

关于不是使用 GridView 使用的是ListView，我又想做功能丰富的分页效果，如何做呢？

我现在的分页是基于 yii2-widget-linkpager 这个来做的

<https://www.yiiframework.com/extension/yii2-widget-linkpager>

18	15861156656	User	2015-04-02 14:07	(not set)	Normal
19	athanasyu@qq.com	User	2015-03-26 17:28	(not set)	Normal
20	111@123.com	User	2015-02-05 14:53	(not set)	Normal

Home Previous 1 2 3 4 Next Last 20

om	User	2015-03-26 17:28	(not set)	Normal
	User	2015-02-05 14:53	(not set)	Normal

Home Page Previous Page 1 2 3 4 Next Page Last Page Jump 1 Page 20

Usage

Once the extension is installed, simply use it in your code by :

GridView options

```
'filterSelector' => "select[name='". $dataProvider->getPagination()->pageSizeParam. "'],input[name='". $dataProvider->getPagination()->pageParam. "']", //这个很重要，是精华部分，这个最终体现在js中，这样的话，我们Jump或者选择一页展示多少条我们change，我们选择的值发生了变化，就会触发jquery的onchange事件，就会自动请求最新的url，相当于提交了form一样，不然这个Jump或者选择一页展示多少条之后没有反应，那就没有意义了
```

```
'pager' => [  
    'class' => \liyunfang\pager\LinkPager::className(),  
    //'template' => '{pageButtons} {customPage} {pageSize}',  
    //'pageSizeList' => [10, 20, 30, 50],  
    //'customPageWidth' => 50,  
    //'customPageBefore' => ' Jump to ',  
    //'customPageAfter' => ' Page ',  
],
```

ModelSearch

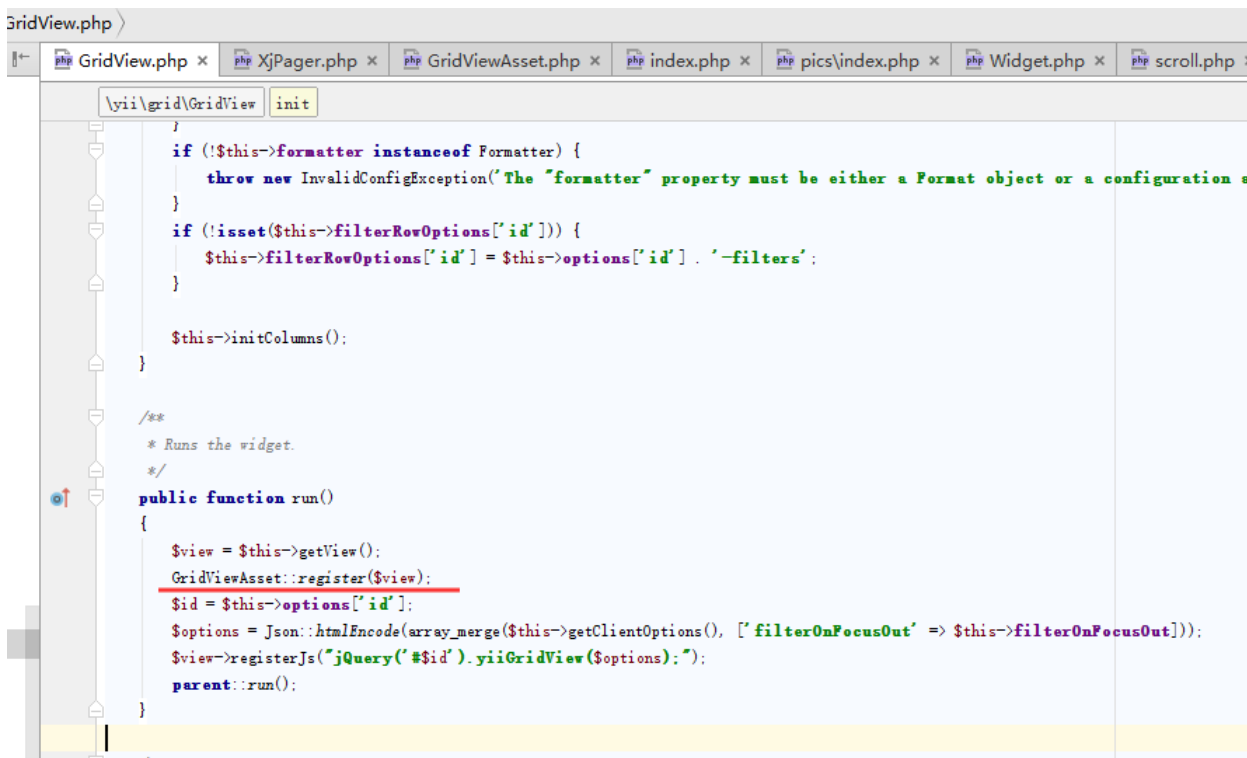
```
public function search($params)  
{  
    ...  
    $pageSize = isset($params['per-page']) ? intval($params['per-page']) : 10; //这个可以不要
```

```

        $dataProvider = new ActiveDataProvider([
            'query' => $query,
            'pagination' => ['pageSize' => $pageSize,], //上面的$pageSize
不要了, 自然这里也就不需要了
        ]);

```

在GridView中使用它, 直接go到第几页, 一页展示多少条都是ok的, 都会有值的变化
onchange之后, 里面请求最新的url数据了, 这些js程序都是在GridView里面使用了
yii.gridView.js文件



所以我们假如基于 ListView展示的数据的话, 分页也采用上面的"liyunfang/yii2-
widget-linkpager": "*", 出现的 go与一页展示多少条数据 就会出现我们选择了没有
反应的现象

因为ListView没有加载yii.gridView.js文件, 自然没有对应的js程序, 所以想要让go与一页展
示多少条数据要么自己 现在单独再写 与yii.gridView.js 一样结果的js程序, 要么在ListView
中也引入使用yii.gridView.js即可

我之前没有直接使用现成的yii.gridView.js, 而是自己写的js程序, 只是写了一页展示多少条
数据, 没有写go的那个, 我想原理也是一样的吧, 都是基于change事件即可
下面是我自定义写的 select那个js程序

我首先自己定义了一个 XjPager类, 继承自LinkPager。

在ListView里面引入使用XjPager类即可

```
<ul class="box topic-list p-0 clearfix">
<?=\yii\widgets\ListView::widget([
    'dataProvider' => $dataProvider,
    'options' => ['tag' => false],
    'itemView' => '_item', //子视图
    'layout' => "{items}\n </ul><div class=\"box-page clearfix\" id=\"long-page\">{pager}</div>",
    'itemOptions' => ['tag' => false],
    'pager' => [
        'class' => \common\widgets\XjPager::class,
        // 'template' => "{pageButtons} <div class=\"custompage\"> {customPage} {pageSize2} </div>\n",
        'template' => "{pageButtons} <div class=\"custompage\"> {customPage} {pageSize} </div>\n",
        'options' => ['id' => 'page6666'],
        'filterRowOptions' => ['id' => 'page6666'],
        'filterSelector' => 'input[name=page], select[name=per-page]',
        'pageSizeOptions' => [
            'onChange' => new \yii\web\JsExpression('
```

精华代码

yii2中使用select 又想要 点亮哪个option, 就需要select name的值对应上的option的value值, 那么这个option就是selected选中的

这里我是使用的js代码, 不是jQuery

```
'onChange' => new \yii\web\JsExpression('
    //获取下拉框选中的索引
    var index=this.selectedIndex;
    //console.log(index);
    //拿到选中项的属性值
    var value=this[index].getAttribute("data-v");
    //console.log(value);
    if(value) window.location = value;
    return true;
'),
```

onChange事件, 事件处理内容是 JsExpression中的内容

this指代的就是select元素, 我使用var index=this.selectedIndex; 就是获取select下的选中option的下标是多少, 赋值给index, 然后使用 this[index]即可得到这个选中的option

我之前是 直接 onChange = 'window.location = this.value'

意思是, select的值即是当时选中你的option的value, select的值发生变化的话, 那么链接跳转至select选中的值, 这就表示我们需要将option的value里面的内容写成链接的样式, 链接里面是包括per-page参数的, 也就是一页展示几条数据的参数

但是结果差强人意, 那么就是可以跳转了, 但是select的展示显示当前是哪个option选中的效果就是有问题, 始终是展示的是第一个option的信息

后来看了yii2的select的点亮selected的写法, 知道了

```

protected function renderPageSize()
{
    $pageSizeList = [];
    foreach ($this->pageSizeList as $value) {
public static function BaseHtml::dropDownList($name, $selection = null, $items = [], $options = []): string
    {
        return Html::dropDownList($this->pagination->pageSizeParam, $this->pagination->getPageSize(), $pageSizeList, $this->pageSizeOptions);
    }
}

```

```

*/
public static function dropDownList($name, $selection = null, $items = [], $options = [])
{
    if (!empty($options['multiple'])) {
        return static::listBox($name, $selection, $items, $options);
    }
    $options['name'] = $name;
    unset($options['unselect']);
    $selectOptions = static::renderSelectOptions($selection, $items, $options);
    return static::tag('select', "\n" . $selectOptions . "\n", $options);
}

```

```

foreach ($items as $key => $value) {
    if (is_array($value)) {
        $groupAttrs = isset($groups[$key]) ? $groups[$key] : [];
        if (!isset($groupAttrs['label'])) {
            $groupAttrs['label'] = $key;
        }
        $attrs = ['options' => $options, 'groups' => $groups, 'encodeSpaces' => $encodeSpaces, 'encode' => $encode];
        $content = static::renderSelectOptions($selection, $value, $attrs);
        $lines[] = static::tag('optgroup', "\n" . $content . "\n", $groupAttrs);
    } else {
        $attrs = isset($options[$key]) ? $options[$key] : [];
        $attrs['value'] = (string) $key;
        if (!array_key_exists('selected', $attrs)) {
            $attrs['selected'] = $selection !== null &&
                (!ArrayHelper::isTraversable($selection) && !strcmp($key, $selection)
                || ArrayHelper::isTraversable($selection) && ArrayHelper::isIn((string) $key, $selection));
        }
        $text = $encode ? static::encode($value) : $value;
        if ($encodeSpaces) {
            $text = str_replace(' ', '&nbsp;', $text);
        }
        $lines[] = static::tag('option', $text, $attrs);
    }
}

```

就是\$attrs很重要，他是组成option的很重要的提前数据

每个option的\$attrs数据就是从options中取对于key的内容

注意option中参数属性selected在yii2中写的是 \$attrs['selected']=true|false

\$attrs['selected'] = \$selection !== null &&

(!ArrayHelper::isTraversable(\$selection) && !strcmp(\$key, \$selection)

|| ArrayHelper::isTraversable(\$selection) && ArrayHelper::isIn((string) \$key, \$selection));

我们看到\$selection我们实际情况是有值的，不是null，值是\$this->pagination->getPageSize()

就是我们一页展示几条数据的结果，是数字，不是链接，

还有\$attrs['value'] = (string) \$key; option的value就是key, 而这个key并且上面还参与了\$attrs['selected']的运算

我们准备的select数据中用作items即options们的数据是这样的

```
[  
    "http://admin.localwuma.net/pics/index?page=3&per-page=10" => 10  
    "http://admin.localwuma.net/pics/index?page=3&per-page=20" => 20  
    "http://admin.localwuma.net/pics/index?page=3&per-page=50" => 50  
    .....  
]
```

key就是那样链接, 显然是与 selection的只有数字是不对应的, 所以, 我们只能将key还是只使用数字, 将链接内容写在别的属性中, 比如自定义属性data-v中

这样我们是不能将data-v写入items中, 像上面的数组item最后生成的是option的value与text内容, 其他option的属性也是可以写的, 但是不是写在数组items中, 是写入options中

```
*/  
public static function dropDownList($name, $selection = null, $items = [], $options = [])  
{  
    if (!empty($options['multiple'])) {
```

例如下就是如何定义属性data-v

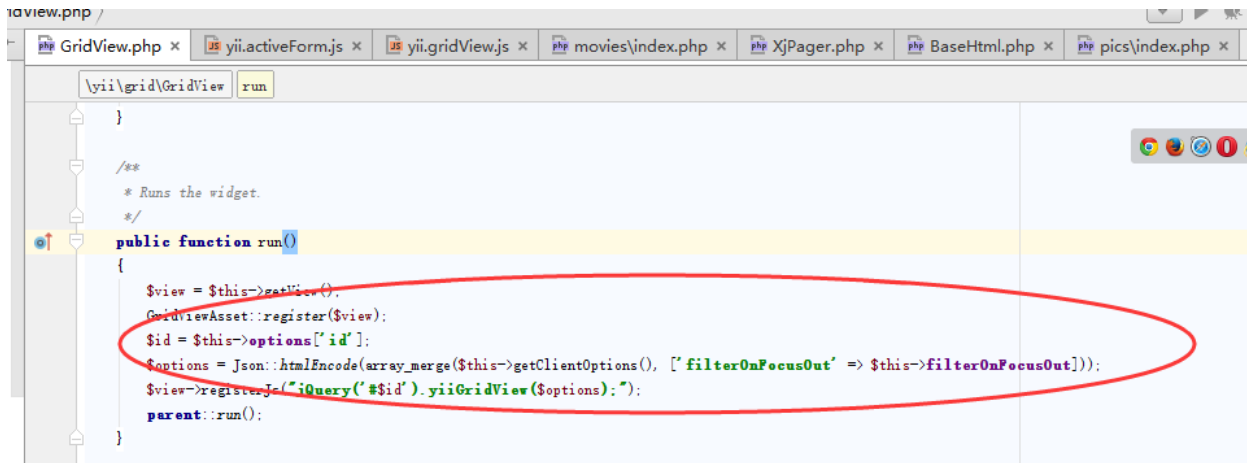
```
//自己改造一下  
protected function renderPageSize2()  
{  
    $pageSizeList = [];  
    $current_url = array_merge(Yii::$app->controller->getRoute(), Yii::$app->request->queryParams);  
    foreach ($this->pageSizeList as $value) {  
        $url = $current_url;  
        $url[$this->pagination->pageSizeParam] = $value;  
        $pageSizeList[$value] = $value;  
        $this->pageSizeOptions['options'][$value] = ["data-v" => Url::to($url)];  
    }  
    return Html::dropDownList($this->pagination->pageSizeParam, $this->pagination->getPageSize(), $pageSizeList, $this->pageSizeOptions);  
}
```

那么我们回到上面, 你就知道了我没有直接使用 onchange = 'window.location = this.value' 了吧, 还是使用了JsExpression中的内容

最后我们通过var value=this[index].getAttribute("data-v"); 获取链接最后跳转链接即可 this[index] 就是得到option, 在通过getAttribute获取option中的属性data-v

上面是说的自己写 select的js程序, 下面说说 Xjpager中如何直接借助yii.gridView.js文件

将GridView中我圈起来的内容复制进去 XjPager类中



```
GridView.php x yii.activeForm.js x yii.gridView.js x movies\index.php x XjPager.php x BaseHtml.php x pics\index.php x
[yii\grid\GridView] run
}
/**
 * Runs the widget.
 */
public function run()
{
    $view = $this->getView();
    GridViewAsset::register($view);
    $id = $this->options['id'];
    $options = Json::htmlEncode(array_merge($this->getClientOptions(), ['filterOnFocusOut' => $this->filterOnFocusOut]));
    $view->registerJs("jQuery('#$id').yiiGridView($options);");
    parent::run();
}
```



```
View.php >
GridView.php x yii.activeForm.js x yii.gridView.js x movies\index.php x XjPager.php x BaseHtml.php x pi
[yii\grid\GridView] getClientOptions
* Returns the options for the grid view JS widget.
* @return array the options
*/
protected function getClientOptions()
{
    $filterUrl = isset($this->filterUrl) ? $this->filterUrl : Yii::$app->request->url;
    $id = $this->filterRowOptions['id'];
    $filterSelector = "#$id input, #$id select";
    if (isset($this->filterSelector)) {
        $filterSelector .= ', ' . $this->filterSelector;
    }
    return [
        'filterUrl' => Url::to($filterUrl),
        'filterSelector' => $filterSelector,
    ];
}
```

对应的属性像


filterUrl

filterOnFocusOut等等也复制过来即可

最重要的就是 触发对应的 change事件的元素

是在\$filterSelector 与 \$this->filterSelector身上，即需要可以onchange的元素内容改变之后需要跳转的，就写进去就好了。这里自然我是一个 go input ，一个select

关于 yii.gridView.js中对于 filterUrl的处理，我们没有指定的话，就是当前的url，例如像下面的



```
<script src="/assets/45125b45/js/bootstrap.js"></script>
<script>
    jQuery(function ($) {
        jQuery('#page6666').yiiGridView({"filterUrl":"/pics/index?page=5\u0026per-page=20","filterSelector":"#page6666 input,#page6666 select, input[name=page], select[name=per-page]","filterOnFocusOut":true});
    });
}
Inherited from
body {
    font-family: yahei;
}
```

链接会使用php的urlencode处理一下

```

* return array the options
*/
protected function getClientOptions()
{
    $filterUrl = isset($this->filterUrl) ? $this->filterUrl : Yii::$app->request->url;
    $id = $this->filterRowOptions['id'];
    $filterSelector = "#$id input, #$id select";
    if (isset($this->filterSelector)) {
        $filterSelector .= ', ' . $this->filterSelector;
    }

    return [
        'filterUrl' => Url::to($filterUrl),
        'filterSelector' => $filterSelector,
    ];
}

```

```

.php >
dView.php x  php BaseUrl.php x  php UrlManager.php x  JS yii.activeForm.js x  JS yii.gridView.js x  JS
\yii\web\UrlManager  createUrl

    }

    $route = ltrim($route, '/');
    return "$baseUrl/{$route}{$anchor}";
}

$url = "$baseUrl?{$this->routeParams}=" . urlencode($route);
if (!empty($params) && ($query = http_build_query($params)) != '') {
    $url .= '&' . $query;
}

return $url . $anchor;
}

```

最后yii.gridView.js中造了一个form表单

```

//
var pos = settings.filterUrl.indexOf('?');
var url = pos < 0 ? settings.filterUrl : settings.filterUrl.substring(0, pos);
var hashPos = settings.filterUrl.indexOf('#');
if (pos >= 0 && hashPos >= 0) {
    url += settings.filterUrl.substring(hashPos);
}
console.log('url:' + url);
$grid.find('form.gridview-filter-form').remove();
var $form = $('<form/>', {
    action: url,
    method: 'get',
    'class': 'gridview-filter-form',
    style: 'display:none',
    'data-pjax': ''
}).appendTo($grid);
$.each(data, function (name, values) {
    $.each(values, function (index, value) {
        $form.append($(' <input/>').attr({type: 'hidden', name: name, value: value}));
    });
});

var event = $.Event(gridEvents.beforeFilter);
$grid.trigger(event);
if (event.result === false) {
    return;
}

$form.submit();

$grid.trigger(gridEvents.afterFilter);
},

```

所以得到一个结论

js中最开始的url是/pics/index内容

在就是 hidden我们 select中写入的元素的 比如input, select,

form提交的时候, 把他们的name值带着url上, 所以一般我们设置 filterUrl参数的时候只需要设置 path路径即? 前面的内容即可, 其实是不需要设置完整的带参数的filterUrl的

完整的demo例子, 请看当前目录下的文件夹 “ListView使用复杂分页demo”