

Services Recipe

This recipe of custom code has been created by Pendo Professional Services, intended to extend the capabilities of Pendo products. All recipes are free to use, and as such, there is NO WARRANTY, SLA or SUPPORT for these recipes. Please do not reach out to Pendo Support for help with these recipes, as custom code is outside of the remit of their team and responsibilities.



Objective

To create a custom notification badge with an API interface to set alerts programmatically.

Background

Pendo Professional Services has developed a custom modification to the Pendo snippet that creates and modifies a new notification badge for the Pendo Resource Center. This package delivered as-is builds and exposes a javascript API to update and maintain the badge and the notification count. It is on the customer to write code to update the information displayed in the badge.

Requirements

- HTML
- CSS
- Javascript
- Resource Center Module id
- [Custom code for the Pendo snippet](#)

Implementation

This code is packaged as a modification to the snippet initialization and therefore is a change to your [Pendo Install](#). The code packaged will not include your other snippet customizations if you have set any up. This includes things like the IDs and Metadata that you're sending or any other agent customizations like annotations. **Be sure to merge your current snippet with the snippet provided to prevent any changes in current behavior on your app.**

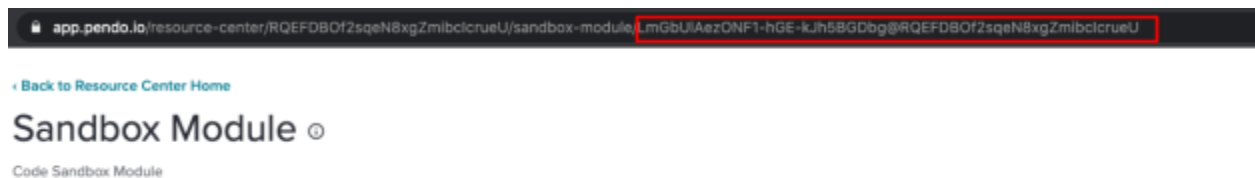
The code can be found under the Requirements section. While we cannot guarantee that this link will work forever if you reach out to services@pendo.io we should be able to provide a copy from our private github repo.

Below are the details of the code changes required. Pay attention to the following (subject to change based on implementation).

Adjust Resource Center module id:

The following line will need an update to change the custom html attribute selector `[data-pendo-module-guide-id="PLACE-RC-MODULE-ID-HERE"]`. The unique identifier will need to be replaced with the id of the resource center module you want the homeview badge to target. You can find this identifier by simply going to the module in Pendo. Additionally, the screenshot below shows where to find the id of the Resource Center module

```
pendo.customModuleSelector =  
'[data-pendo-module-guide-id="ZtnRUQ36CyBVVxrbiFBN-gMnLWI@pQqHDQbxwFzIXVZrd4J5sQrpfn0"  
> ._pendo-text-link > ._pendo-row > .pendo-mock-flexbox-row >  
.pendo-mock-flexbox-element']
```



Set alerts:

In order to take advantage of the new alerts badge you only need to use a single function - `pendo.setAlerts(alertNum)`. This function takes the integer you'd like the badge to have. On initialization this is set to 0, however the value on initialization can be modified. Additionally, the badge pulls the number of new Pendo Announcement Guides automatically and adds them to the integer provided when executing the function.

The badge will not draw when alerts are set to 0 or less. If the alerts are 0 or less and the badge is currently visible it will remove itself from the page. Alerts must be set on every new Pendo initialization and updated as needed when notifications have been seen or cleared.

The code modifying the snippet has no new packages installed or required. However, the code uses [Mutation Observer](#) which is not available on some legacy browsers - Please refer to the [browser support tables](#) to determine if the implementation will work for your supported browsers.

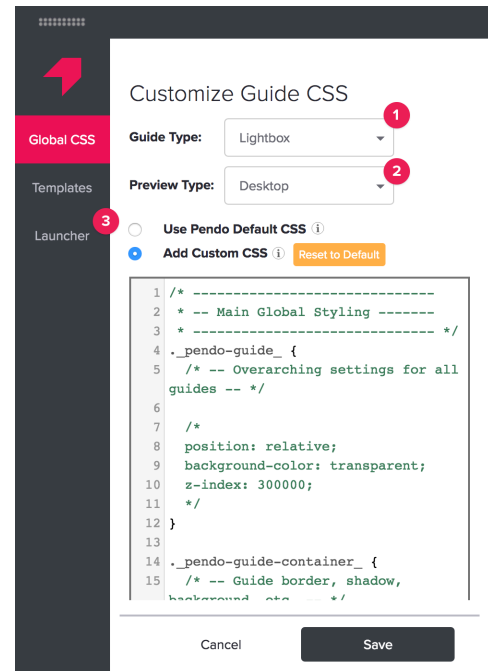
Customization

The secondary badge was built to be customized from within Pendo via [global css](#). This allows for modification of the badge styling without additional snippet changes after implementation.

Note:

This implementation assumes the snippet will be placed on the main application you have Pendo installed. If this is an additional application other than the main, the CSS will need to be moved to your own application stylesheet. To find if your application is the main, navigate to the Subscription settings page and select the app where the installation of the modified snippet will occur. The URL of the page should contain an integer after /app/. If the integer is -323232 you are on the main app of the subscription and can use the global CSS method described below.

To Manage:



1. To customize the colors location of the RC secondary badge you will need to navigate to the Global CSS window. Instructions follow - Going to the [in-app guide designer](#).
2. Select the "Global Settings" button in the top right corner of the window
3. Select the Global CSS tab
4. From here you can add any CSS that you wish to be inherited by all guides. You will see stubs for all of the CSS classes that Pendo uses. You can add styles to those classes, and/or you can add your own classes and IDs.

From this view you are able to add new css that will change how the secondary badge is displayed.

Below is CSS you can append to the top of this text. See comments in-line to understand how the settings affect the badge.

Note:

Some of this CSS is required to have the badge display properly!

```
._pendo-secondary-home-view-bubble{  
  background-color: #359846 !important; /* sets color of homeview badge */  
}
```

```
.pendo-resource-center-badge-secondary-notification-bubble{  
  background-color: #359846; /* sets color of RC badge */  
  top: -2px !important; /* sets top offset of RC Badge */
```

```
    right: -9px !important; /* sets right offset of RC Badge */
    left: auto !important; /* sets left offset of RC Badge */
}

/* */
.pendo-resource-center-badge-secondary-notification-bubble::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-color: #359846;
    border-radius: 100%;
    z-index: -1;
    animation: pulse 2s infinite;
    will-change: transform;
}
```