# Smooth Loader for sits\_collapsible\_panel

## Purpose

Calling certain Tribal JavaScript functions such as sits\_collapsible\_panel or tablesaw can create a clunky user interface as the dom is re-written after it has been painted to the screen.

This solution aims to hide the parts of the dom that we know will be manipulated and gracefully resize placeholders before displaying the content.

Inspiration for styling has been taken from Facebook's newsfeed content placeholders.

## Usage

### As a call-back

Use as a callback for the Tribal provided function sits\_collapsible\_panel:

sits\_collapsible\_panel('#unsubmitted, #pending',true, contentPlaceholder);

### HTML annotation

Content placeholders will need to be created which emulate the "skeleton" html of the content that is being loaded. For instance, if I am loading in an sv-panel with a title and content in the body, my placeholder would be an empty sv-panel.

These placeholders will need marking up with the data attribute " data-content-placeholder" and this will need to be provided with the id of the corresponding panel that is being loaded.

All placeholders will need to be enclosed in an element as this will preserve their positioning in the dom when they are given absolute positioning (to allow them to fade out without seeing a huge lump of the DOM disappear).

The header will need to be given the class "animated-background" to create the Facebook style 'flash' :

<div <!-- the parent element -->><div class="sv-panel sv-panel-default" data-content-placeholder="targetID">

<div class="sv-panel-heading animated-background">Loading Content</div>

…

The panel that is being loaded will need to be given an ID (as used above) and the data attribute: data-loaded-content

<div id=" targetID" class="sv-panel sv-panel-primary" data-loaded-content>

…

Finally the js/css for this solution need to be included <<@RESD-CONTENT-PLACEHOLDERS>>

## Notes

Include the css and script before the content in the DOM so that it is loaded with the correct styles.

## Full Example

<script>

sits\_attach\_event("window","load",function() {

sits\_collapsible\_panel('#complete',false);

sits\_collapsible\_panel('#unsubmitted, #pending',true, contentPlaceholder);

});

</script>

<<@RESD-CONTENT-PLACEHOLDERS>>

<div id="loader">

<div **class**="sv-panel sv-panel-default" data-content-placeholder="unsubmitted">

<div **class**="sv-panel-heading animated-background"> Requests Awaiting Submission</div>

<div **class**="sv-panel-body">

&nbsp;

</div>

</div>

<div **class**="sv-panel sv-panel-default" data-content-placeholder="pending">

<div **class**="sv-panel-heading animated-background">Requests Submitted</div>

<div **class**="sv-panel-body">

&nbsp;

</div>

</div>

<div **class**="sv-panel sv-panel-default" data-content-placeholder="complete">

<div **class**="sv-panel-heading animated-background">Requests Completed</div>

</div>

</div>

<div id="page">

<div id="unsubmitted" **class**="sv-panel sv-panel-danger requestPanel" data-loaded-content>

<div **class**="sv-panel-heading"> Requests Awaiting Submission</div>

<div **class**="sv-panel-body">

I am the first loaded content

</div>

</div>

<div id="pending" **class**="sv-panel sv-panel-success" data-loaded-content>

<div **class**="sv-panel-heading"> Requests Submitted</div>

<div **class**="sv-panel-body">

I am the loaded content!

</div>

</div>

<div id="complete" **class**="sv-panel sv-panel-info" data-loaded-content>

<div **class**="sv-panel-heading"> Requests Completed</div>

<div **class**="sv-panel-body">

I am also loaded content!

</div>

</div>

</div>