alert('Pjax is completed');

});

'); ?>

Or, you can reload the container manually by using its selector:

<?php $this->registerJs('

$("#my-button").on("click", function() {

$.pjax.reload({container:"#my-block"});

});

'); ?>

How it works...

Pjax is a simple wrapper for any code fragment. It subscribes to click events of all links in the fragment and replaces the whole page, reloading it into Ajax calls. We can use the data-pjax attribute for wrapped forms, and any form submissions will trigger an Ajax request.

The widget will load and update on-the-fly widget body content without, loading the layout resources (JS, CSS).

You may configure the $linkSelector of the widget to specify which links should trigger Pjax, and configure $formSelector to specify which form submission may trigger Pjax.

You may disable Pjax for a specific link inside the container by adding the data-pjax="0" attribute to this link.

See also

• For more information about the extension’s usage, see:

o [http://www.viiframework.com/doc-2.0/vii-widgets-piax.html](http://www.yiiframework.com/doc-2.0/yii-widgets-pjax.html) o [https ://github.com/yiisoft/jquery-pjax](https://github.com/yiisoft/jquery-pjax)

• For more information about client-side options and methods, refer to [https://github.com/yiisoft/jquery-pjax#usage](https://github.com/yiisoft/jquery-pjax%23usage)