

# yii2 DataProvider

Created by 杨超, last modified on 2017 Oct 27

- [DataProvider基础](#)
- [WebDataProvider介绍](#)
  - [应用场景](#)
  - [实现思路](#)
  - [实例](#)
  - [具体源码](#)

## DataProvider基础

dataproducer -

常用

`ActiveDataProvider`

`ArrayDataProvider`

所有的DataProvider支持DataProviderInterface

提供获取 `getTotalCount/getCount/getModels/getKeys/getSort/getPagination`

[Collapse source](#)

```
<?php
/**
 * @link http://www.yiiframework.com/
 * @copyright Copyright (c) 2008 Yii Software LLC
 * @license http://www.yiiframework.com/license/
 */

namespace yii\data;

/**
 * DataProviderInterface is the interface that must be implemented by data provider classes.
```

```
*
* Data providers are components that sort and paginate data, and provide them to widgets
* such as [[\yii\grid\GridView]], [[\yii\widgets\ListView]].
*
* For more details and usage information on DataProviderInterface, see the [guide article on data providers](guide:output-
*
* @author Qiang Xue <qiang.xue@gmail.com>
* @since 2.0
*/
interface DataProviderInterface
{
    /**
     * Prepares the data models and keys.
     *
     * This method will prepare the data models and keys that can be retrieved via
     * [[getModels()]] and [[getKeys()]].
     *
     * This method will be implicitly called by [[getModels()]] and [[getKeys()]] if it has not been called before.
     *
     * @param bool $forcePrepare whether to force data preparation even if it has been done before.
     */
    public function prepare($forcePrepare = false);

    /**
     * Returns the number of data models in the current page.
     * This is equivalent to 'count'($provider->getModels())`.
     * When [[getPagination|pagination]] is false, this is the same as [[getTotalCount|totalCount]].
     * @return int the number of data models in the current page.
     */
    public function getCount();

    /**
     * Returns the total number of data models.
     * When [[getPagination|pagination]] is false, this is the same as [[getCount|count]].
     * @return int total number of possible data models.
     */
}
```

```
public function getTotalCount();

/**
 * Returns the data models in the current page.
 * @return array the list of data models in the current page.
 */
public function getModels();

/**
 * Returns the key values associated with the data models.
 * @return array the list of key values corresponding to [[getModels|models]]. Each data model in [[getModels|models]]
 * is uniquely identified by the corresponding key value in this array.
 */
public function getKeys();

/**
 * @return Sort the sorting object. If this is false, it means the sorting is disabled.
 */
public function getSort();

/**
 * @return Pagination|false the pagination object. If this is false, it means the pagination is disabled.
 */
public function getPagination();
}
```

## WebDataProvider介绍

### 应用场景

在做相似项目/基金搜索的时候，遇到一个问题，就是无法直接使用对应的DataProvider，因为是通过search接口来完成数据的获取，所以没有对应的ActiveQuery，如果直接用ArrayDataProvider在数据量大的时候就会有性能问题，所以考虑新写一个WebDataProvider 对应着 搜索或者其他接口，能够提供分页获取数据的需求

## 实现思路

仿照ActiveDataProvider/ArrayDataProvider，定义一下web接口(CurlHttp)的interface，实现获取count和limit/offset的支持（扩展支持sort）

## 实例

## 具体源码

```
<?php
namespace common\data;

use lspbupt\curl\CurlHttp;
use yii\base\InvalidConfigException;
use yii\data\BaseDataProvider;
use common\helpers\ArrayHelper;

/**
 * WebDataProvider implements a data provider based on a data web interface which supports limit & offset
 * NOTE 要求接口返回的格式
 * [
 *     code =>
 *     data => [
 *         'total' => (int),
 *         'items' => []
 *     ]
 * ]
 * ```php
 *     $searchArr = ['query' => '大',];
 *     $provider = new WebDataProvider([
 *         'instance' => Yii::$app->upsearch,
 *         'url' => '/up/searchonline',
 *         'params' => $searchArr,
```

```
*         'pagination' => [
*             'pageSize' => 2,
*         ],
*     ]);
* $rows = $provider->getModels();
* ```
*/

class WebDataProvider extends BaseDataProvider
{
    /**
     * @var CurlHttp
     */
    public $instance;
    public $url;

    /**
     * @var array
     */
    public $params=[];

    /**
     * @var string|callable the column that is used as the key of the data models.
     * This can be either a column name, or a callable that returns the key value of a given data model.
     * If this is not set, the index of the [[models]] array will be used.
     * @see getKeys()
     */
    public $key;

    /**
     * @inheritdoc
     */
    protected function prepareModels()
    {
        $models = [];
        if (!$this->instance instanceof CurlHttp) {
            throw new InvalidConfigException('The "instance" property must be an instance of CurlHttp');
        }
    }
}
```

```
}

if (($pagination = $this->getPagination()) !== false) {
    $pagination->totalCount = $this->getTotalCount();

    if ($pagination->totalCount === 0) {
        return [];
    }
    $params = $this->params;
    $params['limit'] = $pagination->getLimit();
    $params['offset'] = $pagination->getOffset();

    $ret = $this->instance->setMethod(CurlHttp::METHOD_GET)->httpExec($this->url, $params);
    if (!empty($ret) && is_array($ret)) {
        if ($ret['code'] == 0) {
            $models = ArrayHelper::getRightValue($ret, 'data.items', []);
        }
    }
}
return $models;
}

/**
 * @inheritdoc
 */
protected function prepareKeys($models)
{
    if ($this->key !== null) {
        $keys = [];
        foreach ($models as $model) {
            if (is_string($this->key)) {
                $keys[] = $model[$this->key];
            } else {
                $keys[] = call_user_func($this->key, $model);
            }
        }
    }
}
```

```
        return $keys;
    } else {
        return array_keys($models);
    }
}

/**
 * @inheritdoc
 */
protected function prepareTotalCount()
{
    $totalCount = 0;
    if (!$this->instance instanceof CurlHttp) {
        throw new InvalidConfigException('The "container" property must be an instance of CurlHttp');
    }
    $ret = $this->instance->setMethod(CurlHttp::METHOD_GET)->httpExec($this->url, $this->params);

    if (!empty($ret) && is_array($ret)) {
        if ($ret['code'] == 0) {
            $totalCount = ArrayHelper::getRightValue($ret, 'data.total', 0);
        }
    }
    return (int)$totalCount;
}
}
```

Like Be the first to like this

No labels

地址：北京市朝阳区建国路86号佳兆业广场北塔6层梦想加空间601室

以太资本由艾普拉斯投资顾问(北京)有限公司运营，提供早期互联网项目的投融资对接服务

©2014-2017 以太资本 京ICP备14028208号

