

A Slim 3 Primer

Rob Allen ~ @akrabat
May 2015

The C in MVC

Slim 3

- Created by Josh Lockhart (phptherightway.com)
- PSR-7 Request and Response objects
- Middleware architecture
- Built in DIC for configuration

Expecting first beta early June 2015

Installation

```
rob@swiftsure /www/dev/slim3 $ composer require slim/slim:dev-develop
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
- Installing nikic/fast-route (v0.4.0)
  Loading from cache

- Installing pimple/pimple (v3.0.0)
  Loading from cache

- Installing psr/http-message (0.10.1)
  Loading from cache

- Installing slim/slim (dev-develop d929342)
  Cloning d92934250bcefd14da957f94cfa344977341051f

Writing lock file
Generating autoload files
rob@swiftsure /www/dev/slim3 $ █
```

index.php

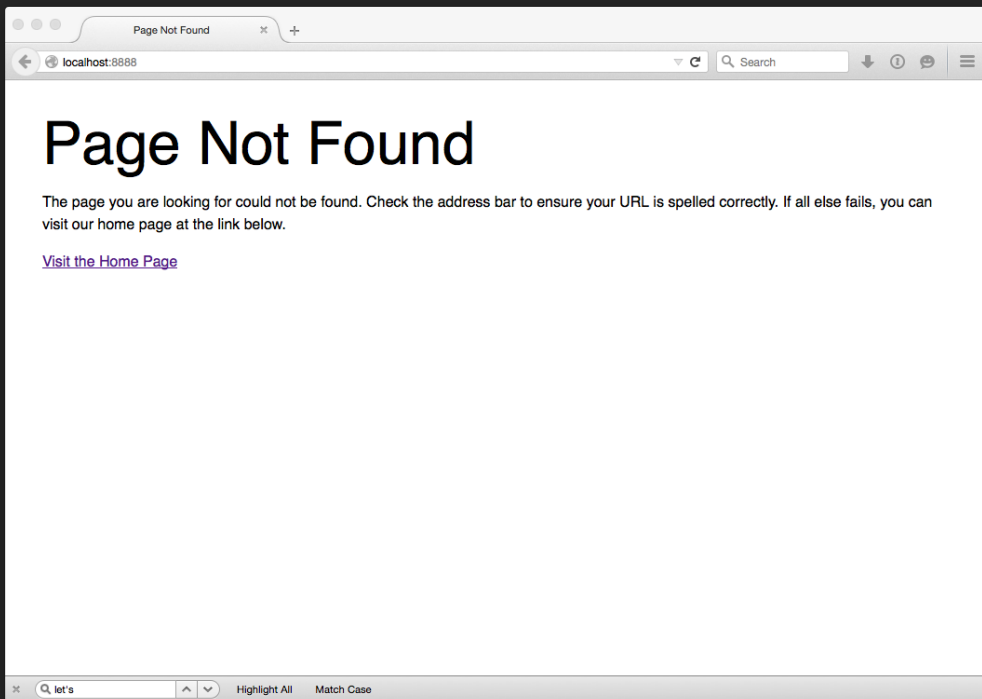
```
<?php
// Setup autoloader
require __DIR__ . '/vendor/autoload.php';

// Prepare app
$app = new \Slim\App();

// Run app
$app->run();
```

Run it

```
php -S localhost:8888
```



Page Not Found

localhost:8888

Search

Page Not Found

The page you are looking for could not be found. Check the address bar to ensure your URL is spelled correctly. If all else fails, you can visit our home page at the link below.

[Visit the Home Page](#)

let's

Highlight All

Match Case

Routes

Routes

```
<?php
require __DIR__ . '/../vendor/autoload.php';
$app = new \Slim\App();

$app->get('/', function($request, $response) {
    $response->write("Hello world");
    return $response;
});

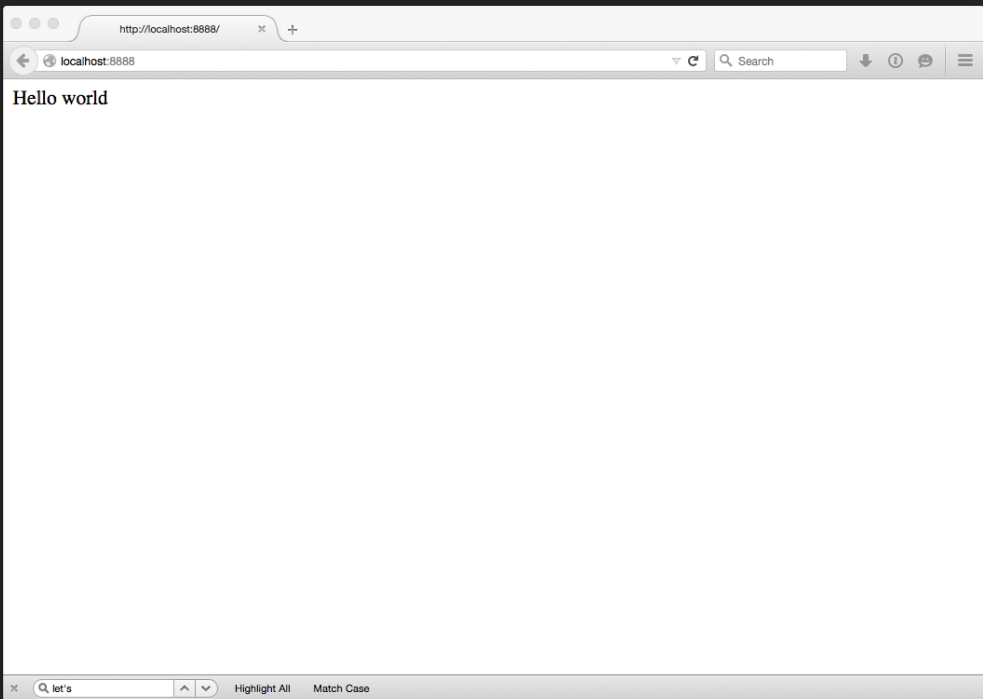
$app->run();
```

Routes

Method Pattern Action



```
$app->get('/', function($request, $response) {  
    $response->write("Hello world");  
    return $response;  
});
```



Dynamic routes

```
$app->get('/hello/{name}',  
  function($request, $response, $args) {  
    $name = $args['name'];  
    $name = htmlspecialchars($name);  
    return $response->write("Hello $name");  
  });
```

It's just Regex

```
$app->get('/user/{id:\d+}', $callable);
```

```
$app->get('/hello/{name:[\w]+}', $callable);
```

```
$app->get('/hello{a:{0,1}}{name:[\w]*}', $callable);
```

Named routes

```
// Name the route
```

```
$app->get('/hello/{name}', function(...) {...})  
    ->setName('hi');
```

```
// build link:
```

```
$link = $app['router']->urlFor('hi', ['name' => 'Rob']);
```

creates: /hello/Rob

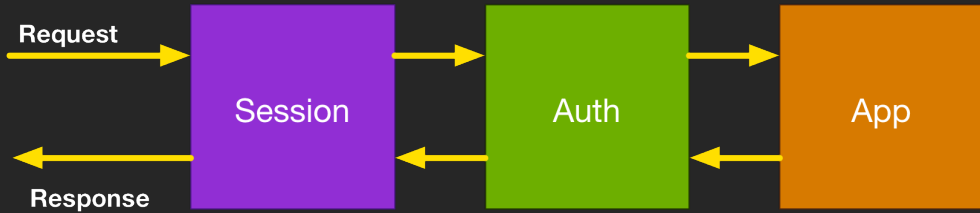
Middleware

Middleware

Middleware is code that exists between the request and response, and which can take the incoming request, perform actions based on it, and either complete the response or pass delegation on to the next middleware in the queue.

Matthew Weier O'Phinney

Middleware



Application middleware

```
$timer = function ($request, $response, $next) {  
    // before  
    $start = microtime(true);  
  
    // call next middleware  
    $response = $next($request, $response);  
  
    // after  
    $taken = microtime(true) - $start;  
    $response->write("<!-- Time taken: $taken -->");  
  
    return $response;  
}  
  
$app->add($timer);
```

Route middleware

Do stuff before or after your action!

```
$app->get('/hello/{name}', function(...) {...})
  ->add(function($request, $response, $next) {

    // before: sanitise route parameter
    $name = strip_tags($request->getAttribute('name'));
    $request = $request->withAttribute('name', $name);

    return $next($request, $response);
  })
```

Leverage middleware

Application level:

- Authentication
- Navigation
- Session

Route level:

- Access control
- Validation

Twig views

Installation

```
rob@swiftsure /www/dev/slim3/nataero $ composer require slim/twig-view
Using version ~1.0 for slim/twig-view
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
- Installing twig/twig (v1.18.1)
  Loading from cache

- Installing slim/twig-view (1.0)
  Loading from cache

Writing lock file
Generating autoload files
rob@swiftsure /www/dev/slim3/nataero $ █
```

Configure the view

```
<?php
return [
    // ...

    'view' => [
        'template_path' => 'app/templates',
        'twig' => [
            'cache' => 'cache/twig',
            'debug' => true,
            'auto_reload' => true,
        ],
    ],
];
```

Register the view

```
// Create the view object
$view = new \Slim\Views\Twig(
    $settings['view']['template_path'],
    $settings['twig']);

// add extensions
$twig = $view->getEnvironment();
$twig->addExtension(new Twig_Extension_Debug());

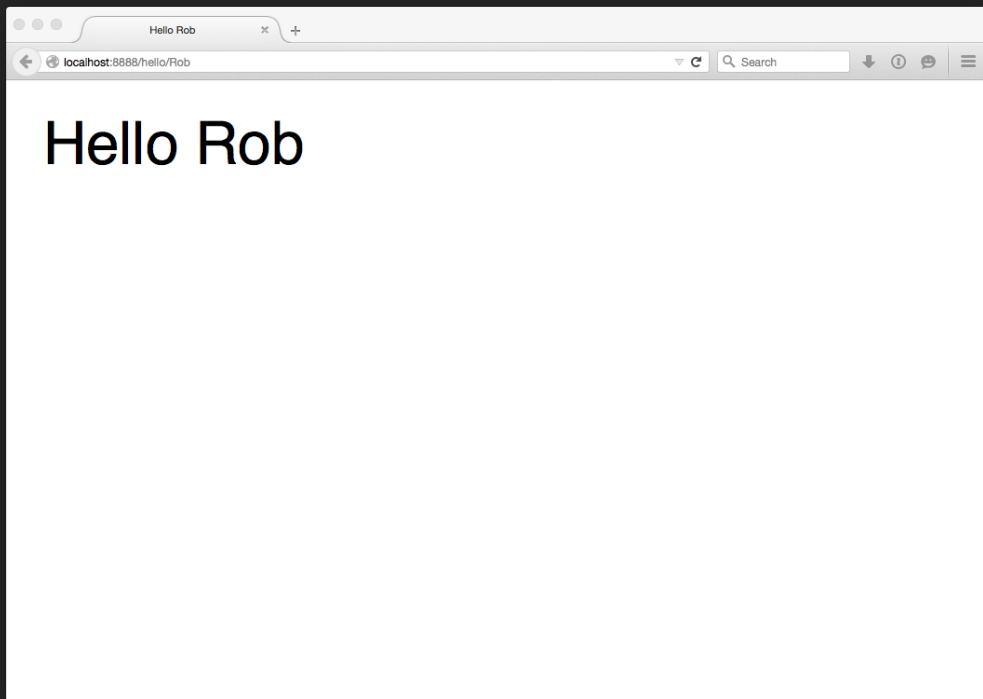
$app->register($app['TwigView']);
```


Template

```
<html>
  <head>
    <title>Hello {{ name }}</title>
    <link rel="stylesheet" href="/css/style.css">
  </head>
  <body>
    <h1>Hello {{ name }}</h1>
  </body>
</html>
```

Render

```
$app->get(  
    '/hello/{name}',  
    function($request, $response, $args) (use $app) {  
        $body = $app['view']->fetch('hello.twig', [  
            'name' => $args['name'],  
        ]);  
  
        return $response->write($body);  
    });
```



Hello Rob

Resources

- <http://slimframework.com>
- <http://docs-new.slimframework.com>
- <https://github.com/slimphp/Slim>

Questions?

Rob Allen - <http://akrabat.com> - @akrabat

Thank you!

Rob Allen - <http://akrabat.com> - @akrabat