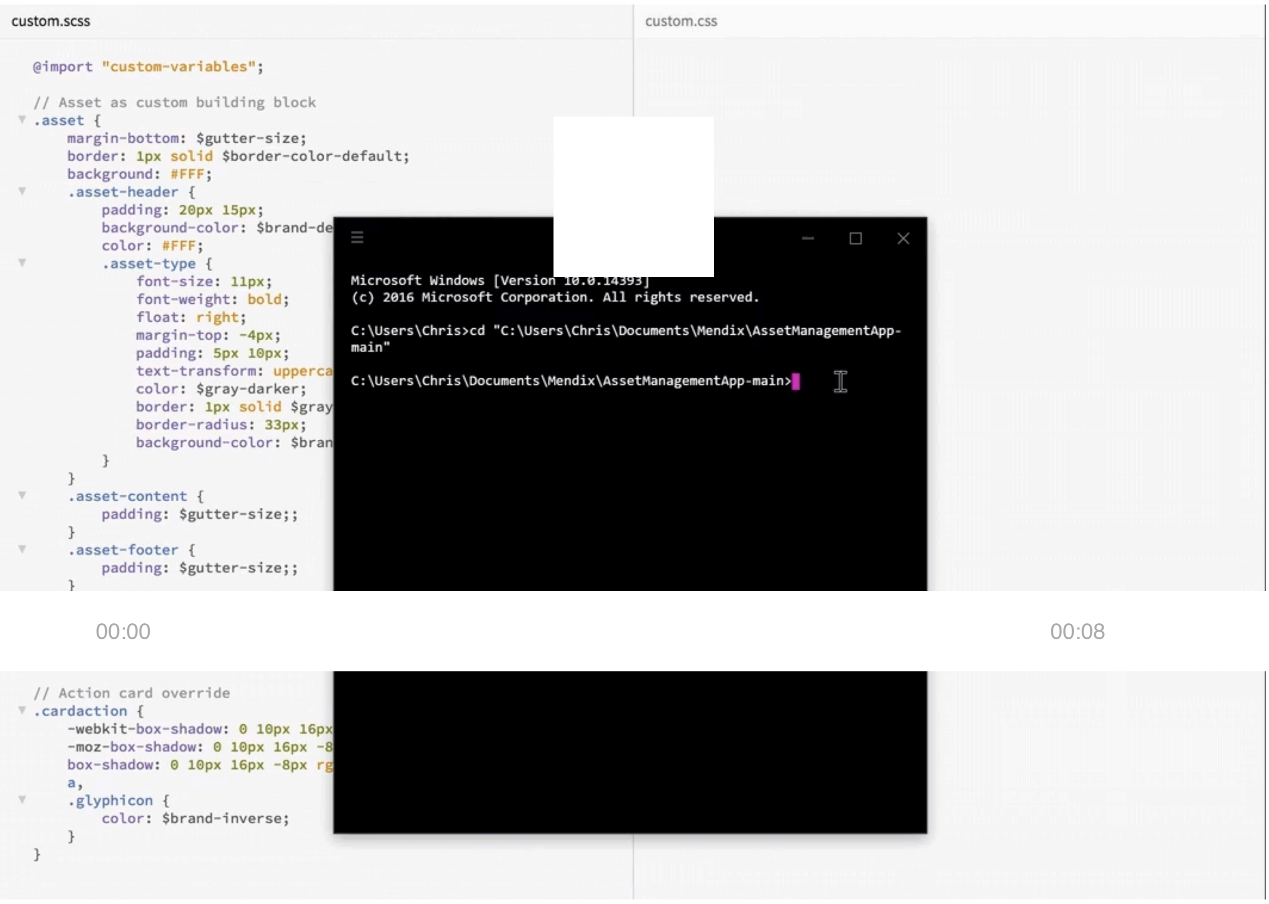
- **Menu widgets** – allows the end-user to navigate through the application (for example, menu bar, navigation tree)
- **Data widgets** – central to building forms in Mendix so that the end-user can view and edit data in the application (for example, data view, data grid, also common input widgets like text box and radio button)
- **Layout widgets** – form the backbone of the interface and are typically used on layouts (for example, layout container, title)
- **Common widgets** – commonly found on any page, layout, and snippet (for example, label, image)
- **Container widgets** – can contain other widgets (for example, table, group box)
- **Input widgets** – make it possible to show and edit the values of attributes and associations (for example, text box, date picker)
- **File widgets** – allow you to work with files, including images stored in files
- **Button widgets** – these buttons trigger actions (for example, save button, close page button, microflow button)
- **Visualization widgets** – more graphical widgets (for example, chart, map)

New widgets are built all the time by both the core Mendix development team and our community of developers using the standard available tooling. You can find available widgets in the Mendix Marketplace and download them to use in your application.

How Can I Customize the Look & Feel of My Apps?

The Mendix front-end uses the Atlas UI framework to render responsive pages or native mobile pages. The framework is based on a default theme that be fully customized. With the Theme Customizer tool, you can easily upload a logo that is used to determine your brand’s colors. The various colors, fonts, and size settings can be quickly configured and previewed.

Next to this, Mendix uses Sass to define the styling. The platform contains a configuration file with many variables that can be used to configure and tweak the theme. The configuration is set up so that it can be easily changed and extended by providing custom Sass files. It is also possible to directly extend the styling with custom CSS. The styling can also be shared, as discussed in the next section about creating consistent user experiences.



```
@import "custom-variables";

// Asset as custom building block
.asset {
    margin-bottom: $gutter-size;
    border: 1px solid $border-color-default;
    background: #FFF;
    .asset-header {
        padding: 20px 15px;
        background-color: $brand-de
        color: #FFF;
        .asset-type {
            font-size: 11px;
            font-weight: bold;
            float: right;
            margin-top: -4px;
            padding: 5px 10px;
            text-transform: upperca
            color: $gray-darker;
            border: 1px solid $gray
            border-radius: 33px;
            background-color: $bran
        }
    }
    .asset-content {
        padding: $gutter-size;;
    }
    .asset-footer {
        padding: $gutter-size;;
    }
```

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Chris>cd "C:\Users\Chris\Documents\Mendix\AssetManagementApp-
main"

C:\Users\Chris\Documents\Mendix\AssetManagementApp-main>
```

```
// Action card override
.cardaction {
    -webkit-box-shadow: 0 10px 16px
    -moz-box-shadow: 0 10px 16px -8
    box-shadow: 0 10px 16px -8px rg
    a,
    .glyphicon {
        color: $brand-inverse;
    }
}
```

How Does Mendix Support Reuse of Design Patterns & Elements?

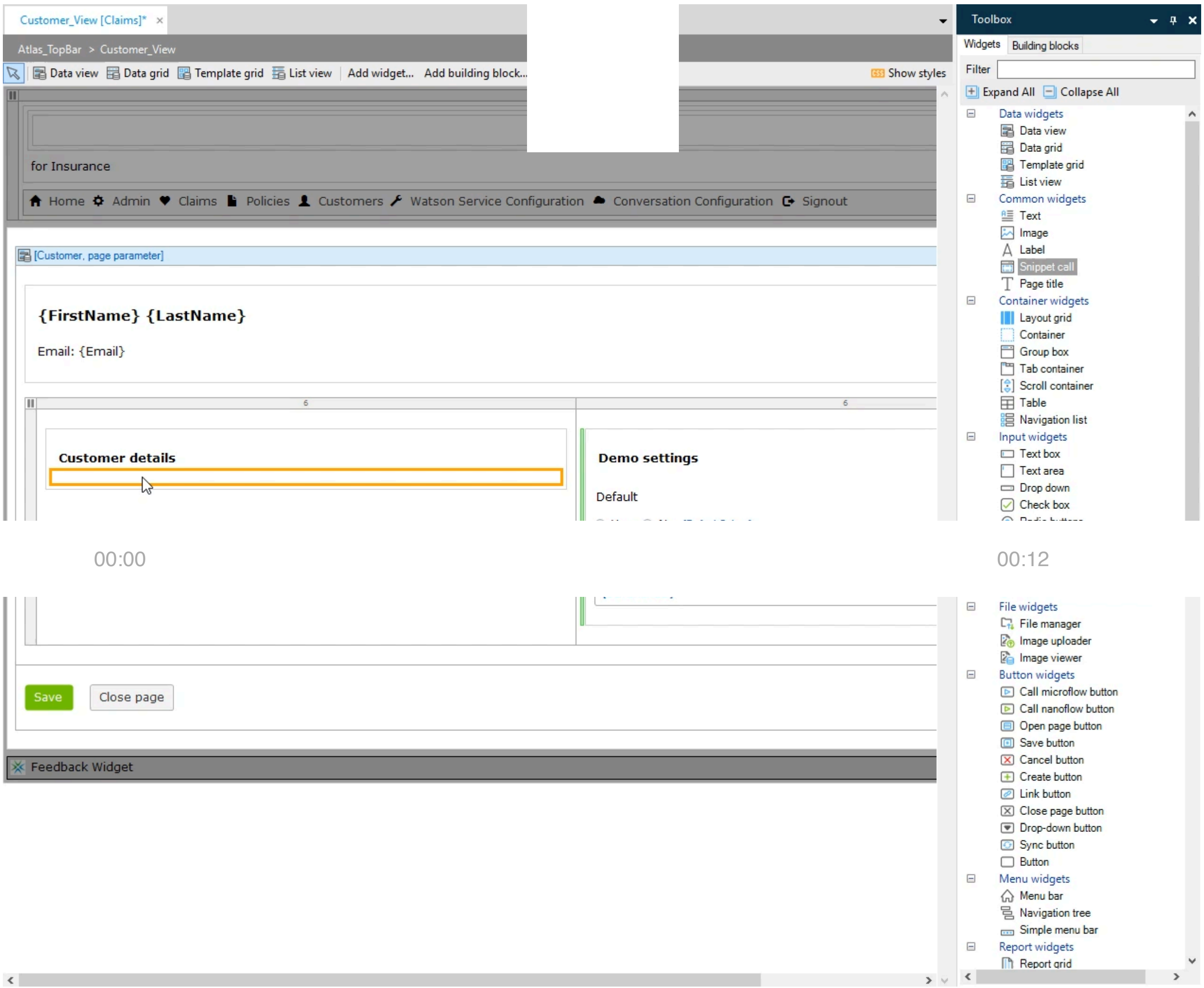
Atlas UI promotes the reuse of elements and layouts by facilitating a layered and componentized approach to application design. What is more, Atlas UI enables your UX designers to extend out-of-the-box components, assemble their own building blocks, create custom widgets, and override default design properties.

New and enhanced design elements are automatically propagated throughout all the applications in which each element is used.

The main types of reuse are described below.

Snippets

Snippets are fragments of page design that contain design elements, widgets, and data. Snippets are built and managed centrally at the model level, and they can be reused across multiple pages within an application.


Customer_View [Claims]*
Atlas_TopBar > Customer_View
Data view
Data grid
Template grid
List view
Add widget...
Add building block...
Show styles
for Insurance
Home
Admin
Claims
Policies
Customers
Watson Service Configuration
Conversation Configuration
Signout
[Customer, page parameter]
{FirstName} {LastName}
Email: {Email}
Customer details
Demo settings
Default
Save
Close page
Feedback Widget
00:00
00:12
Toolbox
Widgets
Building blocks
Filter
Expand All
Collapse All
Data widgets
Data view
Data grid
Template grid
List view
Common widgets
Text
Image
Label
Snippet call
Page title
Container widgets
Layout grid
Container
Group box
Tab container
Scroll container
Table
Navigation list
Input widgets
Text box
Text area
Drop down
Check box
File widgets
File manager
Image uploader
Image viewer
Button widgets
Call microflow button
Call nanoflow button
Open page button
Save button
Cancel button
Create button
Link button
Close page button
Drop-down button
Sync button
Button
Menu widgets
Menu bar
Navigation tree
Simple menu bar
Report widgets
Report grid