

Docs » Data Handling Category » Array Project » Tables

Tables

Transform column groups into rows, key-value pairs into arrays and row keys into columns

Anonymous · 26/10/2026

ArrayTable

```
use Derafu\Array\ArrayTable;
```

Columns into Rows

`columnsToRows($array, $keys = null)` transposes column groups into rows. Columns are read at zero-based positions.

```
ArrayTable::columnsToRows(['name' => ['Alice', 'Bob'], 'age' => [30]]);  
// [  
//     ['name' => 'Alice', 'age' => 30],  
//     ['name' => 'Bob', 'age' => null],  
// ]  
  
ArrayTable::columnsToRows(['name' => ['Alice']], ['name', 'email']);  
// [['name' => 'Alice', 'email' => null]]
```

The longest selected column determines the number of rows. Missing positions and selected columns that do not exist supply `null`. If no selected column exists, the result is empty. With no `$keys`, every column is selected.

Pairs into a Map

`pairsToMap($table)` requires each row to be an array with exactly two columns. The first column becomes the key and the second its value. Column order determines their role, including for rows with named columns.

```
ArrayTable::pairsToMap(['name', 'Alice'], ['age', 30]);  
// ['name' => 'Alice', 'age' => 30]
```

Keys may be scalar or null. Integers stay integers; other accepted keys are converted to strings before PHP applies its array-key rules. For example, `null` becomes `''`, `true` becomes `'1'` (stored as integer 1) and `2.7` becomes `'2.7'`.

Later duplicate keys replace earlier values. Invalid rows, or keys that are arrays, objects or resources, throw a translatable `Derafu\Array\Exception\ArrayException`, which extends `InvalidArgumentException`. Despite its name, this transformation can produce integer keys; use `ValueType::map()` separately if string-only keys are required.

Row Keys into a Column

`withKeyColumn($data, $idField)` adds each row's array key as a column:

```
ArrayTable::withKeyColumn(['u1' => ['name' => 'Alice'], 'u2' => ['name' => 'Bob']],  
    'id');  
// ['u1' => ['id' => 'u1', 'name' => 'Alice'], 'u2' => ['id' => 'u2', 'name' =>  
    'Bob']]
```

Outer keys are preserved. A row that already has the named column keeps its own value: the operation supplies a default identifier rather than overwriting it. All three transformations return new arrays.

Last updated on 26/10/2026