

导入

设置文件路径

`Easy Excel` 会根据传入的文件名后缀自动判断需要读取的文件类型。

```
use Dcat\EasyExcel\Excel;

// 导入xlsx
$allSheets = Excel::import('/tmp/users.xlsx')->toArray();

// 导入csv
$allSheets = Excel::import('/tmp/users.csv')->toArray();

// 导入ods
$allSheets = Excel::import('/tmp/users.ods')->toArray();
```

UploadFile

`Laravel` 和 `Symfony` 框架中可以直接导入浏览器上传的文件

```
use Dcat\EasyExcel\Excel;
use Illuminate\Http\Request;

class IndexController
{
    public function upload(Request $request)
    {
        // 直接读取前端上传的文件
        $allSheets = Excel::import($request->file('user_data'))->toArray();
    }
}
```

Filesystem

`Easy Excel` 对 `league/flysystem` 提供了支持, 通过 `Filesystem` 可以轻松的读取任意服务器上的文件。



此方法会将文件从文件所在服务器下载到本项目服务器临时目录，然后再读取文件内容。如果文件所在服务器和当前项目服务器都是同一台，则会造成资源浪费。

```
use Dcat\EasyExcel\Excel;
use League\Flysystem\Adapter\Local;
use League\Flysystem\Filesystem;

$adapter = new Local(__DIR__);

$filesystem = new Filesystem($adapter);

$allSheets = Excel::import('users.xlsx')->disk($filesystem)->toArray();
```

在Laravel中使用Filesystem

Easy Excel 对 Laravel 的 Filesystem 也提供了支持：

打开 `config/filesystems.php` 中找到如下配置：

```
return [
    ...

    'disks' => [

        'local' => [
            'driver' => 'local',
            'root' => storage_path('app'),
        ],

        ...

    ],
];
```

使用：

```
use Dcat\EasyExcel\Excel;

$array = [...];

// local 对应配置参数 config('filesystems.disks.local')
$allSheets = Excel::import('users.xlsx')->disk('local')->toArray();

// 也可以
$allSheets = Excel::import('users.xlsx')->disk(Storage::disk('local'))->toArray();
```

标题

Easy Excel 默认会把Excel表格中的**第一行**数据当做标题，然后合并到读取的数据行中（作为数据行的key使用）。如果开发者想使用自定义标题，则可以以下方式设置：



此处设置的标题无法改变列顺序。

```
use Dcat\EasyExcel\Excel;

$headings = ['id', 'email', 'name'];

// 导入xlsx
$allSheets = Excel::import('/tmp/users.xlsx')->headings($headings)->toArray();

var_dump($allSheets);
```

将得到如下结果

```
[
    'Sheet1' => [
        2 => ['id' => 1, 'email' => 'treutel@eg.com', 'name' => 'Brakus'],
        3 => ['id' => 2, 'email' => 'josefa@eg.com', 'name' => 'Eichmann'],
    ],
];
```

指定标题所在的行

如果标题不在**第一行**，则可以通过以下方法轻松的指定标题所在的行

```
use Dcat\EasyExcel\Excel;

$headings = ['id', 'email', 'name'];

// 指定第二行为标题行
$allSheets = Excel::import('/tmp/users.xlsx')
    ->headings($headings)
    ->headingRow(2)
    ->toArray();

// 也可以传闭包
$allSheets = Excel::import('/tmp/users.xlsx')
    ->headings($headings)
    ->headingRow(function (int $line, array $row) {
        // $line 为数据行在excel表中的行号，$row 为数据行内容
        return $line = 2;
    })
```

```
->toArray();

var_dump($allSheets);
```

将得到如下结果

```
[
    'Sheet1' => [
        3 => ['id' => 1, 'email' => 'treutel@eg.com', 'name' => 'Brakus'],
        4 => ['id' => 2, 'email' => 'josefa@eg.com', 'name' => 'Eichmann'],
    ],
];
```

禁用标题

```
use Dcat\EasyExcel\Excel;

// 导入xlsx
$allSheets = Excel::import('/tmp/users.xlsx')->headings(false)->toArray();

var_dump($allSheets);
```

将得到如下结果

```
[
    'Sheet1' => [
        2 => [1, 'treutel@eg.com', 'Brakus'],
        3 => [2, 'josefa@eg.com', 'Eichmann'],
    ],
];
```

读取数据

toArray

把所有表格数据转化为数组。

```
use Dcat\EasyExcel\Excel;

// 导入xlsx
$allSheets = Excel::import('/tmp/users.xlsx')->toArray();
```

```
var_dump($allSheets);
```

结果

```
[
    // 如果是xlsx和ods文件，此处下标为工作表名称，即 Sheet1
    // 如果是csv文件，此处下标为工作表序号，从 0 开始
    'Sheet1' => [
        // 此处下标是行号，由于标题占用了 1 行，所以数据行一般是从 2 开始
        2 => ['id' => 1, 'email' => 'treutel@eg.com', 'name' => 'Brakus'],
        3 => ['id' => 2, 'email' => 'josefa@eg.com', 'name' => 'Eichmann'],
    ],
];
```

collect

把所有表格数据转化为 `SheetCollection` 对象。



没错! `SheetCollection` 的功能与 `Laravel` 中的 `collection` 的功能是一模一样的, `collection` 对数组的操作非常方便, 所以此处集成进来了。

```
use Dcat\EasyExcel\Excel;

// 导入xlsx
$allSheetsCollection = Excel::import('/tmp/users.xlsx')->collect();
```

each

循环sheet表格

```
use Dcat\EasyExcel\Excel;
use Dcat\EasyExcel\Contracts\Sheet as SheetInterface;

Excel::import('/tmp/users.xlsx')->each(function (SheetInterface $sheet) {
    // 单独处理每个表格内容
    $sheetArray = $sheet->toArray();

    // 获取表格名称，如果是csv文件，则此方法返回空字符串
    $sheetName = $sheet->getName();

    // 表格序号，从 0 开始
    $sheetIndex = $sheet->getIndex();
```

```
var_dump($sheetArray);
})
```

打印结果

```
[
    // 此处下标是行号，由于标题占用了 1 行，所以数据行一般是从 2 开始
    2 => ['id' => 1, 'email' => 'treutel@eg.com', 'name' => 'Brakus'],
    3 => ['id' => 2, 'email' => 'josefa@eg.com', 'name' => 'Eichmann'],
]
```

first

读取第一个表格内容

```
use Dcat\EasyExcel\Excel;

// Dcat\EasyExcel\Contracts\Sheet
$sheet = Excel::import('/tmp/users.xlsx')->first();

// 表格名称
$sheetName = $sheet->getName();

var_dump($sheet->toArray());
```

打印结果

```
[
    // 此处下标是行号，由于标题占用了 1 行，所以数据行一般是从 2 开始
    2 => ['id' => 1, 'email' => 'treutel@eg.com', 'name' => 'Brakus'],
    3 => ['id' => 2, 'email' => 'josefa@eg.com', 'name' => 'Eichmann'],
]
```

active

读取保存文件前打开的表格内容

```
use Dcat\EasyExcel\Excel;

$activeSheetArray = Excel::import('/tmp/users.xlsx')->active()->toArray();

var_dump($activeSheetArray);
```

sheet

根据表格名称或位置索引读取指定表格内容，默认的表格名称通常为 `Sheet1`。



CSV 文件不支持根据名称读取表格内容。

```
use Dcat\EasyExcel\Excel;

// 根据表格名称读取表格内容
$sheet1 = Excel::import('/tmp/users.xlsx')->sheet('Sheet1');

// 根据表格的位置索引读取指定表格内容
$firstSheet = Excel::import('/tmp/users.xlsx')->sheet(0);
```

Sheet操作接口

Easy Excel 提供了一些非常简单实用的接口用于读取表格数据。

分块读取

当数据表内容比较多时，使用分块读取功能可以有效减少内存消耗，提高效率。

操作所有表格

```
use Dcat\EasyExcel\Excel;
use Dcat\EasyExcel\Contracts\Sheet as SheetInterface;
use Dcat\EasyExcel\Support\SheetCollection;

Excel::import('/tmp/users.xlsx')->each(function (SheetInterface $sheet) {

    // 每100行数据为一批数据进行读取
    $chunkSize = 100;

    $sheet->chunk($chunkSize, function (SheetCollection $collection) {
        // 此处的数组下标依然是excel表中数据行的行号
        $rows = $collection->toArray();

        ...
    });
});
```

操作单表格

```

use Dcat\EasyExcel\Excel;
use Dcat\EasyExcel\Contracts\Sheet as SheetInterface;
use Dcat\EasyExcel\Support\SheetCollection;

// 每100行数据为一批数据进行读取
$chunkSize = 100;

Excel::import('/tmp/users.xlsx')
    ->first()
    ->chunk($chunkSize, function (SheetCollection $collection) {
        // 此处的数组下标依然是excel表中数据行的行号
        $rows = $collection->toArray();

        ...
    });

```

过滤数据行

可以通过 `Sheet::filter` 方法过滤掉读取到的数据行

```

use Dcat\EasyExcel\Excel;
use Dcat\EasyExcel\Contracts\Sheet as SheetInterface;
use Dcat\EasyExcel\Support\SheetCollection;

// 每100行数据为一批数据进行读取
$chunkSize = 100;

$headings = ['id', 'email', 'name'];

Excel::import('/tmp/users.xlsx')
    ->headings($headings)
    ->first()
    ->filter(function (array $row, int $line) {
        // 过滤掉id小于10或者行号小于5的数据行
        return $row['id'] > 10 && $line > 5;
    })
    ->chunk($chunkSize, function (SheetCollection $collection) {
        ...
    });

```

each

```

use Dcat\EasyExcel\Excel;
use Dcat\EasyExcel\Contracts\Sheet as SheetInterface;
use Dcat\EasyExcel\Support\SheetCollection;

// 每100行数据为一批数据进行读取
$chunkSize = 100;

```



```
$headings = ['id', 'email', 'name'];

Excel::import('/tmp/users.xlsx')
    ->headings($headings)
    ->first()
    ->each(function (array $row, int $line, array $originalHeadings) {
        $id      = $row['id'];
        $email    = $row['email'];
    });
```

toArray

读取表格内容并转化为数组

```
use Dcat\EasyExcel\Excel;
use Dcat\EasyExcel\Contracts\Sheet as SheetInterface;

$activeSheetArray = Excel::import('/tmp/users.xlsx')->first()->toArray();

var_dump($activeSheetArray);

// 多表格
Excel::import('/tmp/users.xlsx')->each(function (SheetInterface as $sheet) {
    $sheetArray = $sheet->toArray();
});
```

collect

读取表格内容并转化为 `SheetCollection` 对象

```
use Dcat\EasyExcel\Excel;
use Dcat\EasyExcel\Contracts\Sheet as SheetInterface;

$collection = Excel::import('/tmp/users.xlsx')
    ->first()
    ->collect()
    ->groupBy('name');

var_dump($collection->toArray());

// 多表格
Excel::import('/tmp/users.xlsx')->each(function (SheetInterface as $sheet) {
    $sheetCollection = $sheet->collect();
});
```

0 条评论

未登录用户 



说点什么

 支持 Markdown 语法

[使用 Github 登录](#)

来做第一个留言的人吧!

BUILT BY **JQH**. COPYRIGHT © 2019

DESIGNED BY
ASSETS/SVG/JACK-MCDADE.SVG