ClickBlocks\Data\Converters\Collection

General information

Inheritance	ClickBlocks\Data\Converters\Converter
Child classes	no
Interfaces	no
Source	Framework/data/converters/collection.php

The Collection converter is intended for the array structure converting. By using this class you can remove part of the array or transform it. Examples:

```
// Defines our array that supposed to be converted.
$a = ['foo' => ['t1' => ['a' => 1, 'b' => 2, 'c' => 3],
                't2' => ['a' => 2, 'b' => 3, 'c' => 4],
                't3' => ['a' => 3, 'b' => 4, 'c' => 5]],
      'boo' => ['key1' => 'a',
                'key2' => 'b',
                'key3' => ['foo1' => 'test1',
                           'foo2' => 'test2']]];
// Creates Collection converter.
$converter = new Converters\Collection();
// Sets the converting mode.
// We want to transform array structure to another one.
$converter->mode = 'transform';
// Sets the transform schema.
$converter->schema = ['foo.$.b'    => 'key1.$',
                      'boo.key1'   => 'key2',
                      'boo.key3.*' => 'key3.*'];
// Transforms the array.
print_r($converter->convert($a));
// The output looks like this:
// Array
// (
//     [key1] => Array
//         (
//            [t1] => 2
//            [t2] => 3
//            [t3] => 4
//          )
//     [key2] => a
//     [key3] => Array
//          (
//             [0] => test1
//             [1] => test2
//          )
// )
// Now we want to remove some part of the array.
$converter->schema = ['foo.$.b',
                      'boo.key1',
                      'boo.key3.$'];
$converter->mode = 'exclude';
print_r($converter->convert($a));
// The output is shown below:
// Array
// (
//    [foo] => Array
//        (
//            [t1] => Array
//                (
```

```
//                     [a] => 1
//                     [c] => 3
//                 )
//             [t2] => Array
//                 (
//                     [a] => 2
//                     [c] => 4
//                 )
//             [t3] => Array
//                 (
//                     [a] => 3
//                     [c] => 5
//                 )
//         )
//     [boo] => Array
//         (
//             [key2] => b
//         )
// )
// And now we save some part of the array removing another part.
$converter->mode = 'reduce';
print_r($converter->convert($a));
// The output is shown below:
// Array
// (
//     [foo] => Array
//         (
//             [t1] => Array
//                 (
//                     [b] => 2
//                 )
//             [t2] => Array
//                 (
//                     [b] => 3
//                 )
//             [t3] => Array
//                 (
//                     [b] => 4
//                 )
//         )
//     [boo] => Array
//         (
//             [key1] => a
//             [key3] => Array
//                 (
//                     [foo1] => test1
//                     [foo2] => test2
//                 )
//         )
// )
```

> Note, that we use symbol $ in the schema in order to catch the keys of the converting array and symbol * in order to ignore them.

Public non-static properties

mode

```
public string $mode = 'transform'
```

The mode of the array structure converting. The valid values are "transform", "reduce" and "exclude".

The transformation mode can be used when you need to change array structure. The exclude mode is used for removing part of the array. And the reduce mode is useful if you want to save some part of array while the rest part is removed.

schema

```
public array $schema = []
```

The schema that describes the new array structure and conversion ways. The particular schema format depends on the value of **$mode** property.

separator

```
public string $separator = '.'
```

The separator of the key names in the array schema. If some array key contains the separator symbol you should escape it via backslash.

keyAssociative

```
public string $keyAssociative = '$'
```

This symbol corresponds to any elements with their keys of the transforming array in the array schema. If some array key is the same as **$keyAssociative** you should escape it via backslash.

keyNumeric

```
public string $keyNumeric = '*'
```

This symbol corresponds to any elements without their keys of the transforming array in the array schema. If some array key is the same as **$keyNumeric** you should escape it via backslash.

Public non-static methods

convert()

```
public array convert(array $entity)
```

$entity	array	the array to be converted.

Converts the given array to an array with other structure defining by the specified array schema. If the given value is not an array the exception will be thrown.

Protected non-static methods

transform()

```
protected array transform(array $array)
```

$array	array	the array for transformation.

Changes the array structure according to the given transformation schema.

reduce()

```
protected array reduce(array $array)
```

$array	array	the array to be reduced.

Returns the part of the given array that determining by the array schema.

exclude()

```
protected array exclude(array $array)
```

$array	array	the array to be reduced.

Removes some part of the array according to the array schema and returns the remainder array.

getValues()

```
protected mixed getValues(array $array, string $from, array $keys = null, boolean $allKeys = false)
```

$array	array	the given array.
$from	string	the element of the array schema that determines keys of array elements to be extracted.
$keys	array	the same as **$from** but parsed to an array.
$allKeys	boolean	determines whether the all keys is returned from the method or only keys captured by **$keyAssociative** or **$keyNumeric**.

Sequentially returns the array elements according to their keys.

getKeys()

```
protected array getKeys(string|array $keys)
```

$keys	string, array	a string or an array of the collection keys.

Returns an array of the normalized keys of the converting collection.