

Introduction

About

Plates is a native PHP template system that's fast, easy to use and easy to extend. It's inspired by the excellent [Twig](#) template engine and strives to bring modern template language functionality to native PHP templates. Plates is designed for developers who prefer to use native PHP templates over compiled template languages, such as Twig or Smarty.

Highlights

- Native PHP templates, no new [syntax](#) to learn
- Plates is a template system, not a template language
- Plates encourages the use of existing PHP functions
- Increase code reuse with template [layouts](#) and [inheritance](#)
- Template [folders](#) for grouping templates into namespaces
- [Data](#) sharing across templates
- Preassign [data](#) to specific templates
- Built-in [escaping](#) helpers
- Easy to extend using [functions](#) and [extensions](#)
- Framework-agnostic, will work with any project
- Decoupled design makes templates easy to test
- Composer ready and PSR-2 compliant

Questions?

Plates is maintained by [RJ Garcia](#) and originally created by [Jonathan Reinink](#). Submit issues to [Github](#).

Simple example

Here is a simple example of how to use Plates. We will assume the following directory structure:

```
`-- path
   |-- to
   |   |-- templates
   |       |-- template.php
   |       |-- profile.php
```

Within your controller

```
// Create new Plates instance
$templates = new League\Plates\Engine('/path/to/templates');

// Render a template
```

```
echo $templates->render('profile', ['name' => 'Jonathan']);
```

The page template

profile.php

```
<?php $this->layout('template', ['title' => 'User Profile']) ?>
```

```
<h1>User Profile</h1>
<p>Hello, <?=$this->e($name)?></p>
```

The layout template

template.php

```
<html>
<head>
    <title><?=$this->e($title)?></title>
</head>
<body>

<?=$this->section('content')?>

</body>
</html>
```

Plates – Introduction to PHP Templates

Plates is our template system of choice. It's a native PHP template system that's fast and easy to use. Template systems can be used to many purposes, however I will talk about using it to create a simple bilingual website.

What is a template system?

It's a way to display dynamic data in the same way over multiple pages.

For example, Amazon has untold gazillions of products for sale. Every single one of them has a page. Instead of creating a file for each separate product, they created a template for that page, which is filled with data from some sort of database. If they want to change something on the page, all they need to modify now is one document, rather than a document for each product. Any change would take them years to implement.

Plates

Plates is a native PHP template system. It's very easy to set up and use. Most servers use PHP by default, so it's a perfect solution for someone who doesn't want to or know how to deal with servers.

Installation

All you need to do to is either install it via Composer by typing:

```
composer require league/plates
```

or alternatively you can upload these files straight to your projects' root directory. To download them visit the [releases page](#), select the version you want, and click the "Source code (zip)" download button.

Once that is done, you should see the following folder structure:

```
../project/  
  vendor/  
  composer.json  
  composer.lock  
  composer.phar  
  error.php  
  error_log
```

And that's it.

Simple example

You might have noticed that there are a few things missing. If you put all that on the server, you will get 404 Not Found error.

You need to add a Controller and Templates.

Within your controller (index.php):

```
<?php

require 'vendor/autoload.php';

// Create new Plates instance
$templates = new League\Plates\Engine('templates/');

// Render a template
echo $templates->render('profile', ['name' => 'Jonathan']);
```

You also need to create a new folder called **templates**:

```
../project/
  templates/
    template.php
    profile.php
  vendor/
  composer.json
  composer.lock
  composer.phar
  error.php
  error_log
  index.php
```

Within your page template (profile.php):

```
<?php $this->layout('template', ['title' => 'User Profile']) ?>

<h1>User Profile</h1>
<p>Hello, <?=$this->e($name)?></p>
```

The layout template (template.php):

```
<html>
<head>
  <title><?=$this->e($title)?></title>
</head>
<body>

<?=$this->section('content')?>

</body>
</html>
```

Language selection

The easiest way to create a site in two languages in Plates, is to modify the Controller so that it loads the correct text depending on the language selected.

Do to so we can use attributes. In your link to change the language from English to German, just add:

?lang=de

at the end, so that the full link becomes:

Deutsch

Now, we have to convert the Controller to the following:

```
<?php
require 'vendor/autoload.php';

// Create new Plates instance
$templates = new League\Plates\Engine('templates/');

$slide = ($_GET["lang"]);

if($slide == 'de') {
    // Render a template in german
    echo $templates->render('index', [
        'title' => 'Deutsch',
    ]);
}
else {
    // Render a template in your default language
    echo $templates->render('index', [
        'title' => 'English',
    ]);
}
```

and that's it. On the original load the title will be in English, and after changing the language it will be in German.

Escaping

For pure text strings, the following should be used:

<?=\$this->e(\$title)?>

however, if you want to include links or other HTML code, the following should be used:

<?= \$title ?>

The 1st escape includes a method similar to PHP htmlspecialchars(), and as such any HTML will be rendered as text.

Project

You can find all the information about Plates [here](https://nott-studios.co.uk/plates-introduction/) at their project's home.

<https://nott-studios.co.uk/plates-introduction/>