Pjax jQuery plugin

Pjax is a widget that integrates the pjax jQuery plugin. All content that is wrapped by this widget will be  
reloaded by AJAX without refreshing the current page. The widget also uses the HTML5 History API to  
change the current URL in your browser’s address line.

Подготовка

Создайте новое приложение с помощью диспетчера пакетов Composer, как описано в официальном руководстве по адресу  
<http://www.yiiframework.com/doc-2.0/guide-start-installation.html>.   
По русски <http://yiiframework.domain-na.me/doc/guide/2.0/ru/start-installation>.

How to do it...

In the following example, you can see how you use Pjax with the yii\grid\Gridview widget:

<?php

use yii\widgets\Pjax;

?>

<?php Pjax::begin(); ?>

<?= GridView::widget([...]); ?>

<?php Pjax::end(); ?>

Just wrap any code fragment in the Pjax:: begin () and Pjax: :end () calls.

This will render the following HTML code:

<div id="w1">

<div id="w2" class="grid-view">...</div>

</div>

<script type="text/javascript">jQuery(document).ready(function () {  
jQuery(document).pjax("#w1 a", "#w1", {...});

});</script>

All the wrapped content with pagination and sorting links will be reloaded by AJAX.

Specifying a custom ID

Pjax gets page content from AJAX requests and then extracts its own DOM element with the same ID.  
You can optimize page rendering performance by rendering content without layout, especially for Pjax  
requests:

public function actionIndex()

{

$dataProvider = ...;

if (Yii::$app->request->isPjax) {

return $this->renderPartial('\_items', [

'dataProvider' => $dataProvider,

]);

} else {

return $this->render('index', [

'dataProvider' => $dataProvider,

]);

}

}

By default, the yii\base\widget: : getid method increments identifiers, and therefore widgets, on any  
page that has incremented attributes:

<nav id="w0">...</nav> // Main navigation  
<ul id="w1">...</ul> // Breadcrumbs widget  
<div id="w2">...</div> // Pjax widget

To render using the renderPartial() or renderAjax() methods, without rendering the layout, your own  
page will have only one widget with the number 0:

<div id="w0">...</div> // Pjax widget

In the result, your own widget will not have found its own block with the w2 selector at the next request.

However, Pjax will find the same block with the w2 selector in the Ajax response. In the result, your own  
widget will not have found the block with the w2 selector at the next request.

Therefore, you must manually specify a unique identifier for all your Pjax widgets to avoid different  
conflicts:

<?php Pjax::begin(['id' => 'countries']) ?>

<?= GridView::widget([...]); ?>

<?php Pjax::end() ?>

Using ActiveForm

By default, Pjax works only with links in the wrapped block. If you want to use it with the ActiveForm  
widget, you must use the data-pj ax option of the form:

<?php

use \yii\widgets\Pjax

use \yii\widgets\ActiveForm;

<?php yii\widgets\Pjax::begin(['id' => 'my-block']) ?>

<?php $form = ActiveForm::begin(['options' => [

'data-pjax' => true,

]]); ?>

<?= $form->field($model, 'name') ?>

<?php ActiveForm::end(); ?>

<?php Pjax::end(); ?>

It adds corresponding listeners on the form submitting event.

You can also use the $formSelector option of the Pjax widget to specify which form submission may  
trigger pjax.

Working with the client-side script

You can subscribe to container events:

<?php $this->registerJs('

$("#my-block").on("pjax:complete", function() {

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)