

DPS (2015) HTML Gesture Example

This HTML article example utilizes the Adobe DPS (2015) HTML Gesture API.

API

The **DPS HTML Gesture API** will provide content authors that work with HTML articles (and InDesign web overlays) with the ability to override the default behaviors for the tap and swipe gestures. Now for the first time, developers will gain complete control of when to display the navigation UI and where the swipe gestures will be registered to (HTML article or the Viewer).

There are four main functions that will be accessible:

- **disableNavigation** - Disables the viewer navigation paradigm for this page context, meaning navigation on this page content will NOT be handled by the viewer. Instead the content will need to handle navigation, using the other methods within this API. By default, viewer navigation is enabled. This call will have no effect when viewer navigation is already disabled.
- **enableNavigation** - This function enables the viewer navigation paradigm for the parameterized element, meaning navigation on the element will be handled by the viewer. By default, viewer navigation is enabled. This call will have no effect when viewer navigation is already enabled.
- **relinquishCurrentGesture** - Relinquish control of the current gesture to the viewer. The viewer should then respond to the current gesture. This function will have no effect unless viewer navigation has been disabled.
- **toggleNavigationUI** - Instruct the viewer to toggle the navigation UI. Navigation UI typically includes a HUD that will be automatically re-hidden after a certain amount of time. This function will have no effect unless viewer navigation has been disabled.

The DPS HTML Gesture API is created to ensure a good user experience and prevent a user from getting stuck or blocked inside of the HTML content.

Namespace **adobeDPS.Gesture**

The API is located within the following namespace: `adobeDPS.Gesture`. There are four functions that are provided in this API:

- [enableNavigation](#)
- [disableNavigation](#)
- [relinquishCurrentGesture](#)
- [toggleNavigationUI](#)

function enableNavigation(elements)

This function enables the viewer navigation paradigm for the parameterized element, meaning navigation on the element will be handled by the viewer.

By default, viewer navigation is enabled. This call will have no effect when viewer navigation is already enabled.

```
$(document).ready(function() {  
    // get the reference to the slider(s)  
    var $sliderOne = $('#slideshow-container-one'),  
        $sliderTwo = $('#slideshow-container-two');  
    // pass in the slider object(s) by doing slider[0]  
    adobeDPS.Gesture.enableNavigation([$sliderOne[0], sliderTwo[0]]);  
});
```

Note: for enableNavigation (and disableNavigation), you can either pass in an array of elements or just a single element:

```
adobeDPS.Gesture.enableNavigation($sliderOne[0]);
```

function disableNavigation(elements)

Disables the viewer navigation paradigm for this page context, meaning navigation on this page content will NOT be handled by the viewer. Instead the content will need to handle navigation, using the other methods within this API.

By default, viewer navigation is enabled. This call will have no effect when viewer navigation is already disabled.

```
$(document).ready(function() {  
    // get the reference to the slider(s)  
    var $sliderOne = $('#slideshow-container-one'),  
        $sliderTwo = $('#slideshow-container-two');  
    // pass in the slider object(s) by doing slider[0]  
    adobeDPS.Gesture.disableNavigation([$sliderOne[0], sliderTwo[0]]);  
});
```

function relinquishCurrentGesture()

Relinquish control of the current gesture to the viewer. The viewer should then respond to the current gesture.

```
/*
 * Add a listener to an on swipe event,
 * most slideshow frameworks will have their own custom swipe events.
 *
 * NOTE: do not attach the listener to the END of a swipe,
 * you must active the relinquishCurrentGesture() function during a swipe,
 * this will let the viewer to determine the gesture,
 * and allow the swipe to previous or next article.
 */
$('#slideshow-container-one').on('on-swipe-event', function() {
    // TODO: determines the swipe event, horizontal vs vertical
    if (isSwipeOnFirstOrLast) {
        //
        adobeDPS.Gesture.relinquishCurrentGesture();
    }
});
```

function toggleNavigationUI()

Instruct the viewer to toggle the navigation UI. Navigation UI typically includes a HUD that will be automatically re-hidden after a certain amount of time.

This function will have no effect unless viewer navigation has been disabled.

```
<html>
<head></head>
<body>
    <button onclick="toggleNavUi()">Click me to display the Navigation UI</button>
    <script>
        function toggleNavUi() {
            adobeDPS.Gesture.toggleNavigationUI();
        }
    </script>
</body>
</html>
```