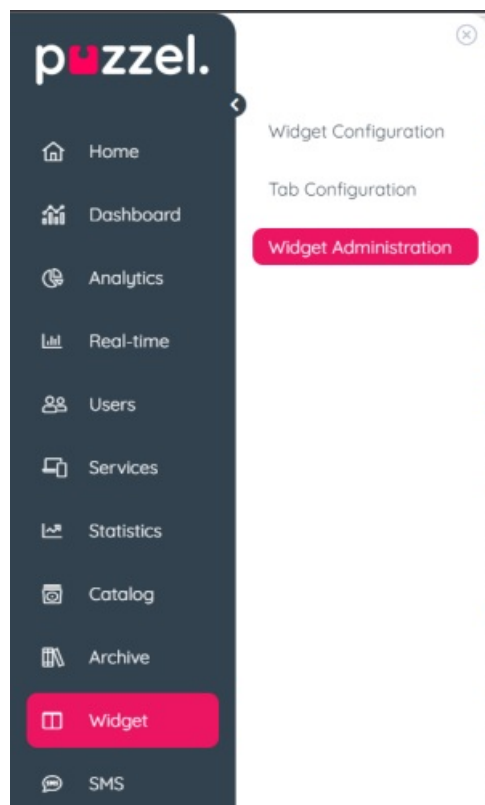


## Widget Administration

This section of the Administration Portal, allows for two important functions:

1. Enabling or disabling visibility for a specific widget for the current customer
2. Adding or editing widgets and its properties

The tab menu looks like this:



### Adding a new Widget

To add a new Widget in the list, simply press the Add Widget.

This will prompt you to enter the following fields:

1. **Name** this is the name of the widget
2. **Title** this allows you to enter a title, associated with the widget, which can be a little bit more descriptive
3. **URL** this is the URL, to which the iframe will point
4. **Handshake** - Sets default value of handshake to enabled. See handshake section below for more information
5. **Allow to get Puzzel access token** - Sets default value of canGetAccessToken to enabled. See canGetAccessToken section below for more information
6. **Type** - Either Widget or Search. Search is used for adding a [custom search source](#)

Once these values are entered, you can press the Add Widget button within the popup and finalise the settings.

## Adding properties to a Widget

Once a Widget has been created and added to the list, properties can be added to it. These properties will then be available for the widget to consume from the Agent Application and the option to be configurable from the [Widget Configuration](#) page. Here is an example of a widget with properties:

Key	Default Value	Configuration Options	Type
Widget Name	Title		Widget
90062 My widget			
Key	Default Value	<input type="checkbox"/> Set property in tab only <input type="checkbox"/> Possible to set option from widget <input type="checkbox"/> Set property in widget admin only	Type
url			String
Key	Default Value	<input type="checkbox"/> Set property in tab only <input type="checkbox"/> Possible to set option from widget <input checked="" type="checkbox"/> Set property in widget admin only	Type
handshake	<input checked="" type="checkbox"/>		Boolean
Key	Default Value	<input type="checkbox"/> Set property in tab only <input type="checkbox"/> Possible to set option from widget <input checked="" type="checkbox"/> Set property in widget admin only	Type
canGetAccessToken	<input type="checkbox"/>		Boolean
Key	Default Value	<input type="checkbox"/> Set property in tab only <input type="checkbox"/> Possible to set option from widget <input checked="" type="checkbox"/> Set property in widget admin only	Type
Custom Widget Property	Global Widget Property		String
Key	Default Value	<input type="checkbox"/> Set property in tab only <input type="checkbox"/> Possible to set option from widget <input type="checkbox"/> Set property in widget admin only	Type
Custom Tab Property	Tab Property - Can only be set in the tab configuration		String
Key	Default Value	<input type="checkbox"/> Set property in tab only <input type="checkbox"/> Possible to set option from widget <input type="checkbox"/> Set property in widget admin only	Type
featurePolicy	camera, microphone		String
Event	Key combination		Type
Custom Shortcut	alt+0		Shortcut

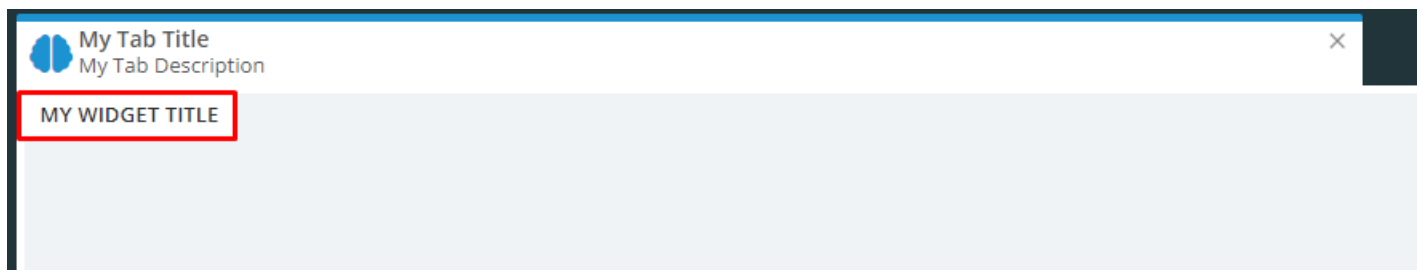
To add a new Widget property, simply press on the Add Widget property button. This will trigger a popup form, which contains the following values:

1. **Key** this is the string value, associated with a key for this widget. There are no restrictions on what one can enter, with the main goal being of making this a meaningful key
2. **Default Value** this is the default value for the key. If no other value is set at a later time, the key will have this value
3. **Type** this is the type of the property being added. The available values are:
  1. **String** this is a standard string, which can accept any string of numeric or text values
  2. **Drop down** this presents a drop down element with options to choose from. Useful for situations, where one needs to present a set of mutually exclusive options
  3. **Password** this is a value, which presents a password field. This is masked initially, however an administrator can set a different value from the Widget Configuration section later
  4. **Boolean** this creates a checkbox field, which similarly to a Boolean field can be checked or unchecked
  5. **Integer** this creates a numeric input field, which accepts numbers only
4. **Set property in tab only** means that the value for the key is set and read within a tab scope when configuring the tab under [Tab Configuration](#). Different tabs that uses the widget can have different values for the key whereas if unchecked the key is global and is set on a widget scope which is the default behaviour.
5. **Possible to set option from widget** allows the widget to update the value of the key. If disabled the widget can only read the value.
6. **Set property in widget admin only** disables the possibility to change the key value from [Widget Configuration](#). The key value can only be changed from Widget Administration.
7. **Event** - Uses the "Add widget shortcut" button. Triggers the entered Event when a key combination is pressed. Used for keyboard shortcuts. Requires that the widget uses the [Widget API Library](#).

## System properties and keys

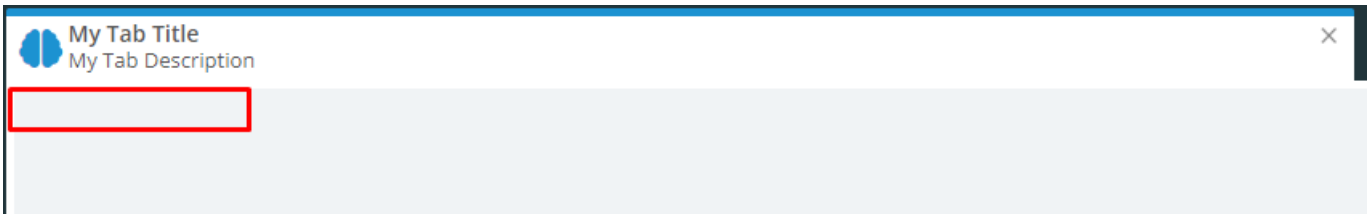
**Widget Name** - The system name of the widget with the syntax of `{customerKey} {Widget Name}`". This is used by back-end functions and can't be changed, however the title of the widget can be changed and will be shown in the Agent Application.

Widget Name	Title
<input type="text" value="90062 My widget"/>	<input type="text" value="My Widget Title"/>



It's also possible to enter a blank space as title to remove the title fully from the widget.

Widget Name	Title
<input type="text" value="90062 My widget"/>	<input type="text"/>



**Key - url** - The URL that the widget loads. Can be changed.

Key	Default Value
<input type="text" value="url"/>	<input type="text" value="https://my-widget-url.com"/>

**Key - handshake** - Used to signal to the Agent Application that the Widget is ready to receive events. Is handled by [Widget API Library](#) if the Widget uses the library. Good to use for single page applications built on for example Aurelia, Angular or React where the [Window load event](#) is not a reliable method to use to signal that the Widget is fully loaded.

Key	Default Value
<input type="text" value="handshake"/>	<input checked="" type="checkbox"/>

**Key - canGetAccessToken** - Needs to be enabled to be able to use the function [auth.getAccessToken](#) to retrieve an active authorization token that can be used to make API-calls to the [Puzzel Contact Centre REST API](#).

Key	Default Value
<input type="text" value="canGetAccessToken"/>	<input type="checkbox"/>

**Key - featurePolicy** - Feature Policy is used to add permissions for the widget to allow it to use features that would otherwise not be permitted for an iframe to use. For example there may be a need for the widget to allow copy to clipboard or use the microphone. You can read more about [feature policies here](#).

Key	Default Value
<input type="text" value="featurePolicy"/>	<input type="text" value="camera; microphone"/>